

Oracle Linux Automation Manager 2

User's Guide



F80481-01
June 2023



Oracle Linux Automation Manager 2 User's Guide,
F80481-01

Copyright © 2022, 2023, Oracle and/or its affiliates.

Contents

Preface

Conventions	v
Documentation Accessibility	v
Access to Oracle Support for Accessibility	v
Diversity and Inclusion	v

1 About Oracle Linux Automation Manager

2 Setting Up Permissions for Organizations, Teams, and Users

Setting Up Organizations	2-3
Setting Up Teams	2-4
Setting Up Users	2-5

3 Setting Up Resources

Setting Up Projects	3-1
Accessing a Local Playbook	3-2
Accessing a Remote Playbook on Git	3-2
Setting Up Inventories and Hosts	3-4
Setting Up an Inventory	3-4
Setting Up a Host	3-4
Running One-Off Commands Against a Host	3-5
Setting Up Credentials	3-6
Setting Up Git Credentials	3-7
Setting Up Machine Credentials	3-7
Setting Up Ansible Galaxy/Automation Hub API Token Credentials	3-8
Setting Up an Ansible Galaxy Credential	3-9
Setting Up a Private Automation Hub Collections Credential	3-10
Setting Up Container Registry Credentials for your Execution Environment	3-10
Setting Up Credentials Permissions	3-11
Setting Up Job Templates	3-12

Setting Up a Job Template	3-12
Setting Up and Running Sample Playbooks	3-14
Setting Up Workflow Templates	3-17
About Workflow Templates	3-17
Setting Up a Workflow Template	3-19
Creating Schedules for Resources	3-21
Managing Execution Environments	3-22
About Execution Environments	3-22
Viewing the Execution Environment Pages	3-22
Creating Execution Environments	3-24
Managing Instance Groups	3-25
About Instance Groups and Service Mesh Node Types	3-25
Viewing the Instance Groups Summary	3-26
Editing an Instance Group	3-27
Creating an Instance Group	3-29

4 Using Views

Using the Dashboard	4-1
Viewing Jobs	4-2
Viewing Scheduled Activities	4-3
Viewing Activity Streams	4-4

Preface

[Oracle Linux Automation Manager 2: User's Guide](#) describes how to use Oracle Linux Automation Manager, which is a task engine management platform with a Web user interface.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

For information about the accessibility of the Oracle Help Center, see the Oracle Accessibility Conformance Report at <https://www.oracle.com/corporate/accessibility/templates/t2-11535.html>.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab>.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry

standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About Oracle Linux Automation Manager

Oracle Linux Automation Manager provides features that allow your organization to effectively manage infrastructure configuration through a browser user interface. Using Oracle Linux Automation Manager you can:

- Create and configure permissions for organizations, users, and teams that use Oracle Linux Automation Manager. You must have at least one organization before you can create a project.
- Create a project in Oracle Linux Automation Manager that imports Oracle Linux Automation Engine playbooks from various sources, such as the local machine which runs Oracle Linux Automation Manager or from a remote git repository.
- Create an inventory which specifies the hosts where playbook tasks run. You can do this manually or import an Oracle Linux Automation Engine inventory file.
- Create credentials which contains SSH authentication details for the hosts where the playbook tasks will run and any required credentials for Oracle Linux Automation Engine playbooks in remote repositories specified in projects.
- Create job templates that specify the projects and playbooks to run.
- Manually launch or schedule the launch of job templates and review the results.

The following example gives an overview of the steps a system administrator might follow to set up users, a team, and an organization to manage the running of playbooks on an inventory of target machines:

Note:

The below is an overview only. The individual chapters in this book discuss the creation of each resource in more detail.

1. Log in to Oracle Linux Automation Manager with the administrator user account created during the installation process.
See [Oracle Linux Automation Manager 2: Installation Guide](#) for more details on the installation process.
2. Create an **organization** called, for example, `Organization_Europe_Production`.
3. Create **users** of `Normal User` type, for example:
 - `NormalUser_Senior`.
 - `NormalUser_1` and `NormalUser_2`.
4. Create a **team**, called for example, `Team_Europe_Production`.
5. Add all the users created in the previous steps to `Team_Europe_Production`.
6. Navigate back to `Organization_Europe_Production`, the organization created in an earlier step.

7. Add team `Team_Europe_Production` to `Organization_Europe_Production` with an `Execute` role.

Each team member of `Team_Europe_Production` is added to `Organization_Europe_Production` with an `Execute` role.

8. In addition to the `Execute` role that user `NormalUser_Senior` receives in the previous step, you might want `NormalUser_Senior` to also have administrative privileges within the `Organization_Europe_Production` resource. If this is the case, add user `NormalUser_Senior` to the organization again, this time as an individual user, and specify the `Admin` role.

Upon completion of this step, the user `NormalUser_Senior` is displayed with both `Admin` and `Execute` roles in their `Roles` list.

9. Create a **project** with **playbooks**. For example, you might create a project called `Project_Europe_Production` with playbooks `Playbook_1`, `Playbook_2`, and `Playbook_3`.

A project must belong to an organization, so assign the project to `Organization_Europe_Production`.

10. Create an **Inventory**. For example, `Inventory_Europe_Production_Servers`.
11. Create **hosts**, for example, `Target_Host_1`, `Target_Host_2`, and `Target_Host_3`.

Each host must belong to an inventory, so add each host to `Inventory_Europe_Production_Servers` created in the previous step.

12. In a clustered installation, you might create an **instance group** of nodes, called for example, `Instance_Group_Production_Nodes`, to run playbooks and jobs on certain inventories.

See [Oracle Linux Automation Manager 2: Installation Guide](#) for more details on the installation process and different nodes in the service mesh.

13. Create a **job template** called, for example, `Job_Template_Production_1`. Steps to complete the creation of a job template include the following:

- In the **Project** field, specify the project created in a previous step, `Project_Europe_Production`.
- Once a Project has been specified, the playbooks associated with the project become available in the **Playbook** drop-down list. Pick one of its playbooks, for example `Playbook_1`.
- In the **Instance Group** field, select `Instance_Group_Production_Nodes`, the instance group created in a previous step.
- Click **Save** upon completion of all necessary fields.

14. **Launch** the playbook or create a **schedule** for it to run at a future time and date.

2

Setting Up Permissions for Organizations, Teams, and Users

This chapter describes how Oracle Linux Automation Manager enables administrators to create organizations, teams, and users where permissions can be allocated at each level. These permissions are based on role-based access controls.

Note:

You can integrate the Oracle Linux Automation Manager access levels discussed in this chapter with external identity management services, such as LDAP. Note that LDAP user account information does not appear in Oracle Linux Automation Manager until after the LDAP user account first logs in to Oracle Linux Automation Manager. See [Oracle Linux Automation Manager 2: Administrator's Guide](#) for more information about LDAP authentication and mappings for users, teams, and organizations.

Each level has the following functions:

- **Organizations:** Administrators can specify which organizations can run a playbook on what inventory by associating an organization to a project and an inventory. An organization can specify multiple projects and inventories, but each project and inventory can specify only one organization.
- **Teams:** A team belongs to one organization and a team can specify default permissions that apply to any user assigned to the team.
- **Users:** A user can belong to one or more organizations or teams. Administrators can create the following types of users:

- **System Administrator**

By default, a system administrator has system-wide administrator privileges.

- **System Auditor**

By default, a system auditor has read-only access for auditing purposes.

- **Normal User**

By default, a normal user has limited privileges. However, a normal user can be assigned high-privilege roles, such as **Admin**, or **Project Admin**, for a specific resource type.

For example, you might create a normal user called `Payroll_Engineer` and assign that user with an **Admin** system role to organization resource types called `Payroll_Organization` and `Finance_Organization`. In addition, you might also add the `Payroll_Engineer` user with an **Admin** system role to a team resource type called `Payroll_Team`.

Thus, a user may have different permissions depending on which resource type they are working with.

 **Note:**

- For purposes of security, Oracle recommends following the **principle of least privilege** when setting up user access.
- For the purpose of illustration, in this book we request you follow the examples with a system administrator account, as this allows the examples to cover the features available in Oracle Linux Automation Manager.

The following table describes the different roles available within each resource in Oracle Linux Automation Manager.

Table 2-1 Role-Based Access Control Role Descriptions

Resource Type	System Role	Description
Job Templates	Admin	Manages all aspects of the job template.
	Executive	Runs the job template.
	Read	Views settings for the job template.
Workflow Job Templates	Admin	Manages all aspects of the workflow job template.
	Executive	Runs the workflow job template.
	Read	Views settings for the workflow job template.
	Approve	Can approve or deny a workflow approval node.
Credentials	Admin	Manages all aspects of the Credential.
	Use	Use the credential in a job template.
	Read	Views settings for the credential.
Inventories	Admin	Manages all aspects of the inventory.
	Update	Updates the inventory.
	Ad Hoc	Runs ad hoc commands on the inventory.
	Use	Use the inventory in a job template.
Projects	Admin	Manages all aspects of the project.
	Update	Updates the project.

Table 2-1 (Cont.) Role-Based Access Control Role Descriptions

Resource Type	System Role	Description
	Use	Use the project in a job template.
	Read	Views settings for the project.
Organizations	Admin	Manages all aspects of the organization.
	Executive	Runs the executable resources in the organization.
	Project Admin	Manages all projects in the organization.
	Inventory Admin	Manages all inventories in the organization.
	Credential Admin	Manages all credentials in the organization.
	Workflow Admin	Manages all workflows in the organization.
	Notification Admin	Manages all notifications in the organization.
	Job Template Admin	Manages all job templates in the organization.
	Execution Environment Admin	Manages all execution environments in the organization.
	Auditor	Views all aspects of the organization
	Member	Makes user a member of the organization.
	Read	Views all organization settings.
	Approve	Approves or denies a workflow approval node.

Setting Up Organizations

The Oracle Linux Automation Manager setup requires that you set up at least one Organization resource.

To setup an organization, do the following:

1. Log in to Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Access** section, click **Organizations**.

The Organizations page appears.

4. Click the **Add** button.

The Create New Organization page appears.

5. In the **Name** field, enter a name for your organization. For example, `Organization 1`.
6. From the **Instance Groups** list, select an instance group for the Organization to run playbooks on.
7. From the **Galaxy Credentials** list, select a credential for <https://galaxy.ansible.com/> or Private Automation Hub, or both for accessing collections when running playbooks. When selecting more than one credential, Oracle Linux Automation Manager checks each server in the order in which the credentials are added to the Organization.
8. Click **Save**.

The newly created organization is displayed in the Details tab.

9. Click the **Access** tab.
10. Click the **Add** button.

The Add Roles dialog appears.

11. Select the **Users** option.
12. Click **Next**.
13. Select the check box next to the user accounts you want to add to the organization.
14. Click **Next**.
15. Select the check box next to the roles you want to assign to the users you have selected.
16. Click **Save**.

Setting Up Teams

To setup a team, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Access** section, click **Teams**.

The Teams page appears.

4. Click the **Add** button.

The Create New Team page appears.

5. In the **Name** field, enter a name for your team. For example, `Team 1`.
6. In the **Organization** field, click the **search** button.

The Select Organization dialog appears.

7. From the Organization list, select an organization.
8. Click **Select**.
9. Click **Save**.

The newly created team is displayed in the Details tab.

10. Click the **Access** tab.
11. Click the **Add** button.

The Add Roles dialog appears.
12. Select the **Users** option.
13. Click **Next**.
14. Select the check box next to the user accounts you want to add to the team.
15. Click **Next**.
16. Select the check box next to the roles you want to assign to the users you have selected.
17. Click **Save**.

Setting Up Users

To setup a user, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Access** section, click **Users**.

The Users page appears.
4. Click the **Add** button.

The Create New User page appears.
5. Optionally, complete the **First Name**, **Last Name** and **Email** fields.
6. In the **UserName** field, enter a user name for your user. For example, `User1`.
7. In the **Password** field, enter a password.
8. In the **Confirm Password** field, reenter the password.
9. From the User Type, select one of the following user types:
 - **Normal User**: You can limit users read and write access to the resources (such as inventory, projects, and so on) based on roles and privileges.
 - **System Auditor**: You can limit users to read-only permissions for all objects within Oracle Linux Automation Manager.
 - **System Administrator**: You can allow full system administration privileges (full read and write) for all objects within Oracle Linux Automation Manager.
10. In the **Organization** field, click the **search** button.

The Select Organization dialog appears.
11. From the organization list, select an organization.
12. Click **Select**.
13. Click **Save**.

The newly created organization is displayed in the Details tab.
14. Click the **Organizations** button.

All organizations that the user is part of appear in the list. This is a read-only page.

15. Click the **Teams** tab.

16. Click the **Associate** button.

The Select Teams page appears.

17. Select the check box next to the user teams you want to associate with the user.

18. Click the **Save**.

19. Click the **Roles** tab.

20. Click the **Add** button.

The Add User Permissions dialog displays.

21. Select one of the following resource types:

- **Job Templates**
- **Workflow job templates**
- **Credentials**
- **Inventories**
- **Projects**
- **Organizations**

22. Click **Next**.

The available options of the resource type you have selected are displayed. For example, if you have selected Job templates, then all available job templates appear.

23. Select the check box next to each resource item you want to add to the user.

24. Click **Next**.

The available roles you can apply to the resources you have selected are displayed.

25. Select the check box next to each role you want to apply to the resources you have selected.

 **Note:**

The roles you opt to apply are applied to all the resources you have selected.

For more information about available roles by resource type, see [Setting Up Permissions for Organizations, Teams, and Users](#)

26. Click **Save**.

3

Setting Up Resources

This chapter describes setting up resources. Resources are the primary configuration touch points between Oracle Linux Automation Manager and playbooks and inventory. These resources are projects, inventories, credentials, job templates, and workflow templates.

Setting Up Projects

A Project is a resource that represents a repository of Oracle Linux Automation Engine playbooks. The repository can be local or on a remote source code management (SCM) system such as Git.

To create a project, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Projects**.

The Projects page appears.

4. Click the **Add** button.

The Create New Project page appears.

5. In the **Name** field, enter a name. For example, `Project1`.
6. In the **Organization** field, click the **search** button.

The Select Organization dialog appears.

If your project has ansible collection requirements, you must ensure you select an organization that enables you to use one or more of the following options:

- **Download Collections from Repositories at Runtime**

To use this option, you must select an organization that has a credential set up for each repository you need to download collections from. See [Setting Up Ansible Galaxy/Automation Hub API Token Credentials](#) for more information.

- **Utilize a Custom Execution Environment Hosted in Private Automation Hub**

To use this option, you must select an organization with an execution environment whose **Image** property references a Private Automation Hub custom execution environment containing ansible collections.

From the organization list, select an organization.

7. Click **Select**.
8. From the **Source Control Credential Type** list, select one of the following:
 - **Manual:**

The **Manual** option enables you to select a local playbook located on the machine hosting Oracle Linux Automation Manager. You select the playbook of your choice from a drop-down list labeled **Playbook Directory**.

See [Accessing a Local Playbook](#) for more information.

- **Git:**

The **Git** option enables you to use a playbook from a remote repository such as from Git. Selecting the Git option displays source control fields for you to complete so that you can access the remote repository of your choice.

Go to [Accessing a Remote Playbook on Git](#) for information about setting up source details and credentials.

9. Click **Save**.

Accessing a Local Playbook

To make a local playbook accessible to your instance of Oracle Linux Automation Manager, ensure you have the required administrative privileges and carry out the following steps:

1. Log onto the server hosting Oracle Linux Automation Manager.
2. Create a directory to store your playbooks. The default location is `/var/lib/ol-automation-manager/projects`.

 **Note:**

The Oracle Linux Automation Manager might list the Project Base Path as `/var/lib/awx/projects` instead of `/var/lib/ol-automation-manager/projects`. Both locations are the same as `/var/lib/awx` is a symbolic link to `/var/lib/ol-automation-manager`.

3. Copy or create playbook files in the directory. For sample playbooks, see `/var/lib/ol-automation-manager/projects/demo_bundled`.
4. Set the playbook directory and files to the same user and group and permissions that Oracle Linux Automation Manager uses.

 **Note:**

Check the SELinux permissions and context settings if you have issues adding a project path.

Accessing a Remote Playbook on Git

Git is a distributed version control system containing repositories where you can store playbooks that you can clone to Oracle Linux Automation Manager systems as part of projects.

To access a remote playbook on Git, do the following:

1. Set up a new Git repository or use an existing Git repository on a remote Git server.
For example, GitHub hosts the public Ansible example projects at <https://github.com/ansible/ansible/tree/v2.12.2/examples>.

 **Caution:**

As this is a public repository, Oracle is not responsible for examples contained therein.

2. If you are using a private repository that requires SSH authentication, you must set up your SSH private and public keys on the system running Oracle Linux Automation Manager and copy the public key to your Git user account.
3. Return to the Oracle Linux Automation Manager UI and from the New Projects page, in the **Source Control URL** field, enter a URL to the SCM repository.
4. If you need to specify an SCM branch, tag, or commit in the **Source Control Branch/Tag/Commit** field, enter a branch, tags, or commit hash.
5. If you need to retrieve a refspec from a branch, in the **Source Control Refspec** field, enter a refspec.
6. If the Git repository you specified is private and requires credentials to access, from the **Source Control Credential** list, select a credential. For more information about setting up credentials, see [Setting Up Credentials](#).
7. Click **Save**.
8. If Oracle Linux Automation Manager requires access to the Git repository through a proxy, expand the left side menu and click **Settings**.
The Settings page appears.
9. Click **Jobs settings**.
The Details page appears.
10. Click the **Edit** button.
11. In the **Extra Environment Variables** field, enter the following:

```
{  
  "https_proxy": "proxy_address:proxy_port"  
}
```

In the previous example, *proxy_address* is the proxy address and *proxy_port* is the port for the proxy.

12. Click **Save**.

 **Note:**

If your playbook uses the OCI Ansible collection, see <https://docs.oracle.com/iaas/Content/API/SDKDocs/ansible.htm> and find the setup instructions relating to AWX.

Setting Up Inventories and Hosts

The following sections provide information about setting up inventories and hosts.

Setting Up an Inventory

An inventory is a collection of hosts against which jobs and commands can be launched.

To set up an inventory, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Inventories**.
The Inventories page appears.
4. Click the **Add** button.
A menu appears.
5. Click the **Add Inventory** option.
The Create new inventory page appears.
6. In the **Name** field, enter a name for your inventory. For example, `Inventory1`.
7. In the **Organization** field, click the **search** button.
The Select Organization dialog appears.
8. From the Organization list, select an organization.
9. Click **Select**.
10. In the **Variables** field (a code box), you can enter YAML or JSON code if you want to specify any inventory variables by doing the following:
 - a. Click the **YAML** or the **JSON** button to specify which syntax you wish to use.
 - b. Type the variables in the code edit box.
11. Click **Save**.

Setting Up a Host

The following sections show you different ways of navigating to a **Create new host** page.

Each host must belong to an inventory. To create a host by first selecting the inventory in which you wish to create the host, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Inventories**.
The Inventories page appears with a list of available inventories.

4. Click on the name of the inventory you in which you wish to set up a host.
The inventory is displayed in the Details Tab.
5. Click the **Hosts** tab to switch to the Hosts tab.
6. Click the **Add** button.
The Create new host page appears.
7. In the **Name** field, enter the IP address or host name.
8. Click **Save**.

The following steps show you an alternative method to the previous procedure for creating a host for a specific inventory:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Hosts**.
The Hosts page appears.
4. Click the **Add** button.
The Create New Host page appears.
5. In the **Name** field, enter the IP address or host name.
6. In the **Inventory** field click the **search** button.
The Select Inventory page appears.
7. From the inventory list, select an inventory.
8. Click **Select**.
9. Click **Save**.

Running One-Off Commands Against a Host

To run single commands using Oracle Linux Automation Engine built-in modules against one or more hosts in an inventory (similar to running a module on the command line), do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Inventories**.
The Inventories page appears with a list of available inventories.
4. Click on the name of the inventory containing the hosts against which you wish to run a command.
The inventory is displayed in the Details Tab.
5. Click the **Hosts** tab to switch to the Hosts tab.
The Hosts page appears listing any hosts associated with the inventory.
6. Select the check box beside one or more hosts on the left side.

 **Note:**

If you do not select any of the check boxes the system assumes you wish to run the command against all the hosts.

7. Click the **Run Command** button.
The Run Command page appears.
8. From the **Modules** list, select a module. For example, ping.
9. In the **Arguments** field, enter any arguments that are required by the selected module.
10. Click **Next**.
A list of available execution environments is displayed.
11. Select the **Execution Environment**.
If you do not select an environment, the command will run in the `OLAM_EE` execution environment by default.
12. Click **Next**.
A list of available machine credentials is listed.
13. Select the **Machine Credential** for the machines. See [Setting Up Credentials](#) for information about setting up credentials.
14. Click **Launch**.
The Output page appears showing the job operations in real-time and eventually displays a status of **Success** or **Failed**.

Setting Up Credentials

Oracle Linux Automation Manager provides functionality to create and store various types of credentials needed for working with private repositories and target inventory hosts. Examples of such credentials include:

1. Git Credentials:

Git credentials are used to access private registries you might be using as a source of projects and playbooks, and container images for your execution environments. They can include usernames and passwords, and SCM Private keys and passphrases.

2. Machine credentials:

Machine credentials are used to access target hosts through ssh. For example, a machine credential could be a user name and password pair, or a user name and ssh private key.

In Oracle Linux Automation Manager the credentials you enter are encrypted before being stored to the database. The encryption/decryption algorithm uses a variation of Fernet: a symmetric encryption cipher utilizing AES-256 in CBC mode alongside a SHA-256 HMAC. The key is derived from the `SECRET_KEY`, which is set in the Oracle Linux Automation Manager installation (see [Oracle Linux Automation Manager 2: Installation Guide](#) for more details).

The credentials are decrypted when they are extracted and used.

The following sections provide information about setting up various kinds of credentials on Oracle Linux Automation Manager.

Setting Up Git Credentials

To set up Git credentials, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Credentials**.
The Credentials page appears.
4. Click the **Add** button.
The Create New Credential page appears.
5. In the **Name** field, enter the name of the credential. For example, `GitCredential11`.
6. In the **Organization** field, click the **search** button.
The Select Organization dialog appears.
7. From the organization list, select an organization.
8. Click **Select**.
9. From the **Credential Type** list, select **Source Control**.
10. In the **UserName** field, enter the user name, if required.
11. In the **Password** field, enter a password, if required.
12. In the **SCM Private Key** field, enter the SSH private key (if you are using SSH authentication) that you set up on the host running Oracle Linux Automation Manager.
13. In the **Private Key Passphrase** field, enter a passphrase (if you have set up a passphrase for SSH authentication).
14. Click **Save**.

Setting Up Machine Credentials

Machine credentials are used to access target hosts through ssh.

To set up a machine credential, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Credentials**.
The Credentials page appears.
4. Click the **Add** button.
The Create New Credential page appears.
5. In the **Name** field, enter the name of the credential. For example, `InventoryMachineCredential11`.

6. In the **Organization** field, click the **search** button.
The Select Organization dialog appears.
7. From the organization list, select an organization.
8. Click **Select**.
9. From the **Credential Type** list, select **Machine**.
10. In the **Username** field, enter the user name, if required.
11. In the **Password** field, enter a password, if required.
12. Select **Prompt on launch** checkbox above the Password field if you want a password prompt to appear when launching job templates or job workflows against an inventory.
13. In the **SSH Private Key** field, enter the SSH private key that Oracle Linux Automation Engine uses to access target hosts when using SSH authentication.
14. In the **Signed SSH Certificate** field, enter a signed SSH certificate, if required.
15. In the **Private Key Passphrase** field, enter a passphrase (if you have set up a passphrase for SSH authentication).
16. Select the **Prompt on launch** checkbox above the Private Key Passphrase field if you want a private key passphrase prompt to appear when launching job templates or job workflows against an inventory.
17. From the **Privilege Escalation Method** list, select an escalation method if the playbooks you run in the machine require an escalation method, for example, `sudo`.
18. In the **Privilege Escalation Username** field, enter the user name, if required.
19. In the **Privilege Escalation Password** field, enter a password, if required.
20. Select **Prompt on launch** above the Privilege Escalation Password field if you want a password prompt to appear when launching job templates or job workflows against an inventory.
21. Click **Save**.

Setting Up Ansible Galaxy/Automation Hub API Token Credentials

Oracle Linux Automation Manager provides the Ansible Galaxy/Automation Hub Api Token credential type to enable you to access and download collections from a repository location. For example, you might create the following credentials:

- A credential called `My Ansible Galaxy Credential`, to access the publicly open location <https://galaxy.ansible.com/>. Such a location, being publicly open, does not require you to enter any authentication information, such as API Token, into the credential.
- A credential called `My Private Auto Hub API Token Credential`, to access one of your Private Automation Hub repositories, located for instance at `https://myprivateautomationhub/published`. Such a location would require authentication, so you would have to enter an API Token obtained from the Private Automation Hub console associated with that registry.

You can add more than one Ansible Galaxy/Automation Hub API Token credential to an organization. The order in which you add the credentials to an organization is

important as it determines the order in which repositories are searched when collections are to be downloaded for a project assigned to that organization.

! **Important:**

Make Sure You Have Enabled the Download of Collections and Roles

To allow collections and roles listed in an SCM project `requirements.yml` to be dynamically downloaded, you must enable the following **Job Settings**:

- **Enable Collection(s) Download**
- **Enable Role Download**

To edit Job settings:

1. Click **Settings** in the left-hand navigation menu.
2. Click the **Job settings** link.
3. Click **Edit** and proceed to edit the settings.
4. Click **Save**.

The following examples show you the steps to follow to create such credentials.

Setting Up an Ansible Galaxy Credential

To set up an Ansible Galaxy Credential for accessing collections, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Credentials**.
The Credentials page appears.
4. Click the **Add** button.
The Create New Credential page appears.
5. In the **Name** field, enter the name of the credential. For example, `My Ansible Galaxy Credential`.
6. In the **Organization** field, click the **search** button.
The Select Organization dialog appears.
7. From the organization list, select an organization.
8. Click **Select**.
9. From the **Credential Type** list, select **Ansible Galaxy/Automation Hub API Token**.
10. In the **Galaxy Server URL** field, enter the galaxy ansible URL <https://galaxy.ansible.com/>.
11. Click **Save**.

Setting Up a Private Automation Hub Collections Credential

Private Automation Hub credentials are used to access a Private Automation Hub instance using a URL and API token which contains collections in various repositories.

To set up a Private Automation Hub credential for accessing collections, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Credentials**.
The Credentials page appears.
4. Click the **Add** button.
The Create New Credential page appears.
5. In the **Name** field, enter the name of the credential. For example, `My Private Auto Hub API Token Credential`.
6. In the **Organization** field, click the **search** button.
The Select Organization dialog appears.
7. From the organization list, select an organization.
8. Click **Select**.
9. From the **Credential Type** list, select **Ansible Galaxy/Automation Hub API Token**.
10. In the **Galaxy Server URL** field, enter a Private Automation Hub repository URL. For more information about different repository types, see [Oracle Linux Automation Manager 2: Private Automation Hub User's Guide](#).
11. In the API Token field, enter an Private Automation Hub API token. For more information about creating an API token in Private Automation Hub, see [Oracle Linux Automation Manager 2: Private Automation Hub User's Guide](#).
12. Click **Save**.

Setting Up Container Registry Credentials for your Execution Environment

Oracle Linux Automation Manager

To set up a Private Automation Hub credential for accessing collections, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Credentials**.
The Credentials page appears.
4. Click the **Add** button.

The Create New Credential page appears.

5. In the **Name** field, enter the name of the credential. For example, `Registry Credentials`.
6. In the **Organization** field, click the **search** button.
The Select Organization dialog appears.
7. From the organization list, select an organization.
8. Click **Select**.
9. From the **Credential Type** list, select **Container Registry**.
10. In the Authentication URL field, enter the IP address or host name of the container registry you want to connect with. For example, this might be a Private Automation Hub instance hosting your execution environment container images or it could be <https://container-registry.oracle.com/>. For more information about Private Automation Hub, see [Oracle Linux Automation Manager 2: Private Automation Hub User's Guide](#).
11. In the **Username** field, enter the username of the container registry user that has access to the execution environment images you need.
12. In the **Password or Token** field, enter the password for the user.
13. Enable Verify SSL if your container registry is setup for SSL authentication.
14. Click **Save**.

Setting Up Credentials Permissions



Note:

You can only assign credential access to users and teams that belong to the same organization that the credential belongs to.

To assign users and teams permissions to use a credential carry out the following steps:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Credentials**.
The Credentials page appears.
4. Select an existing credential.
The selected credential is displayed with the Details tab active.
5. Click the **Access** tab.
The Access tab lists users that already have access to the credential, for example users with full administrative privileges.
6. Click the **Add** button.
The Add Roles page appears and shows options **Users** and **Teams**.
7. Click one of the options. For example, **Users**.
A list of available users appears.

 **Note:**

Users with full administrator privileges do not appear because they have permissions for all entities in Oracle Linux Automation Manager.

8. From the left column, select the check box beside each user you want to provide permission to use the credential.
9. Click **Next**.
The permissions you can assign to your selected users is displayed.
10. Select the check box beside each permission you wish to assign to your selected users.
11. Click **Save**.

In the previous example it is assumed that users, rather than teams, are being assigned new roles to access the credential. If instead of users you wish to assign access to teams, then you follow the same steps as the previous procedure except you choose teams rather than users when asked to specify which resource type you are configuring with new permissions.

Setting Up Job Templates

The following sections provide information about setting up job templates.

Setting Up a Job Template

Job Templates allow you to run playbooks task that are associated to a project against the hosts contained in an inventory. From a job template, you can also specify:

- Permissions: Which teams, users, or both can run the job.
- Notification: What notifications the job template triggers.
- Completed Jobs: A list of all completed jobs.
- Schedules: A list of schedules for when a job runs.

To set up a job template, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Templates**.
The Templates page appears.
4. Click the **Add** button.
A menu appears.
5. Click the **Add job template** option.
The Create New Job Template page appears.
6. In the **Name** field, enter a name for your job template. For example, `JobTemplate1`.
7. From the **Job Type** list, select one of the following:

- **Run:** Runs the playbook.
 - **Check:** Checks that the playbook can run without errors.
8. Select the **Prompt on launch** check box next to the **Job Type** field if you want a dialog box to confirm the choice of job type when the job is launched.

 **Note:**

Some fields have a **Prompt on launch** check box displayed next to them. Selecting such a check box next to a field means a dialog is displayed when the job is launched to confirm the value to be used for the field.

9. In the **Inventory** field, click the **search** button.
The Select Inventory dialog appears.
10. From the inventory list, select the inventory containing the hosts you want this job to manage.
11. In the **Project** field, click the **search** button.
The Select Project dialog appears.
12. Select the project containing the playbook you want this job to execute.
For example, you might select a project that contains the following sample Oracle Linux Automation Engine playbooks:
 - The sample Yum playbook that runs a yum update on your inventory.
 - The sample HTTPD runbook that create an HTTPD on your inventory.
13. Select the **Execution Environment**.
If you do not select an environment, the command will run in the `OLAM EE` execution environment by default.
14. From the **Playbook** list, select a playbook from your selected project. For example, if you are using the sample Yum playbook, select `yum_update.yaml`.
15. From the **Credentials** lists, select a credential you have set up to access the hosts specified in the inventory you selected, if required.
16. In the **Labels** field, enter a label that describes the job. You can use these labels in various Oracle Linux Automation Manager views to group and filter job templates and completed jobs.
17. In the **Variables** field (a code box), you can enter YAML or JSON code if you want to pass extra command-line variables to the playbook by doing the following:
 - a. Click the **YAML** or the **JSON** button to specify which syntax you wish to use.
 - b. Type the variables in the code edit box.
18. In the **Forks** field, use scroll arrows to select the number of forks allowed. This is the number of parallel or simultaneous processes that can run while executing an Oracle Linux Automation Engine playbook. An empty value or a value of 1 uses the default specified in `/etc/ansible/ansible.cfg`.
19. In the **Limit** field, you can enter the hostname patterns or IP addresses that further limit the hosts specified in the inventory where the job can run, if required.

20. From the **Verbosity** list, select a level of output that Oracle Linux Automation Manager displays as the job template runs. Options are as follows:
 - **0 (Normal)**
 - **1 (Verbose)**
 - **2 (More Verbose)**
 - **3 (Debug)**
 - **4 (Connection Debug)**
21. In the **Job Slicing** field, use scroll arrows to select the number of job slices allowed. Job slices enable you to divide the work done by a job template into the specified number of job slices, each running the same tasks against a portion of the inventory.
22. In the **Timeout** field, use scroll arrows to select the time in seconds before a job times out.
23. Enable **Show Changes** to show the changes made by an Oracle Linux Automation Engine task in a job output.
24. From the **Instance Groups** list, select the instance group you created when you installed Oracle Linux Automation Manager.
25. In the **Job Tags** field, enter one or more job tags. Tags enable you to run portions of large playbook that contain tags. You can specify more than one tag using commas to separate each tag.
26. In the **Skip Tags** field, enter one or more job tags to skip. You can specify more than one tag using commas to separate each tag.
27. In the **Options** section, check the following options if required:
 - **Privilege Escalation:** The playbook runs with administrator privileges.
 - **Provisioning Callbacks:** Creates a provisioning callback URL so a host can contact Oracle Linux Automation Manager and request a configuration update using this job template.
 - **Enable Webhook:** The playbook enables webhooks.
 - **Concurrent Jobs:** Oracle Linux Automation Manager can run multiple instances of the job template at the same time.
 - **Enable Fact Storage:** Uses cached facts if available and stores discovered facts in the cache.
28. Click **Save**.

Setting Up and Running Sample Playbooks

When you install Oracle Linux Automation Manager, sample Oracle Linux Automation Engine playbooks are also included in the `/var/lib/ol-automation-manager/projects/demo_bundled` directory. This directory contains the following playbooks:

- `httpd.yaml`: This playbook installs an Apache HTTP server on your target inventory.
- `yum_update.yaml`: This playbook runs a yum update on your target inventory.

To set up and run these playbooks as the admin user, do the following:

1. Log into Oracle Linux Automation Manager as the admin user.
2. Set up an organization, for example `Sample Yum and Apache Org`. For more information, see [Setting Up Organizations](#).
3. Set up credentials for the hosts where you want to run the playbooks as required, for example `Sample Yum and Apache Host Credential`. For more information, see [Setting Up Machine Credentials](#).
4. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
5. From the **Resources** section, click **Projects**.
The Projects page appears.
6. Click the **Add** tab.
The Create New Project page appears.
7. In the **Name** field, enter a name, for example, `Sample Yum and Apache project`.
8. In the **Organization** field, click the **search** button.
The Select Organization dialog appears.
9. From the Organization List, select an organization you have created, for example `Sample Yum and Apache Org`.
10. From the **Source Control Credential Type** list, select **Manual**.
11. From the Playbook Directory list, select **demo_bundled**.
12. Click **Save**.

 **Note:**

For more information on creating projects see [Setting Up Projects](#)

13. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
14. From the **Resources** section, click **Inventory**.
The Inventories page appears.
15. Click the **Add** button.
A menu appears.
16. Click the **Add Inventory** option.
The Create new inventory page appears.
17. In the **Name** field, enter a name for your inventory. For example, `Sample Yum an Apache Inventory`.
18. In the **Organization** field, click the **search** button.
The Select Organization dialog appears.
19. From the Organization list, select an organization you have created, for example, `Sample Yum and Apache Org`.
20. Click **Save**.

 **Note:**

For more information on creating inventories see [Setting Up an Inventory](#)

21. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
22. From the **Resources** section, click **Hosts**.
The Hosts page appears.
23. Click the **Add** button.
The Create New Host page appears.
24. In the **Name** field, enter the IP address or host name of the system you are connecting to. The host must be reachable.
25. In the **Inventory** field click the **search** button.
The Select Inventory page appears.
26. From the inventory list, select an inventory you have created, for example `Sample Yum` and `Apache Inventory`.
27. Click **Save**.

 **Note:**

Also see [Setting Up a Host](#) for information on setting up hosts for an inventory.

28. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
29. From the Resources section, click **Templates**.
The Templates page appears.
30. Click the **Add** button.
A menu appears.
31. Click the **Add job template** option.
The Create New Job Template appears.
32. In the **Name** field, enter a name for your job template. For example, `Sample Yum` or `Apache Job`.
33. From the Job Type list, select **Run**.
34. From the Inventory list, select an inventory, for example `Sample Yum` and `Apache Inventory`.
35. From the **Project** list, select a project, for example, `Sample Yum` and `Apache project`.
36. From the **Playbook** list, select either `yum_update.yaml` if you want to run a yum update on your host or `httpd.yaml` if you want to set up an Apache instance on your host.

37. From the **Credentials** lists, select a credential, for example `Sample Yum and Apache Host Credential`.
38. From the **Verbosity** list, select **0 (Normal)**.
39. If you are running the `httpd.yaml` playbook, enable **Enable Privilege Escalation** so the playbook runs with administrator privileges.
40. Click **Save**.
41. Click **Launch**.
42. The **Output** page will appear so you can observe the progress of the job.

 **Note:**

For more information on setting up a template see [Setting Up a Job Template](#)

Setting Up Workflow Templates

The following sections provide information about setting up workflow templates.

About Workflow Templates

A workflow template enables you to use a graphical tool, the **Workflow Visualizer**, to configure the run sequence of disparate components such as job templates and management jobs, as nodes in a linear graph-like design.

The components in a workflow do not have to share the same organizations and inventories.

Node Types in Workflow Templates

The nodes in a workflow can be any of the following components:

- Job Template
- Management Job
- Workflow Job Template
- Approval
- Project Syncs
- Inventory Source Sync

Workflow Design, Features, and Properties

The following list highlights some key aspects of workflow design, features and properties:

- **Workflow Graphs Read From Left to Right**

The Workflow Visualizer's graph-like design is read from left to right. For example, if a workflow design has `example_node_1` on the left connected by a straight line to `example_node_2` to the right, this indicates `example_node_1` runs immediately before `example_node_2`.

- **Parent Nodes and Child Nodes**

Continuing with the previous example, if *example_node_1* is configured to run immediately before *example_node_2*, then *example_node_1* is said to be the parent node of *example_node_2*.

A node can have more than one parent node. For example, you might have a workflow where you configure parent nodes *RunSampleYumPlaybook* and *RunSampleApachePlaybook* to run immediately before child node *InstallWebApplication*. In such a case the Workflow Visualizer shows the lines from each parent node converging directly on child node *InstallWebApplication*. Also see [Convergence Property](#).

- **Run Type Condition**

When you add a child node you must select from one of the following options to stipulate the condition under which the child node should run:

- **On Success** (default):

Select **On Success** to specify the child node you are adding should only run if the present node you are editing completes its run in a successful state. If you select this option, the connection to the added child node is shown as a **green line**.

- **On Failure** :

Select **On Failure** to specify the child node you are adding should only run if the present node you are editing completes its run in a failure state. If you select this option, the connection to the added child node is shown as a **red line**.

- **Always** :

Select **Always** to specify the child node you are adding should always run, regardless of the final state of the present node you are editing. If you select this option, the connection to the added child node is shown as a **blue line**.

- **Convergence Property**

The Convergence property of a node determines whether or not it runs based on the final status of the parent nodes that converge directly to it. This option is significant if the node has more than one parent node linking directly to it. You can select one of the following values for the Convergence property:

- **Any** (default): The **Any** option stipulates this node can run if any of the nodes that converge on this node complete as specified.

- **All** : The **All** option implies this node can only run if all of the nodes that converge on this node complete as specified.

Setting Up a Workflow Template

 **Note:**

For the purposes of the example, it is assumed the aim is to create a workflow of job templates that run the following sample playbooks:

- `yum_update.yaml`: This playbook runs a yum update on your target inventory.
- `httpd.yaml`: This playbook installs an Apache HTTP server on your target inventory.

For more information on setting up jobs to run the sample playbooks see [Setting Up and Running Sample Playbooks](#)

To set up a workflow template, do the following:

1. Create a job template named `RunSampleYumPlaybook` that runs the playbook `yum_update.yaml` against an inventory of servers.
2. Create a job template named `RunSampleApachePlaybook` that runs the play book `httpd.yaml` against an inventory of servers.
3. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
4. From the **Resources** section, click **Templates**.
The Templates page appears.
5. Click the **Add** button.
A menu appears.
6. Click the **Add workflow template** option.
The Create New Workflow Template page appears.
7. In the **Name** field, enter a name for your workflow template. For example, `SampleWorkflowTemplate`.

 **Note:**

The Name field is the only mandatory field on this page. The organization and inventory fields, for instance, do not have to be set at the workflow level.

8. Click **Save**.
The **Workflow Visualizer** page is displayed in the browser with a **Start** button.
9. Click the **Start** button in the Workflow Visualizer page.
10. The **Add Node** dialog appears.
 - a. The **Node Type** drop-down list at the top of the Add Node dialog allows you to select the type of resource to be added as a node. Accept the default choice of `Job Template`.

- b. Select `RunSampleYumPlaybook` from the list of available job templates.
- c. Simply accept the default option (`Any`) from the **Convergence** drop-down list.

 **Note:**

The property is only significant if the node you are adding has more than one parent converging on it.

- d. Optionally enter a **Node Alias** if you want the node to be displayed with a particular display name, different to the resource name, in the Workflow Visualizer.
- e. Click **Save**.

The **Add Node** dialog is dismissed.

The added node appears in the Workflow Visualizer.

11. Hover your mouse over the newly added node.
A small toolbar appears within the Workflow Visualizer next to the node.
12. Click the **add (“+”)** icon on the toolbar to add another node after the first node.
13. The **Add Node** dialog appears.

- a. As this is the second node in the sequence the dialog initially displays the following **Run Type** conditions to select from:

- `On Success` .
- `On Failure` .
- `Always` .

Accept the option selected by default (`On Success`).

- b. Click **Next**.

The **Add Node** dialog at this stage displays the same fields (**Node Type**, **Convergence**, **Node Alias**) as discussed earlier in the instructions dealing with the first node.

- c. Select `RunSampleApachePlaybook` and complete the fields
- d. Click **Save**.

The **Add Node** dialog is dismissed.

The added node, `RunSampleApachePlaybook`, appears in the Workflow Visualizer.

 **Note:**

The newly added node, `RunSampleApachePlaybook`, appears to the right of the first node (its parent) and the 2 nodes are connected by a **green line** (the green line indicates that the second node runs only after the first one completes successfully).

14. Click **Save** on the top-right of the Workflow Visualizer.

The Workflow Visualizer is dismissed and the newly created workflow template is displayed with the details tab active.

15. Click **Launch** when you are ready to run the workflow.

The **Output** page is displayed and the node currently in the running state is shown with a flashing green icon in its top left corner. The second node (`RunSampleApachePlaybook`) upon successful completion of the parent (`RunSampleYumPlaybook`).

Upon successful completion of the workflow each job displays the time it took for it to complete

Creating Schedules for Resources

Oracle Linux Automation Manager enables you to create schedules for the following resources:

- Job Templates
- Workflow Templates
- Inventory Sources
- Projects

The following example shows you how to create a schedule for a resource, in this case a job template:

1. Log into Oracle Linux Automation Manager.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Resources** section, click **Templates**.

The Templates page appears displaying a list of available templates.

4. Click on the name of the job template you wish to create a schedule for.

A tabbed view of the job template appears. Initially the Details tab is open.

5. Switch to the **Schedules** tab.
6. Click the **Add** button.

The Create New Schedule dialog appears. Type a descriptive name in the Name field.

Take note of the value in the **Local time zone field** when completing the **start date** calendar and **start time** drop-down list controls.

Note:

Best Practice: Set Your Schedules in UTC time.

The application saves your schedules in UTC, a world standard that is not adjusted for daylight saving time. Therefore, to avoid any confusion involving seasonal clock changes, it is best to set and track your schedules in UTC.

7. Click **Save**.

Your saved schedule is saved and displayed on the Schedule Details page.

It also appears in the Schedules view.

Managing Execution Environments

The following sections provide an overview on execution environments and how to create and view the execution environments in your system.

About Execution Environments

Execution environments are portable container images in which your Oracle Linux Automation Engine automation tasks run. Execution environments replace the Python virtual environments used in prior versions of the Oracle Linux Automation Engine product.

Execution Environments include the following components:

- The `oraclelinux:8` container image.
- Oracle Linux Automation Engine based on the Ansible version 2.12 open-source software.
- Python 3.8

You can also use custom execution environments created with the Builder utility. These custom execution environments include the base components listed above with additional components added. For more information about the Builder utility, see [Oracle Linux Automation Manager 2: Private Automation Hub User's Guide](#).

Note:

While the Builder utility enables you to create custom execution environment, in the case of an issue with the custom execution environment, Oracle support may ask you to revert to the Oracle provided OLAM-EE default image to troubleshoot the problem.

As portable self-contained container images, execution environments allow automation tasks to run consistently across multiple platforms.

The Oracle Linux Automation Manager comes with the following default execution environments:

- OLAM EE
- Control Plane Execution Environment

Viewing the Execution Environment Pages

To view an execution environment, do the following:

1. Log into Oracle Linux Automation Manager.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Administration** section, click **Execution Environments**.

The **Execution Environments** page displays a summary table relating to the execution environments available in your system. The table columns include those described in the following list:

- **Name**

The Name column contains the name of the execution environment.

- **Image**

The Image column contains the full image location, including the container registry, image name, and version tag, of the image used for the execution environment. For example, `container-registry.oracle.com/oracle_linux_automation_manager/olam-ee:latest`. Alternatively, the image location might be a Private Automation Hub instance hosting the `olam-ee` image or a custom image. For more information see, [Oracle Linux Automation Manager 2: Private Automation Hub User's Guide](#).

- **Organization**

The Organization column contains information about which organizations in your system can access the execution environment.

4. Click on the Name of an execution environment for more details.

The execution environment will be displayed in a tabbed view with the **Details** tab active. In addition to the Name, Image and Organization fields, the Details tab contains some additional fields, including the following:

- **Created**

The Created field contains the date and time the execution environment was created.

- **Last Modified**

The Last Modified column contains the date and time the execution environment was last modified.

- **Pull**

The Pull field contains a value that stipulates the conditions under which Oracle Linux Automation Manager should pull the container image from its repository location before running a resource job. The following values are possible:

- **Always**

This option stipulates that a container image is always pulled before running the container.

- **Missing**

This option stipulates that a container image is only pulled if the image is not present before running the container.

- **Never**

This option stipulates that a container image never pulled before running the container.

 **Note:**

It is important to keep the container images up to date to ensure the images have the latest CVE fixes.

5. Click the **Back to execution environments** button (the first button in the tab header row) to get back to the **Execution Environments** page.

Creating Execution Environments

To create an execution environment, do the following:

1. Log into Oracle Linux Automation Manager.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Administration** section, click **Execution Environments**.
4. Click **Add**.
The Create new execution environment page appears.
5. In the **Name** field, enter the name of the execution environment.
6. In the **Image** field, enter the image location, including the container registry, image name, and version tag, of the image used for the execution environment. For example, `container-registry.oracle.com/oracle_linux_automation_manager/olam-ee:latest`. Alternatively, you can point to a Private Automation Hub instance hosting the `olam-ee` image or a custom image. For more information see, [Oracle Linux Automation Manager 2: Private Automation Hub User's Guide](#).
7. From the **Pull** list, select one of the following values that stipulates the conditions under which Oracle Linux Automation Manager should pull the container image from its repository location before running a resource job:
 - **Always**
This option stipulates that a container image is always pulled before running the container.
 - **Missing**
This option stipulates that a container image is only pulled if the image is not present before running the container.
 - **Never**
This option stipulates that a container image never pulled before running the container.

 **Note:**

It is important to keep the container images up to date to ensure the images have the latest CVE fixes.

8. **Organization**
The Organization field contains information about which organizations in your system can access the execution environment.
9. **Registry credential**
The Registry credential field contains information about which credential your system must use to connect to a container registry. For more information, see [Setting Up Container Registry Credentials for your Execution Environment](#).

10. Click **Save**.

Managing Instance Groups

This chapter looks at how you create and manage instance groups with Oracle Linux Automation Manager.

About Instance Groups and Service Mesh Node Types

The service mesh provided by Oracle Linux Automation Manager is an overlay network that provides a mesh of nodes of different types, such as controller nodes and execution nodes, to enable you to manage your jobs across different inventories.

Oracle Linux Automation Manager provides instance groups to enable you to organize the different hosts in its service mesh into separate groups, depending upon their node type and purpose. For example, you may wish to create an instance group called `mesh_nodes_for_production` with nodes to run jobs on your production machines, and another group called `mesh_nodes_for_development_machines` to run jobs on your development servers.

The following list outlines key points about instance groups and nodes:

- **One Job Queue per Instance Group**

Each instance group has its own job queue, and any node in the instance can process a job from that queue.

- **Configuration of Additional Nodes**

The nodes initially available for association with Instance Groups are those you configured for your mesh topology during the installation process. If you require additional nodes, for example for a new instance group you have created, then you need to follow the same process you used to create the initial nodes (see [Oracle Linux Automation Manager 2: Installation Guide](#) for more information on configuring nodes for a mesh topology).

- **Nodes Can Belong to More Than One Instance Group**

For example, you may wish to make some nodes available across the cluster, whilst reserving others for a specific instance group.

- **Associating Instance Groups With Resources**

By default, jobs are run on nodes in the **Execution** instance group you created during the installation process (see [Oracle Linux Automation Manager 2: Installation Guide](#)).

However, if required, you can specify instance groups of your choice by associating them with one of the following resources:

- **Job Template**
- **Inventory**
- **Organization**

When a job runs, Oracle Linux Automation Manager first checks the job template associated with the job to see which instance group the job should be assigned to. If there is no instance group associated with the job template, Oracle Linux Automation Manager next checks the inventory resource linked to the job. If there is no instance group associated with the inventory resource, the Organization resource is checked.

- **Instance Groups Provided by Default**

Oracle Linux Automation Manager comes with the following instance groups:

- **controlplane** :

The `controlplane` instance group contains control nodes. Control nodes run persistent Oracle Linux Automation Manager services, such as the task dispatcher, as well as project and inventory updates and system jobs, but not regular jobs. Control nodes do not have execution capability.

- **execution**:

The `execution` instance group contains execution nodes. Execution nodes run jobs under `ansible-runner` with `podman` isolation. Execution nodes only run user-space jobs.

The following sections show you how to view and edit instance groups, how to associate jobs and resources with specific instance groups, as well as how to create groups of your own.

Viewing the Instance Groups Summary

To view the Instance Groups page, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Administration** section, click **Instance Groups**.

The Instance Groups page appears.

The **Instance Groups** page displays a summary table relating to the instance groups in the system. The table columns include those described in the following list:

- **Name**

The Name column contains the name of the Instance Group. Clicking the name takes you to a tabbed view of that instance's properties. The **Details** tab is the active tab initially.

- **Running Jobs**

The Running Jobs column contains the number of jobs associated with the instance group that are currently running.

- **Total Jobs**

The Total Jobs column contains the sum of the number of jobs associated with the instance group that have completed, and the number of jobs associated with the group that are still running.

- **Capacity**

The Capacity column displays a **Used capacity** progress bar that indicates the percentage of capacity of the instance group that is currently being used.

- **Action**

The Action column displays an **Edit instance group** icon that allows you to edit the instance.

Editing an Instance Group

The following sections show you how to edit an instance group and change properties on the nodes associated with the group.

Editing Instance Group Policy Properties

You can set the following Policy properties on an instance group:

- **Policy instance minimum**
This is the minimum number of instances that will be automatically assigned to the group when new provisioned instances come online.
- **Policy instance percentage**
Minimum percentage of all instances that will be automatically assigned to this group when new provisioned instances come online.

To set the policy properties of an instance group, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account and navigate to the **Instance Groups** page.
2. In the summary table displayed on the Instance Groups page, go to the row with the group whose policy instance properties you wish to edit.
3. Click on the **Edit instance group** icon in the **Action** column.
The **Edit details** page appears.
4. Enter values of your choice in the properties **Policy instance minimum** and **Policy instance percentage** fields.
5. Click **Save**.
The tabbed view of the instance appears.

Associating and Disassociating Nodes with a Group Instance

The following steps show you how to associate and disassociate nodes with a group instance:

1. Log into Oracle Linux Automation Manager with an administrator user account and navigate to the **Instance Groups** page.
2. In the summary table displayed on the Instance Groups page, go to the row with the group whose node association you are looking to change.
3. Click on the group instance name in the **Name** column.
A tabbed view of that instance's properties appears. Initially the **Details** tab is open.
4. Click on the **Instances** tab.
The node instances currently associated with the group are displayed in a table.
5. To **associate** more node instances with the group, do the following:
 - a. Click on the **Associate** button.
The Select Instances dialog appears displaying available node instances.
 - b. Select the check boxes next to the nodes you wish to associate with the group instance.

- c. Click **Save**.

The Instances page will be displayed and the nodes you selected appear in the list on the page.

6. To **disassociate** nodes from the current group instance, do the following:
 - a. Select the check box next to each host you wish to disassociate from the current group and click the **Disassociate** button.
A dialog will ask you to confirm your actions.
 - b. Confirm your actions if you are happy to proceed by clicking **Disassociate** button on the dialog.

The dialog is dismissed and the nodes you disassociated from the group no longer appear in the list.

Modifying Node Instance Properties

The following procedures show you how to modify the properties on a specific node.

A node instance's **capacity** determines how many forks it can run simultaneously. Capacity thus impacts the number of systems a node can run jobs for in parallel. To modify a node instance's **capacity**, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account and navigate to the **Instance Groups** page.
2. In the summary table displayed on the Instance Groups page, go to the row with the group associated with the node you wish to edit.
3. Click on the group instance name in the **Name** column.
A tabbed view of that instance's properties appears. Initially the **Details** tab is open,
4. Click on the **Instances** tab.
The node instances associated with the group are displayed in a table.
5. Go to the row with the node instance you wish to modify.
6. To adjust the **Capacity**, drag the slider control in the **Capacity Adjustment** column to the value of your choice.

Note:

- The change is saved automatically. There is no Save button.
- Consider whether the new capacity setting is suitable for other instance groups the node might belong to.

The **enable/disable** setting enables you to control whether or not a node instance is available for jobs to be assigned to it. For example, you might need to temporarily disable one of your node instances for maintenance. To change a node instance's **enable/disable** setting, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account and navigate to the **Instance Groups** page.

2. In the summary table displayed on the Instance Groups page, go to the row with the group associated with the node you wish to edit.
3. Click on the group instance name in the **Name** column.
A tabbed view of that instance's properties appears. Initially the **Details** tab is open,
4. Click on the **Instances** tab.
The node instances associated with the group are displayed in a table.
5. Go to the row with the node instance you wish to modify.
6. To change a node instance's **enable/disable** setting, toggle the enable-disable toggle switch in the **Action** column.

 **Note:**

- The change is saved automatically. There is no Save button.
- Consider whether the new setting is suitable for other instance groups the node might belong to.

Creating an Instance Group

The following steps show you how to create an instance group:

1. Log into Oracle Linux Automation Manager with an administrator user account and navigate to the **Instance Groups** page.
2. Click the **Add** button.
A menu appears.
3. Click the **Add instance group** option.
The Create new instance group appears.
4. In the **Name** field, enter a name for your instance group. For example, `NodesForDataCentre1`.
5. Optionally, enter integers into the **Policy instance minimum** and **Policy instance percentage** field.

 **Note:**

The policy instance fields are defined as follows:

- **Policy instance minimum:**
This is the minimum number of instances that will be automatically assigned to the group when new provisioned instances come online.
- **Policy instance percentage:**
Minimum percentage of all instances that will be automatically assigned to this group when new provisioned instances come online

6. Click **Save**.

The newly created instance group is displayed in a tabbed view with the Details tab active.

7. Click the **Instances** tab to switch to the Instances tab.

The node instances currently associated with the group are displayed in a table.

8. To associate more node instances with the group, click on the **Associate** button.

The Select Instances dialog appears displaying available node instances.

9. Select the check boxes next to the nodes you wish to associate with the group instance.

Click **Save**.

The Instances page will be displayed and the nodes you selected appear in the list on the page.

4

Using Views

This chapter describes how to use Oracle Linux Automation Manager views.

Using the Dashboard

The Dashboard view provides a summary of the latest state of your jobs, inventories, hosts, and projects.

To view the **Dashboard**, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Views** section, click **Dashboard**.

The Dashboard page consists of the following groups of controls:

Statistic Buttons

At the top of the page there is a row of clickable buttons with high level summary statistics for hosts, inventories, and projects.

Job activity tabs

The tabs available are

1. Job status
2. Recent Jobs
3. Recent Templates

When the dashboard page loads the Job status tab is selected by default and displays a graph of historical data for various job activities.

The following sections describe the groups of controls in more detail.

Statistic Buttons

The statistic buttons consolidate the following statistics in one display area:

Hosts

The Hosts button shows the total number of hosts included in all inventories to which the user has access.

Clicking the Hosts button takes you to the Hosts page.

Failed Hosts

The Failed Hosts button shows the number of hosts, to which the user has access, where the last job failed.

Clicking the Failed Hosts button takes you to a view that lists these hosts.

Inventories

The Inventories button shows the total number of inventories to which the user has access.

Clicking the Inventories button takes you to the inventories page.

Inventory Sync Failures

The Inventory Sync Failures button shows the number of inventories, to which the user has access, where synchronization with source inventory lists have failed.

Clicking the Inventory Sync Failures button takes you to a view that lists these inventories.

Projects

The Projects button shows the total number of projects to which the user has access.

Clicking the Projects button takes you to the Projects page.

Project Sync Failure

The Project Sync Failure button shows the number of projects, to which the user has access, where synchronization with the source Oracle Linux Automation Engine playbook have failed.

Clicking the Project Sync Failure button takes you to a view that lists these projects.

Jobs Activity Tabs

The following list describes the job activity tabs.

Job status

This tab consists of a configurable graph that displays historical data for various job activities. You can select from the following filter options to configure the graph to view the data you are most interested in:

1. **Time Period**

The time period filter options are `Past month`, `Past two weeks`, `Past week`, and `Past 24 hours`.

2. **Job Type**

The time period filter options are `All job types`, `Inventory sync`, `SCM update`, and `Playbook run`.

3. **Status**

The Status filter options are `All jobs`, `Successful jobs`, and `Failed jobs`.

Recent Jobs

The Recent Jobs tab shows a table of the most recently run jobs. The table includes job status as well as start and finish times for each job.

Each job that can be relaunched also has a **Relaunch Job** button in its row.

Recent Templates

The Recent Templates tab shows a table of the most recently used templates.

The templates that can be launched have **Launch Template** button in their rows.

Viewing Jobs

To view the **Jobs** page, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.

2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Views** section, click **Jobs**.

The Jobs view table that lists all activities that have been run in Oracle Linux Automation Manager including job template runs, workflow runs, project synchronizations, inventory synchronizations, one-off commands on hosts, and so on.

The table columns are described in the following list:

Column 1 (Unnamed column)

The first column contains a toggle button to expand or collapse the row. Expanding the row reveals more fields relating to the job, for example **Launched By**, **Project** and **Inventory**.

Column 2 (Unnamed column)

The second column contains a check box that enables you to select the row.

Name

The Name column contains the name of the job. You can sort by this column by clicking the column header.

Status

The Status column contains the status of the job, for example `Running`, `Successful`, `Failed`. You can sort by this column by clicking the column header. If the job is in `Running` status you have the option of canceling it by selecting the check box in the leftmost column and clicking the **Cancel jobs** button.

Type

The Type column contains the type of job, for example `Management Job`, `Command`, `Playbook Run`.

Start Time

The Start Time column contains the Start time of job. You can sort by this column by clicking the column header.

Finish Time

The Finish Time column contains the Finish time of job. You can sort by this column by clicking the column header.

Actions

This column contains a **Relaunch Job** button if the job can be relaunched.

Viewing Scheduled Activities

To view the **Schedules** page, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Views** section, click **Schedules**.

The Schedules view displays a table that shows scheduled management jobs, job template runs, workflow template runs, and project SCM updates from the Schedules view.

The table columns are described in the following list:

Column 1 (Unnamed column)

The first column contains a check box that enables you to select the row.

Name

The Name column contains the name of the schedule. You can sort by this column by clicking the column header.

Type

The Type column contains the type of the schedule, for example `Management Job`.

Next Run

The Next Run column contains the date and time of the next scheduled run. You can sort by this column by clicking the column header.

Actions

The Action column contains the following buttons:

- **Edit Schedule** button that enables you to click and edit the schedule.
- **Schedule is active** on/off toggle switch that enables you to toggle the active setting of the schedule.

Viewing Activity Streams

The Activity Stream page displays table that lists changes that have taken place on all objects in Oracle Linux Automation Manager, for example edits and updates on jobs, inventories, credentials, users, schedules etc.

To view the **Activity Stream** page, do the following:

1. Log into Oracle Linux Automation Manager with an administrator user account.
2. Display the left navigation menu if it is not already visible by toggling the Global navigation menu button in the top-left corner of the page.
3. From the **Views** section, click **Activity Stream**.

 **Note:****You can also use the View activity stream Button on Pages**

In addition to the Activity Stream link in the left navigation menu, you can also click the **View activity stream** button in the top-right corner of most pages.

One advantage of the buttons on the pages is that they append a filter parameter to the Activity Stream URL so the view initially shows only the data that relates to the object you are currently viewing. For example, the View activity stream button on the Jobs page points to `https://
olam.example.com/#/activity_stream?type=job` .

The columns of the table displaying the changes are described in the following list:

Time

The time column contains the date and time of the change. You can sort by this column by clicking the column header.

Initiated by

The Initiated by column gives the user name of the account that initiated the change. You can sort by this column by clicking the column header.

Event

The Event column describes the change.

The following are example entries from a test system:

- updated job_template Sample Yum
- associated MyTestOrg execute_role to NewUser1

Actions

The Actions column contains a **View event details** button. Clicking the button displays an event log for the change.

Filters:

You can set a filter on the Activity Stream view to limit the events to a particular object. The following filters are available from the Options menu drop-down list in the top-right of the page.

The following filters options are available:

- Views
- Dashboard (all activity)
- Jobs
- Schedules
- Workflow Approvals
- Resources
- Templates
- Credentials
- Projects
- Inventories
- Hosts
- Access
- Organizations
- Users
- Teams
- Administration
- Credential Types
- Notification Templates
- Instances
- Instance Groups
- Applications & Tokens
- Execution Environments

- Settings