Get Started

Ansible Automation Platform 2.6

May 12, 2026

Contents

1 Get started

Get started automating with playbooks

An Ansible Playbook is a blueprint for automation tasks, which are actions executed with limited manual effort across an inventory of solutions. Playbooks tell Ansible what to do on which devices.

A playbook automatically executes the same action across a specified inventory type, such as a set of routers, replacing manual repetitive work.

Playbooks are regularly used to automate IT infrastructure, such as operating systems, Kubernetes platforms, networks, security systems, and code repositories such as GitHub.

You can use playbooks to program applications, services, server nodes, and other devices, without the effort of creating everything from scratch. Playbooks, and the conditions, variables, and tasks within them, can be saved, shared, or reused indefinitely. This makes it easier for you to codify operational knowledge and ensure that the same actions are performed consistently.

How do Ansible Playbooks work

Ansible Playbooks are lists of tasks that run automatically for your specified inventory or groups of hosts. One or more Ansible tasks can be combined to make a play, that is, an ordered grouping of tasks mapped to specific hosts.

Tasks are executed in the order in which they are written.

A playbook can include one or more plays.

A playbook is composed of one or more `plays` in an ordered list.

The terms `playbook` and `play` are sports analogies.

Each play executes part of the overall goal of the playbook, running one or more tasks.

Each task calls an Ansible module.

Playbook

A list of plays that define the order in which Ansible performs operations, from top to bottom, to achieve an overall goal.

Play

An ordered list of tasks that maps to managed nodes in an inventory.

Task

A reference to a single module that defines the operations that Ansible performs.

Roles

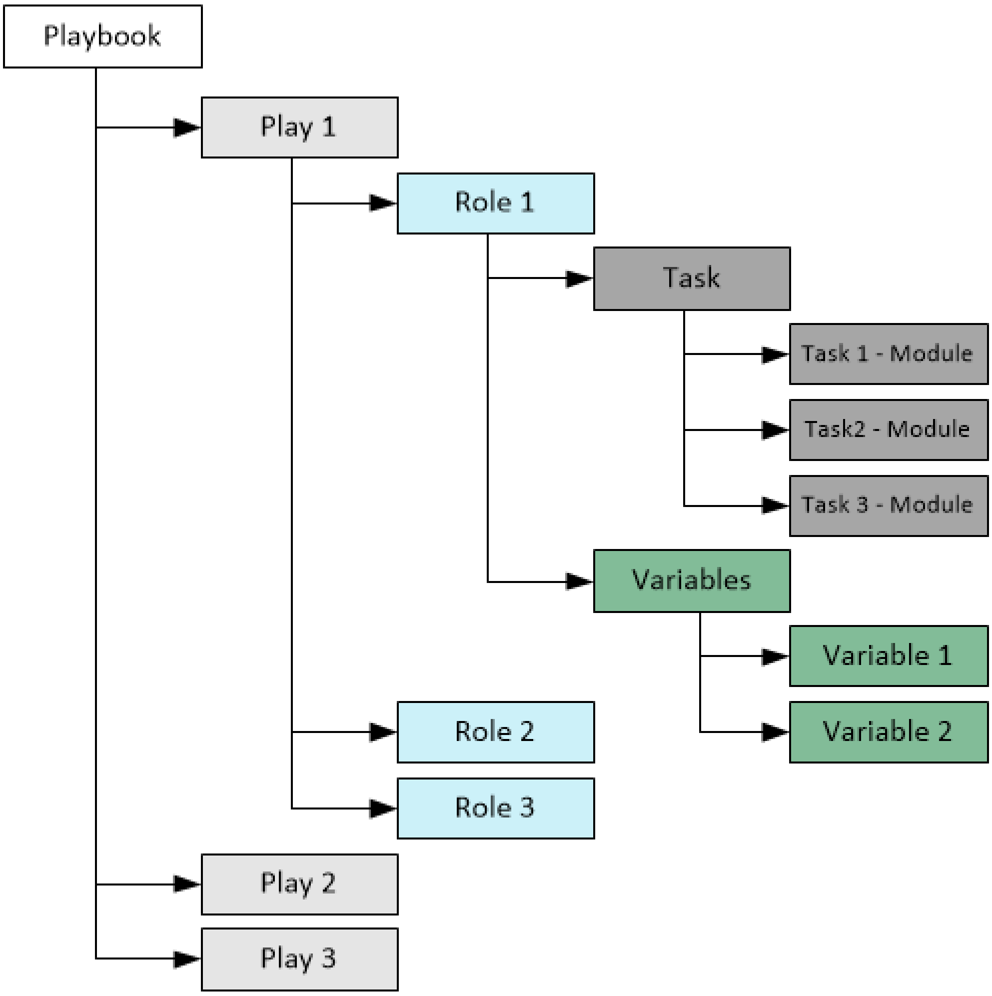
Roles are a way to make code in playbooks reusable by putting the functionality into "libraries" that can then be used in any playbook as needed.

Module

A unit of code or binary that Ansible runs on managed nodes.

Ansible modules are grouped in collections with a *Fully Qualified Collection Name* (FQCN) for each module. Tasks are executed by modules, each of which performs a specific task in a playbook. A module contains metadata that determines when and where a task is executed, and which user executes it. There are thousands of Ansible modules that perform all kinds of IT tasks, such as:

- Cloud management
- User management
- Networking
- Security
- Configuration management
- Communication



Get started as a platform administrator

As a platform administrator, Ansible Automation Platform helps you enable your users and teams to develop and run automation.

Learn the basic steps to get set up as an administrator for Ansible Automation Platform, including configuring and maintaining the platform for users.

To get started as an administrator, learn to:

- Log in for the first time.
- Configure authentication.
- Manage user access with role-based access control.

Log in for the first time

Log in to the Ansible Automation Platform as an administrator and enter your subscription information. You can then create user profiles and assign roles.

Procedure

1. With the login information provided after your installation completed, open a web browser and log in to Red Hat Ansible Automation Platform by navigating to the server URL at: `https://<AAP_SERVER_NAME>/`
2. Use the credentials specified during the installation process to login:
   - The default username is **admin**.
   - The password for **admin** is the value specified during installation.

Next steps

After your first login, you must add your subscription information to begin using the platform.

Add your subscription

To add your subscription information, you can either upload your subscription manifest, or use your service account credentials to find the subscription associated with your account.

Before you begin

To add your subscription by uploading a subscription manifest, you must first:

- Obtain your manifest file. See Obtaining a manifest file for more information.

To add your subscription using your service account credentials, you must first:

- Have created a service account and saved the client ID and client secret.
- Add your service account to the Subscription viewer user group to give it the ability to see your subscriptions. See the "Updates to subscription management" section in the Knowledgebase article Configure Ansible Automation Platform to authenticate through service account credentials for instructions on how to do so.

Procedure

1. To add your subscription by uploading a subscription manifest:
   a. Drag the file to the field beneath **Red Hat subscription manifest** or browse for the file on your local machine.
2. To add your subscription with your service account credentials:
   a. Click the **Service Account** tab.
   b. Enter the **client ID** you received when you created your service account in the field labeled Client ID.
   c. Enter the **client secret** you received when you created your service account in the field labeled Client secret.

   Your subscription appears in the **Subscription** list.
3. Select your subscription and click **Next**.
   a. Check the box indicating that you agree to the **End User License Agreement**.
   b. Review your information and click **Finish**.

If you enter your client ID and client secret but cannot locate your subscription, you might not have the correct permissions set on your service account. For more information and troubleshooting guidance for service accounts, see Configure Ansible Automation Platform to authenticate through service account credentials.

Related information
Attaching your Ansible Automation Platform subscription to your instance

Configure authentication

After your first login as an administrator, begin configuring authentication for your users. Depending on your organization’s needs and resources, you can either:

- Set up authentication by creating users, teams, and organizations, and then assigning them roles that govern access.
- Use an external source such as GitHub to configure authentication for your system.

The following sections serve as an introduction to authentication in Ansible Automation Platform.

Related information
Configure central authentication for Ansible Automation Platform
Managing access with role-based access control

Manage user access with role-based access control

Role-based access control (RBAC) restricts user access based on the user's role within the organization they are assigned to in Ansible Automation Platform. The roles in RBAC refer to the levels of access that users have to Ansible Automation Platform components and resources.

Use RBAC to control what users can do with the components of Ansible Automation Platform at a broad or granular level. You can choose whether the user is a system administrator or normal user, and align roles and access permissions with their positions within the organization.

You can define roles with multiple permissions that can then be assigned to resources, teams, and users. The permissions that make up a role govern what the assigned role allows. Permissions are allocated with only the access needed for a user to perform the tasks appropriate for their role.

The following procedures show how to get started with RBAC by creating a team, and a user to assign to the team.

> **IMPORTANT:**
> When managing users, teams, and organizations, use the Unified UI or the platform gateway API to ensure real-time synchronization across all platform components, including Event-Driven Ansible controller. If you use the legacy automation controller API, changes can take up to 15 minutes to propagate to Event-Driven Ansible controller, which can result in authentication errors for new users or teams.

Create an organization

Ansible Automation Platform automatically creates a default organization. If you have a self-support level license, you have only the default organization available and cannot delete it.

Procedure

1. From the navigation panel, select **Access Management › Organizations**.
2. Click **Create organization**.
3. Enter the **Name** and give a **Description** for your organization.

> **NOTE:**
> If automation controller is enabled on the platform, continue with Step 4. Otherwise, proceed to Step 6.

4. Select the name of the **Execution environment** or search for one that members of this organization can use to run automation.
5. Enter the name of the **Instance Groups** on which to run this organization.
6. Optional: Enter the **Galaxy credentials** or search from a list of existing ones.
7. Select the **Max hosts** for this organization. The default is 0. When this value is 0, it signifies no limit. If you try to add a host to an organization that has reached or exceeded its cap on hosts, an error message displays:

```
You have already reached the maximum number of 1 hosts allowed for your
organization. Contact your System Administrator for assistance.
```

8. Click **Next**.
9. If you selected more than 1 instance group, you can manage the order by dragging and dropping the instance group up or down in the list and clicking **Confirm**.

> **NOTE:**
> The execution precedence is determined by the order in which the instance groups are listed.

10. Click **Next** and verify the organization settings.
11. Click **Finish**.

Create a team

Manage teams by creating them, assigning an organization, and adding users or administrators. Team members automatically inherit all assigned roles and permissions. Users must exist in the system before they can be added to a team.

Procedure

1. From the navigation panel, select **Access Management › Teams**.
2. Click **Create team**.
3. Enter a **Name** and optionally give a **Description** for the team.
4. Select an **Organization** to be associated with this team.

> **NOTE:**
> Each team can only be assigned to one organization.

5. Click **Create team**. The **Details** page opens, where you can review and edit your team information and access.

Create a user

You can create three types of users in Ansible Automation Platform:

Normal user

Normal users have read and write access limited to the resources (such as inventory, projects, and job templates) for which that user has been granted the appropriate roles and privileges. Normal users are the default type of user when no other **User type** is specified.

Ansible Automation Platform Administrator

An administrator (also known as a Superuser) has full system administration privileges, with full read and write privileges over the entire installation. An administrator is typically responsible for managing all aspects of and delegating responsibilities for day-to-day work to various users.

Ansible Automation Platform Auditor

Auditors have read-only capability for all objects within the environment.

Procedure

1. From the navigation panel, select **Access Management > Users**.
2. Click **Create user**.
3. Enter the details about your new user in the fields on the **Create user** page. Fields marked with an asterisk (*) are required.
4. Normal users are the default when no **User type** is specified. To define a user as an administrator or auditor, select a **User type** from the drop-down menu.

> **NOTE:**
> If you are modifying your own password, log out and log back in for the change to take effect.

5. Select the **Organization** to be assigned for this user. For information about creating a new organization, see Creating an organization.
6. Click **Create user**.

   When the user is successfully created, the **User** details screen opens. From here, you can review and change the user's teams, roles, tokens and other membership details.

> **NOTE:**
> If the user is not newly-created, the details screen displays the user's last login activity.

Next steps

If you log in as yourself, and view the details of your user profile, you can manage tokens from your user profile by selecting the **Tokens** tab. For more information, see Adding a token.

Get started as an automation developer

As an automation developer, you can use Ansible Automation Platform to implement your organization's automation strategy.

Ansible Automation Platform can help you write, test, and share automation content, and download and use Red Hat certified collections. Learn the basic workflow to get set up as an automation developer on Ansible Automation Platform, including how to:

- Set up your development environment
- Create, publish, and use custom automation content
- Build and use automation execution environments and decision environments
- Create and run rulebook activations for Event-Driven Ansible
- Create and use job templates

Set up your development environment

Before you begin to create content, consult our guidance on developing automation content. There you can find information on Ansible development tools, which you can integrate into your environment, and learn how to scaffold a playbook project.

Related information
Create, test, and deploy automation content with ansible-dev-tools

Create automation content with playbooks

Ansible playbooks are blueprints that tell Ansible Automation Platform what tasks to perform with which devices. Use a playbook to define automation tasks for the platform to run.

Create a playbook

A playbook contains one or more plays. A basic play contains the following parameters:

- **Name**: a brief description of the overall function of the playbook, which assists in keeping it readable and organized for all users.
- **Hosts**: identifies the target or targets for Ansible to run against.
- **Become statements**: this optional statement can be set to `true` or `yes` to enable privilege escalation using a become plugin (such as `sudo`, `su`, `pfexec`, `doas`, `pbrun`, `dzdo`, `ksu`).
- **Tasks**: this is the list of actions that get executed against each host in the play.

Here is an example of a play in a playbook. You can see the name of the play, the host, and the list of tasks included in the play.

```
- name: Set Up a Project and Job Template
  hosts: host.name.ip
  become: true

  tasks:
    - name: Create a Project
      ansible.controller.project:
        name: Job Template Test Project
        state: present
        scm_type: git
        scm_url: https://github.com/ansible/ansible-tower-samples.git

    - name: Create a Job Template
      ansible.controller.job_template:
        name: my-job-1
        project: Job Template Test Project
        inventory: Demo Inventory
        playbook: hello_world.yml
        job_type: run
        state: present
```

For more detailed guidance on authoring playbooks, consult the following documentation:

- Developing automation content

- Creating playbooks and viewing playbook explanations in the Red Hat Ansible Lightspeed with IBM watsonx Code Assistant User Guide to learn how to generate a playbook with AI.
- Getting started with playbooks

Related information
Create, test, and deploy automation content with ansible-dev-tools
Create playbooks and view playbook explanations
Get started automating with playbooks

Write a playbook

Create a playbook that pings your hosts and prints a "Hello world" message.

Ansible uses the YAML syntax. YAML is a human-readable language that enables you to create playbooks without having to learn a complicated coding language.

Procedure

1. Create a file named `playbook.yaml` in your `ansible_quickstart` directory, with the following content:

```
- name: My first play
  hosts: myhosts
  tasks:
    - name: Ping my hosts
    ansible.builtin.ping:

    - name: Print message
    ansible.builtin.debug:
      msg: Hello world
```

2. Run your playbook:

```
$ ansible-playbook -i inventory.ini playbook.yaml
```

Result

Ansible returns the following output:

```
PLAY [My first play] ********************************************************

TASK [Gathering Facts] ******************************************************
ok: [192.0.2.50]
ok: [192.0.2.51]
ok: [192.0.2.52]

TASK [Ping my hosts] ********************************************************
ok: [192.0.2.50]
ok: [192.0.2.51]
ok: [192.0.2.52]

TASK [Print message] ********************************************************
ok: [192.0.2.50] => {
    "msg": "Hello world"
}
ok: [192.0.2.51] => {

    "msg": "Hello world"
}
ok: [192.0.2.52] => {
    "msg": "Hello world"
}

PLAY RECAP ******************************************************************
192.0.2.50: ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
   ignored=0
192.0.2.51: ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
   ignored=0
192.0.2.52: ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
   ignored=0
```

Related information
[Getting started with playbooks]
[Red Hat Ansible Lightspeed with IBM watsonx Code Assistant]

Define events with rulebooks

An Ansible rulebook is a collection of rulesets that references one or more sources, rules, and conditions.

Rulebooks are to Event-Driven Ansible what playbooks are to Ansible Automation Platform as a whole. Like a playbook, a rulebook defines automation tasks for the platform, along with when they should be triggered.

Rulebook actions

Rulebooks use an "if-this-then-that" logic that tells Event-Driven Ansible what actions to activate when a rule is triggered. Event-Driven Ansible listens to the controller event stream and, when an event triggers a rule, activates an automation action in response.

Rulebooks can trigger the following activations:

- `run_job_template`
- `run_playbook` (only supported with ansible-rulebook CLI)
- `debug`
- `print_event`
- `set_fact`
- `post_event`
- `retract_fact`
- `shutdown`

Related information
[Actions]
[Automation decisions]

Package and reuse content with roles

An role is a portable, self-contained unit of automation that bundles tasks, variables, templates, and handlers. Roles allow you to organize and reuse automation content to fit specific system needs.

Instead of creating huge playbooks with hundreds of tasks, you can use roles to break the tasks apart into smaller, more discrete units of work.

Related information
What is an Ansible Role—and how is it used?

Create a role

Create an Ansible role by using the `ansible-galaxy` CLI tool to build reusable automation components.

The Ansible Automation Platform bundle includes the Ansible Galaxy CLI tool. Access role-specific commands from the `role` subcommand:

```
ansible-galaxy role init <role_name>
```

Standalone roles outside of Collections are supported. Create new roles inside a Collection to take advantage of the features Ansible Automation Platform has to offer.

Procedure

1. In a terminal, navigate to the `roles` directory inside a collection.
2. Create a role called `my_role` inside the collection:

```
$ ansible-galaxy role init my_role
```

The collection now includes a role named `my_role` inside the `roles` directory, as you can see in this example:

```
~/.ansible/collections/ansible_collections/<my_namespace>/<my_collection_name>
    ...
    └── roles/
        └── my_role/
            ├── .travis.yml
            ├── README.md
            ├── defaults/
            │   └── main.yml
            ├── files/
            ├── handlers/
            │   └── main.yml
            ├── meta/
            │   └── main.yml
            ├── tasks/
            │   └── main.yml
            ├── templates/
            ├── tests/
            │   ├── inventory
            │   └── test.yml
            └── vars/
                └── main.yml
```

3. A custom role skeleton directory can be supplied by using the `--role-skeleton` argument. This allows organizations to create standardized templates for new roles to suit their needs.

```
$ ansible-galaxy role init my_role --role-skeleton ~/role_skeleton
```

This creates a role named `my_role` by copying the contents of `~/role_skeleton` into `my_role`. The contents of `role_skeleton` can be any files or folders that are valid inside a role directory.

Package and distribute automation content with collections

Collections are a distribution format for Ansible content that can include playbooks, roles, modules, and plugins. Red Hat provides Ansible Content Collections on Ansible automation hub that contain both Red Hat Ansible Certified Content and Ansible validated content.

If you have installed private automation hub, you can create collections for your organization and push them to private automation hub so that you can use them in job templates in Ansible Automation Platform. You can use collections to package and distribute plug-ins. These plug-ins are written in Python.

You can also create collections to package and distribute Ansible roles, which are expressed in YAML. You can also include playbooks and custom plug-ins that are required for these roles in the collection. Typically, collections of roles are distributed for use within your organization.

Publish to a collection

You can configure your projects to be uploaded to Git, or to the source control manager of your choice.

Procedure

1. From the navigation panel, select **Automation Execution › Projects**.
2. Locate or create the project that you want to publish to your source control manager.
3. In the project **Details** tab, select **Edit project**.
4. Select **Git** from the **Source Control Type** drop-down menu.
5. Enter the appropriate details into the following fields:
    a. **Source Control URL** - see an example in the tooltip.
    b. Optional: **Source control branch/tag/commit**: Enter the SCM branch, tags, commit hashes, arbitrary refs, or revision number (if applicable) from the source control to checkout. Some commit hashes and references might not be available unless you also provide a custom refspec in the next field. If left blank, the default is `HEAD`, which is the last checked out branch, tag, or commit for this project.
    c. **Source Control Refspec** - This field is an option specific to Git source control and only advanced users familiar and comfortable with Git should specify which references to download from the remote repository. For more information, see Job branch overriding.
    d. **Source Control Credential** - If authentication is required, select the appropriate source control credential.

6. Optional: **Options** - select the launch behavior, if applicable:
    a. **Clean** - Removes any local modifications before performing an update.
    b. **Delete** - Deletes the local repository in its entirety before performing an update. Depending on the size of the repository this can significantly increase the amount of time required to complete an update.
    c. **Track submodules** - Tracks the latest commit. See the tooltip for more information.
    d. **Update Revision on Launch** - Updates the revision of the project to the current revision in the remote source control, and caches the roles directory from Collections support. Automation controller ensures that the local revision matches and that the roles and collections are up-to-date with the last update. In addition, to avoid job overflows if jobs are spawned faster than the project can synchronize, selecting this enables you to configure a cache timeout to cache previous project synchronizations for a given number of seconds.
    e. **Allow Branch Override** - Enables a job template or an inventory source that uses this project to start with a specified SCM branch or revision other than that of the project. For more information, see Job branch overriding.
7. Click **Save** to save your project.

Upload a collection to automation hub

If you want to share a collection that you have created with the rest of the Ansible community, you can upload it to automation hub.

Before you begin

- You have configured the `ansible-galaxy` client for automation hub.
- You have at least one namespace.
- You have run all content through `ansible-test sanity`

NOTE:
Sharing a collection with the Ansible community requires getting the collection certified or validated by our Partner Engineering team. This action is available only to partner clients. For more about becoming a partner, see our documentation on software certification.

You can upload your collection by using either the automation hub user interface or the `ansible-galaxy` client.

Procedure

1. From the navigation panel, select **Automation Content > Namespaces**.
2. Within the My namespaces tab, locate and click into the namespace to which you want to upload a collection.

3. Select the **Collections** tab, and then click **Upload collection**.
4. In the New collection modal, click **Select file**. Locate the file on your system.
5. Click **Upload**.
6. Optional: you can also upload from the command line. Using the `ansible-galaxy` client, enter the following command:

```
$ ansible-galaxy collection publish path/to/my_namespace-
my_collection-1.0.0.tar.gz --api-key=SECRET
```

Create automation runtimes with execution and decision environments

All automation in Red Hat Ansible Automation Platform runs on container images called automation execution environments.

Automation execution environments are consistent and shareable container images that serve as Ansible control nodes. Automation execution environments reduce the challenge of sharing Ansible content that has external dependencies.

If automation content is similar to a script that a developer has written, an automation execution environment is like a replica of that developer's environment, thereby enabling you to reproduce and scale that automation content. In this way, execution environments make it easier for you to implement automation in a range of environments.

Automation execution environments contain:

- Ansible Core
- Ansible Runner
- Ansible Collections
- Python libraries
- System dependencies
- Custom user needs

You can either use the default base execution environment included in your Ansible Automation Platform subscription, or you can define and create an automation execution environment using Ansible Builder.

Use the base automation execution environment

Ansible Automation Platform provides access to some base automation execution environments. You can use a base execution environment as a starting point for creating a customized execution environment.

Before you begin

- You have a valid Red Hat Ansible Automation Platform subscription.

Ansible Automation Platform includes the following execution environments:

- `Minimal` - Includes the latest Ansible-core 2.15 release along with Ansible Runner, but does not include collections or other content
- `EE Supported` - Minimal, plus all Red Hat-supported collections and dependencies

Base images included with Ansible Automation Platform are hosted on the Red Hat Ecosystem Catalog (`registry.redhat.io`).

Procedure

1. Log in to `registry.redhat.io`.

```
$ podman login registry.redhat.io
```

2. Pull the base images from the registry:

```
$ podman pull registry.redhat.io/aap/<image name>
```

Related information
Install Ansible Builder to create or edit execution environments

Add an execution environment to a job template

After you have built an execution environment, use it to run jobs. To do so, first associate the execution environment with a job template.

Before you begin

- An execution environment created using `ansible-builder` as described in Define, create, and build execution environments.
- Organization administrator privileges (if the execution environment is associated with an organization).
- A credential with a username, host, and password (if assigned to the execution environment).

Use the following procedure to add an execution environment to a job template.

Procedure

1. From the navigation panel, select **Automation Execution › Infrastructure › Execution Environments**.
2. Click **Create execution environment** to create an execution environment.
3. Enter the appropriate details into the following fields:
    a. **Name** (required): Enter a name for the execution environment.
    b. **Image** (required): Enter the image name. The image name requires its full location (repository), the registry, image name, and version tag, as in the following example: `quay.io/ansible/awx-ee:latestrepo/project/image-name:tag`
    c. Optional: **Pull**: Choose the type of pull when running jobs:
        i. **Always pull container before running**: Pulls the latest image file for the container.
        ii. **Only pull the image if not present before running**: Only pulls the latest image if none are specified.
        iii. **Never pull container before running**: Never pull the latest version of the container image.

        NOTE: If you do not set a type for pull, the value defaults to **Only pull the image if not present before running**.

    d. Optional: **Description**: Enter an optional description.
    e. Optional: **Organization**: Assign the organization to specifically use this execution environment. To make the execution environment available for use across multiple organizations, leave this field blank.
    f. **Registry credential**: If the image has a protected container registry, provide the credential to access it.
4. Click **Create execution environment**. Your newly added execution environment is ready to be used in a job template.
5. To add an execution environment to a job template, navigate to **Automation Execution › Templates** and select your template.

a. Click **Edit template** and specify your execution environment in the field labeled **execution environment**.

Result

After you add an execution environment to a job template, the template is listed in the **Templates** tab in your execution environment details.

About container registries

If you have many automation execution environments that you want to support, you can store them in a container registry linked to your private automation hub.

For more information, see Populating your private automation hub container registry.

Build and use a decision environment

Event-Driven Ansible includes an ansible.eda collection, which contains sample sources, event filters and rulebooks. All the collections, ansible rulebooks and their dependencies use a decision environment, which is an image that can be run on either Podman or Kubernetes.

In decision environments, sources, which are typically Python code, are distributed through ansible-collections. They inject external events into a rulebook for processing. The rulebook consists of the following:

- The python interpreter
- Java Runtime Environment for Drools rule engine
- ansible-rulebook python package
- ansible.eda collection

You can use the base decision environment and build your own customized Decision Environments with additional collections and collection dependencies. You can build a decision environment using a Dockerfile or optionally you can deploy your CA certificate into the image.

Set up a new decision environment

The following steps describe how to import a decision environment into the platform.

Before you begin

- You have set up any necessary credentials. For more information, see Setting up credentials.
- You have pushed a decision environment image to an image repository or you chose to use the image `de-supported` provided at registry.redhat.io.

Procedure

1. Navigate to **Automation Decisions › Decision Environments**.
2. Click **Create decision environment**.
3. Enter the following:

   Name

   Insert the name.

   Description

   This field is optional.

   Image

   This is the full image location, including the container registry, image name, and version tag.

   Credential

   This field is optional. This is the token needed to use the decision environment image.
4. Select **Create decision environment**.

Result

Your decision environment is now created and can be managed on the **Decision Environments** page.

After saving the new decision environment, the decision environment's details page is displayed. From there or the **Decision Environments** list view, you can edit or delete it.

Organize and define automation sources and targets

Projects and inventory files help you organize your automation sources and targets.

Create an automation execution project

A project is a logical collection of playbooks. Projects are useful as a way to group your automation content according to the organizing principle of your choice.

You can set up an automation execution project in the platform UI.

Procedure

1. From the navigation panel, select **Automation Execution > Projects**.
2. On the **Projects** page, click **Create project** to launch the **Create Project** window.
3. Enter the appropriate details into the following required fields:
    - **Name** (required)
    - Optional: **Description**
    - **Organization** (required): A project must have at least one organization. Select one organization now to create the project. When the project is created you can add additional organizations.
    - Optional: **Execution Environment**: Enter the name of the execution environment or search from a list of existing ones to run this project.
    - **Source Control Type** (required): Select an SCM type associated with this project from the menu. Options in the following sections become available depending on the type chosen. For more information, see Managing playbooks manually or Managing playbooks using source control.
    - Optional: **Content Signature Validation Credential**: Use this field to enable content verification. Specify the GPG key to use for validating content signature during project synchronization. If the content has been tampered with, the job will not run. For more information, see Project signing and verification.
4. Click **Create project**.

Create an automation decision project

Like automation execution projects, automation decision projects are logical collections of automation decision content. You can use the project function to organize your automation decision content in a way that makes sense to you.

Before you begin

- You have set up any neccessary credentials. For more information, see Setting up credentials.
- You have an existing repository containing rulebooks that are integrated with playbooks contained in a repository to be used by automation controller.

Procedure

1. From the navigation panel, select **Automation Decisions > Projects**.
2. Click **Create project**.

3. Enter the following information:

    - **Name**: Enter project name.
    - **Description**: This field is optional.
    - **Organization**: Select the organization associated with the project.
    - **Source control type**: Git is the only SCM type available for use.
    - **Proxy**: Proxy used to access HTTP or HTTPS servers.
    - **Source control branch/tag/commit**: Branch to checkout. Can also be tags, commit hashes, or arbitrary refs.
    - **Source control refspec**: A refspec to fetch. This parameter allows access to references through the branch field not otherwise available.
    - Optional: **Source control credential**: The token needed to use the source control URL.
    - **Content signature validation credential**: Enables content signing to verify that the content has remained secure during project syncing. If the content is tampered with, the job will not run.
    - **Options**: Checking the box next to **Verify SSL** verifies the SSL with HTTPS when the project is imported.

4. Click **Create project**.

Next steps

Your project is now created and can be managed in the **Projects** screen.

After saving the new project, the project's details page is displayed. From there or the **Projects** list view, you can edit or delete it.

About inventories

An inventory is a file listing the collection of hosts managed by Ansible Automation Platform.

Organizations are assigned to inventories, while permissions to launch playbooks against inventories are controlled at the user or team level.

You can find inventories in the UI by navigating to **Automation Execution > Infrastructure > Inventories**. The Inventories window displays a list of the inventories that are currently available. You can sort the inventory list by name and search by inventory type, organization, description, inventory creators or modifiers, or additional criteria. Use the following procedure to create a new inventory.

Procedure

1. From the navigation panel, select **Automation Execution › Infrastructure › Inventories**. The **Inventories** view displays a list of the inventories currently available.
2. Click **Create inventory**, and from the list menu select the type of inventory you want to create.
3. Enter the appropriate details into the following fields:
    - **Name**: Enter a name for the inventory.
    - Optional: **Description**: Enter a description.
    - **Organization**: Choose among the available organizations.
    - Only applicable to Smart Inventories: **Smart Host Filter**: Filters are similar to tags in that tags are used to filter certain hosts that contain those names. Therefore, to populate this field, specify a tag that contains the hosts you want, not the hosts themselves. Filters are case-sensitive. For more information, see Smart host filters.
    - **Instance groups**: Select the instance group or groups for this inventory to run on. If the list is extensive, use the search to narrow the options. You can select multiple instance groups and sort them in the order that you want them run.
    - Optional: **Labels**: Add labels that describe this inventory, so they can be used to group and filter inventories and jobs.
    - Only applicable to constructed inventories: **Input inventories**: Specify the source inventories to include in this constructed inventory. Empty groups from input inventories are copied into the constructed inventory.
    - Optional and only applicable to constructed inventories: **Cache timeout (seconds)**:Set the length of time you want the cache plugin data to timeout.
    - Only applicable to constructed inventories: **Verbosity**: Control the level of output that Ansible produces as the playbook executes related to inventory sources associated with constructed inventories. Select the verbosity from Normal to various Verbose or Debug settings. This only appears in the "details" report view.
        - Verbose logging includes the output of all commands.
        - Debug logging is exceedingly verbose and includes information on SSH operations that can be useful in certain support instances. Most users do not need to see debug mode output.
    - Only applicable to constructed inventories: **Limit**: Restricts the number of returned hosts for the inventory source associated with the constructed inventory. You can paste a group name into the limit field to only include hosts in that group. For more information, see the **Source variables** setting.
    - Only applicable to standard inventories: **Options**: Check the box next to **Prevent instance group fallback** to enable only the instance groups listed in the **Instance groups** field to execute the job. If unchecked, all available instances in the execution pool will be used based on the hierarchy described in Control where a job runs . Click the tooltip for more information.

> **NOTE:** Set the `prevent_instance_group_fallback` option for smart inventories through the API.

- **Variables** (**Source variables** for constructed inventories):
    - **Variables**: Variable definitions and values to apply to all hosts in this inventory. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.
    - **Source variables** for constructed inventories are used to configure the constructed inventory plugin. Source variables create groups under the `groups` data key. The variable accepts Jinja2 template syntax, renders it for every host, makes a `true` or `false` evaluation, and includes the host in the group (from the key of the entry) if the result is `true`.

4. Click **Create inventory**.

Next steps

After creating the new inventory, you can proceed with configuring permissions, groups, hosts, sources, and viewing completed jobs, if applicable to the type of inventory.

Reliably run playbooks with job templates

A job template is a definition and set of parameters for running an Ansible job.

A job template combines an Ansible playbook from a project and the settings required to launch it, including information about the target host against which the playbook will run, authentication information to access the host, and any other relevant variables . Job templates are useful to run the same job many times. Job templates also encourage the reuse of Ansible playbook content and collaboration between teams.

Related information
Standardize and streamline automation with automation job templates

Get started with job templates

Job templates are definitions and sets of parameters for running an Ansible job. A job template ties together all the necessary information to launch an Ansible job.

As part of the initial setup, a **Demo Job Template** is created for you.

Procedure

1. To review existing templates, select **Automation Execution > Templates** from the navigation panel.

2. Click **Demo Job Template** to view its details.

Set your domains of interest

With Domain filtering lets you customize content in Automation Execution's **Jobs** and **Templates** sections. Jobs and templates are linked to labels; selecting one filters out less-relevant items, providing easy access to resources tailored to your specific area of interest.

Procedure

1. From the navigation panel, select **Automation Execution › Jobs** or **Automation Execution › Templates**.
2. Beneath the page heading, next to **Domains**, is a list of topic-related labels. Select a label to filter jobs and job templates so that only content related to the labels is shown. You can choose more than one label.
3. To clear your selection, click the **X**.
4. To customize your domain options, select the icon. In the modal that appears, select **Add Domain** to add new domains to filter with.

Next steps

You can add labels to your individual job templates to make the templates appear as resources when you filter with the related domain label. Go to **Automation Execution › Templates**, select your job template, and click **Edit template**. On the editing screen, enter the label you want to use in the **Labels** field and click **Save job template**.

Create a job template

Use the following procedure to create a job template.

Procedure

1. From the navigation panel, select **Automation Execution › Templates**.
2. On the **Templates** page, select **Create job template** from the **Create template** list.
3. Enter the appropriate details in the following fields:

> **NOTE:**
> If a field has the **Prompt on launch** checkbox selected, launching the job prompts you for the value for that field when launching.
>
> Most prompted values override any values set in the job template.
>
> Exceptions are noted in the following table.

Field	Options	Prompt on Launch
Name	Enter a name for the job.	N/A
Description	Enter an arbitrary description as appropriate (optional).	N/A
Job Type	Choose a job type: • Run: Start the playbook when launched, running Ansible tasks on the selected hosts. • Check: Perform a "dry run" of the playbook and report changes that would be made without actually making them. Tasks that do not support check mode are missed and do not report potential changes.	Yes
Inventory	Choose the inventory to use with this job template from the inventories available to the logged in user. A System Administrator must grant you or your team permissions to be	Yes. Inventory prompts show up as its own step in a later prompt window.

Field	Options	Prompt on Launch
	able to use certain inventories in a job template.	
Project	Select the project to use with this job template from the projects available to the user that is logged in.	N/A
Source control branch	This field is only present if you chose a project that allows branch override. Specify the overriding branch to use in your job run. If left blank, the specified SCM branch (or commit hash or tag) from the project is used. For more information, see Job branch overriding.	Yes
Execution Environment	Select the container image to be used to run this job. You must select a project before you can select an execution environment.	Yes. Execution environment prompts show up as its own step in a later prompt window.
Playbook	Choose the playbook to be launched with this job template from the available playbooks. This field automatically populates with the names of the playbooks found in the project base path for the selected project. Alternatively, you can enter the name of the playbook if it is not listed, such as the name of a file (such as foo.yml) you want to use to run with that playbook. If you	N/A

Field	Options	Prompt on Launch
	enter a filename that is not valid, the template displays an error, or causes the job to fail.	
Credentials	Select the icon to open a separate window. Choose the credential from the available options to use with this job template. Use the drop-down menu list to filter by credential type if the list is extensive. Some credential types are not listed because they do not apply to certain job templates.	• If selected, when launching a job template that has a default credential and supplying another credential replaces the default credential if it is the same type. The following is an example of this message: `Job Template default credentials must be replaced with one of the same type. Please select a credential for the following types in order to proceed: Machine.` • You can add more credentials as you see fit. • Credential prompts show up as its own step in a later prompt window.
Labels	• Optionally supply labels that describe this job template, such as `dev` or `test`. • Use labels to group and filter job templates and	• If selected, even if a default value is supplied, you are prompted when launching to supply additional labels, if needed. • You cannot delete existing labels, selecting only

Field	Options	Prompt on Launch
	completed jobs in the display. • Labels are created when they are added to the job template. Labels are associated with a single Organization by using the Project that is provided in the job template. Members of the Organization can create labels on a job template if they have edit permissions (such as the admin role). • Once you save the job template, the labels appear in the **Job Templates** overview in the Expanded view. • Select ✖ beside a label to remove it. When a label is removed, it is no longer associated with that particular Job or Job Template, but it remains associated with any other jobs that reference it. • Jobs inherit labels from the Job Template at the time of launch. If you delete a label from a Job Template, it is also	removes the newly added labels, not existing default labels.

Field	Options	Prompt on Launch
	deleted from the Job.	
Forks	The number of parallel or simultaneous processes to use while executing the playbook. A value of zero uses the Ansible default setting, which is five parallel processes unless overridden in `/etc/ansible/ansible.cfg`.	Yes
Limit	A host pattern to further constrain the list of hosts managed or affected by the playbook. You can separate many patterns by colons (:). As with core Ansible: • a:b means "in group a or b" • a:b:&c means "in a or b but must be in c" • a:!b means "in a, and definitely not in b"	Yes If not selected, the job template executes against all nodes in the inventory or only the nodes predefined on the **Limit** field. When running as part of a workflow, the workflow job template limit is used instead.
Verbosity	Control the level of output Ansible produces as the playbook executes. Choose the verbosity from Normal to various Verbose or Debug settings. This only appears in the **details** report view. Verbose logging includes the output of all commands. Debug logging is exceedingly verbose and includes information	Yes

Field	Options	Prompt on Launch
	about SSH operations that can be useful in certain support instances. Verbosity `5` causes automation controller to block heavily when jobs are running, which could delay reporting that the job has finished (even though it has) and can cause the browser tab to lock up.	
Job Slicing	Specify the number of slices you want this job template to run. Each slice runs the same tasks against a part of the inventory. For more information about job slices, see Job Slicing.	Yes
Timeout	This enables you to specify the length of time (in seconds) that the job can run before it is canceled. Consider the following for setting the timeout value:	Yes

Field	Options	Prompt on Launch
	• There is a global timeout defined in the settings which defaults to 0, indicating no timeout. • A negative timeout (<0) on a job template is a true "no timeout" on the job. • A timeout of 0 on a job template defaults the job to the global timeout (which is no timeout by default). • A positive timeout sets the timeout for that job template.	
Show Changes	Enables you to see the changes made by Ansible tasks.	Yes
Instance Groups	Choose Instance and Container Groups to associate with this job template. If the list is extensive, use the icon to narrow the options. Job template instance groups contribute to the job scheduling criteria, see Job Runtime Behavior and Control where a job runs for rules. A System Administrator must grant you or your team permissions to be able to use an instance group in a job template. Use of a	• Yes. If selected, you are providing the jobs preferred instance groups in order of preference. If the first group is out of capacity, later groups in the list are considered until one with capacity is available, at which point that is selected to run the job. • If you prompt for an instance group, what you enter replaces the

Field	Options	Prompt on Launch
	container group requires admin rights.	normal instance group hierarchy and overrides all of the organizations' and inventories' instance groups. • The Instance Groups prompt shows up as its own step in a later prompt window.
Job Tags	Type and select the **Create** menu to specify which parts of the playbook should be executed.	Yes
Skip Tags	Type and select the **Create** menu to specify certain tasks or parts of the playbook to skip.	Yes
Extra Variables	• Pass extra command line variables to the playbook. This is the "-e" or "-extra-vars" command line parameter for ansible-playbook that is documented in the Ansible documentation at Defining variables at runtime. • Provide key or value pairs by using either YAML or JSON. These variables have a maximum value of precedence and overrides other	Yes. If you want to be able to specify `extra_vars` on a schedule, you must select **Prompt on launch** for Variables on the job template, or enable a survey on the job template. Those answered survey questions become `extra_vars`.

Field	Options	Prompt on Launch
	variables specified elsewhere. The following is an example value: `git_branch: production release_version: 1.5`	

4. You can set the following options for launching this template, if necessary:

    - **Privilege escalation**: If checked, you enable this playbook to run as an administrator. This is the equal of passing the `--become` option to the `ansible-playbook` command.
    - **Provisioning callback**: If checked, you enable a host to call back to automation controller through the REST API and start a job from this job template. For more information, see Provisioning Callbacks.
    - **Enable webhook**: If checked, you turn on the ability to interface with a predefined SCM system web service that is used to launch a job template. GitHub and GitLab are the supported SCM systems. If you enable webhooks, other fields display, prompting for additional information:
        - **Webhook service**: Select which service to listen for webhooks from.
        - **Webhook URL**: Automatically populated with the URL for the webhook service to POST requests to.
        - **Webhook key**: Generated shared secret to be used by the webhook service to sign payloads sent to automation controller. You must configure this in the settings on the webhook service in order for automation controller to accept webhooks from this service.
        - **Webhook credential**: Optionally, give a GitHub or GitLab personal access token (PAT) as a credential to use to send status updates back to the webhook service.
          Before you can select it, the credential must exist.

          See Credential Types to create one.
        - For additional information about setting up webhooks, see Working with Webhooks.
    - **Concurrent jobs**: If checked, you are allowing jobs in the queue to run simultaneously if not dependent on one another. Check this box if you want to run job slices simultaneously. For more information, see Automation controller capacity determination and job impact.

- **Enable fact storage**: If checked, automation controller stores gathered facts for all hosts in an inventory related to the job running.

- **Prevent instance group fallback**: Check this option to allow only the instance groups listed in the **Instance Groups** field to run the job. If clear, all available instances in the execution pool are used based on the hierarchy described in Control where a job runs.

5. Click **Create job template**, when you have completed configuring the details of the job template.

   Creating the template does not exit the job template page but advances to the Job Template **Details** tab. After saving the template, you can click **Launch template** to start the job. You can also click **Edit** to add or change the attributes of the template, such as permissions, notifications, view completed jobs, and add a survey (if the job type is not a scan). You must first save the template before launching, otherwise, **Launch template** remains disabled.

Result

1. From the navigation panel, select **Automation Execution › Templates**.

2. Verify that the newly created template appears on the **Templates** page.

Edit a job template

As part of the initial setup, you can leave the default **Demo Job Template** as it is, but you can edit it later.

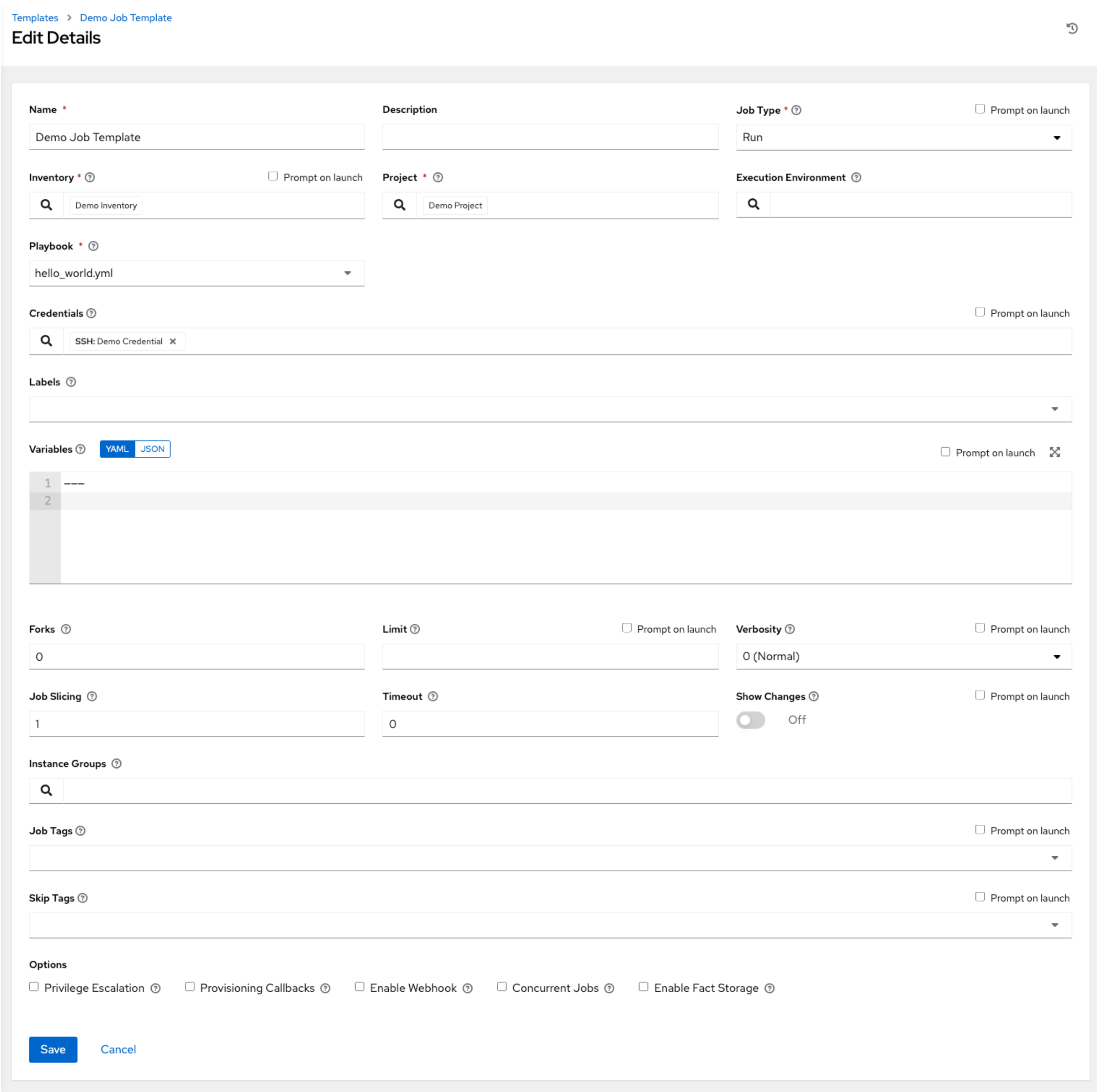
Procedure

1. Open the template to edit it by using one of these methods:

   - Click **Edit** in the job template Details page.

   - From the navigation panel, select **Automation Execution › Templates**. Click **Edit** next to the template name and edit the appropriate details.

2. Save your changes.



3. To exit after saving and return to the **Templates** list view, use the breadcrumb navigation links or click **Cancel**. Clicking **Save** does not exit the **Details** dialog.

Configure rulebooks to take action in response to events or conditions

In Event-Driven Ansible, a rulebook activation is a process running in the background defined by a decision environment executing a specific rulebook.

Set up a rulebook activation

Follow the steps in this procedure to set up a rulebook activation.

Before you begin

- You have set up a project.
- You have set up a decision environment.

Procedure

1. From the navigation panel, select **Automation Decisions › Rulebook Activations**.
2. Click **Create rulebook activation**.
3. Enter the following information:
    - **Name**: Insert the name.
    - **Description**: This field is optional.
    - **Organization**: This field is optional.
    - **Project**: This field is optional.
    - **Rulebook**: Rulebooks are displayed according to the project you selected.
    - **Credential**: Select 0 or more credentials for this rulebook activation. This field is optional.

      NOTE: The credentials that display in this field are customized based on your rulebook activation and only include the following credential types: Vault, Red Hat Ansible Automation Platform, or any custom credential types that you have created. For more information about credentials, see Credentials .

    - **Decision environment**: A decision environment is a container image to run Ansible rulebooks.

      NOTE: In Event-Driven Ansible controller, you cannot customize the pull policy of the decision environment. By default, it follows the behavior of the **always** policy. Every time an activation is started, the system tries to pull the most recent version of the image.

    - **Restart policy**: This is the policy that determines how an activation should restart after the container process running the source plugin ends. Select from the following options:
        - **Always**: This restarts the rulebook activation immediately, regardless of whether it ends successfully or not, and occurs no more than 5 times.

- **Never**: This never restarts a rulebook activation when the container process ends.
- **On failure**: This restarts the rulebook activation after 60 seconds by default, only when the container process fails, and occurs no more than 5 times.

- **Log level**: This field defines the severity and type of content in your logged events. Select from one of the following options:
  - **Error**: Logs that contain error messages that are displayed in the **History** tab of an activation.
  - **Info**: Logs that contain useful information about rulebook activations, such as a success or failure, triggered action names and their related action events, and errors.
  - **Debug**: Logs that contain information that is only useful during the debug phase and might be of little value during production. This log level includes both error and log level data.
- **Service name**: This defines a service name for Kubernetes to configure inbound connections if the activation exposes a port. This field is optional.
- **Rulebook activation enabled?**: Toggle to automatically enable the rulebook activation to run.
- **Variables**: The variables for the rulebook are in JSON or YAML format. The content would be equivalent to the file passed through the `--vars` flag of ansible-rulebook command.
- **Options**: Check the **Skip audit events** option if you do not want to see your events in the Rule Audit.

4. Click **Create rulebook activation**.

Result

Your rulebook activation is now created and can be managed on the **Rulebook Activations** page.

> **NOTE:** Occasionally, when a source plugin shuts down, it causes a rulebook to exit gracefully after a certain amount of time. When a rulebook activation shuts down, any tasks that are waiting to be performed will be canceled, and an info level message is sent to the activation log. For more information, see Rulebooks.

After saving the new rulebook activation, the rulebook activation’s details page is displayed, with either a **Pending**, **Running**, or **Failed** status. From there or the **Rulebook Activations** list view, you can restart or delete it.

Rulebook activation list view

Use the **Rulebook Activations** list view to quickly monitor the operational status, event fire count, and restart frequency of all your deployed automation services.

If the **Status** is **Running**, it means that the rulebook activation is running in the background and executing the required actions according to the rules declared in the rulebook.

You can view more details by selecting the activation from the **Rulebook Activations** list view.

For all activations that have run, you can view the **Details** and **History** tabs to get more information about what happened.

Enable and disable rulebook activations

Toggle the state of an activation to start or stop event processing instantly, allowing for temporary pauses, maintenance, or troubleshooting without deletion.

Procedure

1. Select the switch on the row level to enable or disable your chosen rulebook.
2. In the window, select **Yes, I confirm that I want to enable/disable these X rulebook activations**.
3. Select **Enable/Disable rulebook activation**.

Restart rulebook activations

Restart an activation to immediately apply configuration changes, update rulebook content, or quickly recover from unexpected failures.

NOTE:
You can only restart a rulebook activation if it is currently enabled and the restart policy was set to **Always** when it was created.

Procedure

1. Select the **More Actions** icon ⋮ next to **Rulebook Activation enabled/disabled** toggle.
2. Select **Restart rulebook activation**.

3. In the window, select **Yes, I confirm that I want to restart these X rulebook activations**.
4. Select **Restart rulebook activations**.

Delete rulebook activations

End and permanently remove a rulebook activation and its configuration when its automated event-driven workflow is no longer required.

Procedure

1. Select the **More Actions** icon ⋮ next to the **Rulebook Activation enabled/disabled** toggle.
2. Select **Delete rulebook activation**.
3. In the window, select **Yes, I confirm that I want to delete these X rulebook activations**.
4. Select **Delete rulebook activations**.

Activate webhook-based automation in Openshift

In Openshift environments, you can activate webhooks by creating a route to expose the activation's service, enabling external systems to send events and trigger automation.

Before you begin

- You have created a rulebook activation.

NOTE:
The following is an example of rulebook with a given webhook:

```
- name: Listen for storage-monitor events
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5000
  rules:
    - name: Rule - Print event information
    condition: event.meta.headers is defined
    action:
      run_job_template:
        name: StorageRemediation
        organization: Default
        job_args:
          extra_vars:
             message: from eda
             sleep: 1
```

Procedure

1. Create a Route (on OpenShift Container Platform) to expose the service. The following is an example Route for an ansible-rulebook source that expects POST's on port 5000 on the decision environment pod:

```
kind: Route
apiVersion: route.openshift.io/v1
metadata:
  name: test-sync-bug
  namespace: dynatrace
  labels:
    app: eda
    job-name: activation-job-1-5000
spec:
  host: test-sync-bug-dynatrace.apps.aap-dt.ocp4.testing.ansible.com
  to:
    kind: Service
    name: activation-job-1-5000
    weight: 100
  port:
    targetPort: 5000
  tls:
    termination: edge
    insecureEdgeTerminationPolicy: Redirect
  wildcardPolicy: None
```

2. When you create the Route, test it with a **Post to the Route URL**:

NOTE:
You do not need the port as it is specified on the Route (targetPort).

```
curl -H "Content-Type: application/json" -X POST
test-sync-bug-dynatrace.apps.aap-dt.ocp4.testing.ansible.com -d
'{}'
```

Get started as an automation operator

As an automation operator, Ansible Automation Platform helps you monitor and execute automation projects with Red Hat certified collections or custom content for your organization.

View automation execution projects

A project is a logical collection of Ansible playbooks that you can manage in Ansible Automation Platform.

Platform administrators and automation developers have the permissions to create projects. As an automation operator you can view and sync projects. The following procedure desribes how to view available projects.

Procedure

1. From the navigation panel, select **Automation Execution > Projects**. The **Projects** page displays a list of projects that are currently available.
2. Click a project to view its details.

Next steps

For each project listed, you can sync the latest revision, edit the project, or copy the project's attributes using the icons next to each project.

Work with job templates

A job template is a definition and set of parameters for running an Ansible job.

A job template combines an Ansible Playbook from a project with the settings required to launch the job. Job templates are useful for running the same job many times. Job templates also encourage the reuse of Ansible Playbook content and collaboration between teams.

Platform administrators and automation developers have the permissions to create job templates. As an automation operator you can launch job templates and view their details.

Related information
Standardize and streamline automation with automation job templates

Launch a job template

Ansible Automation Platform provides push-button deployment with job templates. These templates store playbook parameters, inventories, and credentials normally passed through the command line.

Procedure

1. From the navigation panel, select **Automation Execution > Templates**.

2. Select a template to view its details. A default job template is created during your initial setup to help you get started, but you can also create your own.
3. From the **Templates** page, click the launch icon to run your job template.

Result

The **Templates** list view shows job templates that are currently available. The default view is collapsed (Compact), showing the template name, template type, and the timestamp of the last job that ran using that template. You can click the arrow icon next to each entry to expand and view more information. This list is sorted alphabetically by name, but you can sort by other criteria, or search by various template fields and attributes.

From this screen you can launch, edit, and copy a job template.

Related information
Job Templates
Workflow job templates

Surveys in job templates

Surveys provide a way to prompt users for input when launching a job from a job template. This input can then be used as variables in the playbook run.

Job types of **Run** or **Check** provide a way to set up surveys in the **Job Template** creation or editing screens. Surveys set extra variables for the playbook similar to **Prompt for Extra Variables** does, but in a user-friendly question and answer way. Surveys also permit for validation of user input. Select the **Survey** tab to create a survey.

Example

You can use surveys for several situations. For example, operations want to give developers a "push to stage" button that they can run without advance knowledge of Ansible. When launched, this task could prompt for answers to questions such as "What tag should we release?".

You can ask many types of questions, including multiple-choice questions.

Create a survey

You can create a survey for a job template to prompt users for input when they launch a job based on that template. Surveys can include many questions of various types, such as text input, multiple choice, and passwords.

The responses provided by users are stored in Ansible variables that can be used within the playbook associated with the job template.

Procedure

1. From the navigation panel, select **Automation Execution > Templates**.
2. Select the job template you want to create a survey for.
3. From the **Survey** tab, click **Create survey question**.
4. A survey can consist of any number of questions. For each question, enter the following information:
   - **Question**: The question to ask the user.
   - Optional: **Description**: A description of what is being asked of the user.
   - **Answer variable name**: The Ansible variable name to store the user's response in. This is the variable to be used by the playbook. Variable names cannot contain spaces.
   - **Answer type**: Choose from the following question types:
     - **Text**: A single line of text. You can set the minimum and maximum length (in characters) for this answer.
     - **Textarea**: A multi-line text field. You can set the minimum and maximum length (in characters) for this answer.
     - **Password**: Responses are treated as sensitive information, similar to the way an actual password is treated. You can set the minimum and maximum length (in characters) for this answer.
     - **Multiple Choice (single select)**: A list of options, of which only one can be selected at a time. Enter the options, one per line, in the **Multiple Choice Options** field.
     - **Multiple Choice (multiple select)**: A list of options, any number of which can be selected at a time. Enter the options, one per line, in the **Multiple Choice Options** field.
     - **Integer**: An integer number. You can set the minimum and maximum length (in characters) for this answer.
     - **Float**: A decimal number. You can set the minimum and maximum length (in characters) for this answer.
   - **Required**: Whether or not an answer to this question is required from the user.
   - **Minimum length** and **Maximum length**: Specify if a certain length in the answer is required.
   - **Default answer**: The default answer to the question. This value is pre-filled in the interface and is used if the answer is not provided by the user.
5. Once you have entered the question information, click **Create question** to add the question.

   The survey question displays in the **Survey** list. For any question, you can click ✎ to edit it.

   Check the box next to each question and click **Delete** to delete the question, or use the toggle option in the menu bar to enable or disable the survey prompts.

If you have more than one survey question, click **Edit Order** to rearrange the order of the questions by clicking and dragging on the grid icon.

6. To add more questions, click **Add**.

View hosts being automated in inventory files

An inventory is a file listing the collection of hosts managed by Ansible Automation Platform. Organizations are assigned to inventories, while permissions to launch playbooks against inventories are controlled at the user or team level.

Platform administrators and automation developers have the permissions to create inventories. As an automation operator you can view inventories and their details.

Execute an inventory

The following steps describe how to execute an inventory.

Procedure

1. From the navigation panel, select **Automation Execution › Infrastructure › Inventories**. The **Inventories** window displays a list of inventories that are currently available, along with the following information:
   - **Name**: The inventory name.
   - **Status**: The statuses are:
     - **Success**: The inventory sync completed successfully.
     - **Disabled**: No inventory source added to the inventory.
     - **Error**: The inventory source completed with error.
   - **Type**: Identifies whether the inventory is a standard inventory, a smart inventory, or a constructed inventory.
   - **Organization**: The organization to which the inventory belongs.
2. Select an inventory name to display the **Details** page for the inventory, including the inventory’s groups and hosts.

Related information
Inventories

View the status and output of automation jobs

A job is an instance of Ansible Automation Platform launching an Ansible Playbook against an inventory of hosts. You can view the status and output of automation jobs.

Review a job status

The **Jobs** list view displays a list of jobs and their statuses, shown as completed successfully, failed, or as an active (running) job.

Procedure

1. From the navigation panel, select **Automation Execution > Jobs**.

   The default view is collapsed (Compact) with the job name, status, job type, start, and finish times. You can click the arrow icon to expand and see more information. You can sort this list by various criteria, and perform a search to filter the jobs of interest.

2. From this screen, you can complete the following tasks:

   - View a job's details and standard output.
   - Relaunch jobs.
   - Remove selected jobs.
     The relaunch operation only applies to relaunches of playbook runs and does not apply to project or inventory updates, system jobs, or workflow jobs.

Review job output

When you relaunch a job, the jobs **Output** view is displayed.

Procedure

1. From the navigation panel, select **Automation Execution > Jobs**.
2. Select a job. This takes you to the **Output** view for that job, where you can filter job output by these criteria:

   - The **Search output** option allows you to search by keyword.

- The **Event** option enables you to filter by the events of interest, such as errors, host failures, host retries, and items skipped. You can include as many events in the filter as necessary.

Red Hat product documentation legal notices

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions. “This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code. The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions. All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law. No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work’s users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies. You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions. You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so. A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms. You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms. “Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination. You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies. You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients. Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents. A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom. If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License. Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License. The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty. THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES

SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16. If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see link:https://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read link:https://www.gnu.org/licenses/why-not-lgpl.html.

Apache license

Version 2.0, January 2004

http://www.apache.org/licenses/

Terms and Conditions for use, reproduction, and distribution

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, **"control"** means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or **"Your"**) shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, **"submitted"** means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as **"Not a Contribution."**

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a **"NOTICE"** text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS