Red Hat Ansible Automation Platform

2.4

Event-Driven Ansible controller user guide

Learn to configure and use Event-Driven Ansible controller to enhance and expand automation
Red Hat Ansible Automation Platform 2.4 Event-Driven Ansible controller user guide

Learn to configure and use Event-Driven Ansible controller to enhance and expand automation
Abstract

This guide helps you configure your Event-Driven Ansible controller to set up new projects, decision environments, tokens to authenticate to Ansible Automation Platform Controller, and rulebook activation.
Table of Contents

PREFACE ................................................................................................................. 3

MAKING OPEN SOURCE MORE INCLUSIVE ......................................................... 4

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION ...................................... 5

CHAPTER 1. EVENT-DRIVEN ANSIBLE CONTROLLER OVERVIEW ......................... 6

CHAPTER 2. PROJECTS .............................................................................................. 7
  2.1. SETTING UP A NEW PROJECT .................................................................. 7

CHAPTER 3. DECISION ENVIRONMENTS ................................................................. 8
  3.1. SETTING UP A NEW DECISION ENVIRONMENT ...................................... 8
  3.2. BUILDING A CUSTOM DECISION ENVIRONMENT FOR EVENT-DRIVEN ANSIBLE WITHIN ANSIBLE AUTOMATION PLATFORM .................. 8

CHAPTER 4. SETTING UP A TOKEN ......................................................................... 11
  4.1. SETTING UP A TOKEN TO AUTHENTICATE TO ANSIBLE AUTOMATION PLATFORM CONTROLLER .......................................................... 11

CHAPTER 5. RULEBOOK ACTIVATIONS .................................................................. 12
  5.1. SETTING UP A RULEBOOK ACTIVATION .............................................. 12
  5.2. ENABLING AND DISABLING RULEBOOK ACTIVATIONS ....................... 13
  5.3. RESTARTING RULEBOOK ACTIVATIONS .............................................. 13
  5.4. DELETING RULEBOOK ACTIVATIONS .................................................. 13
  5.5. ACTIVATING WEBHOOK RULEBOOKS ................................................. 13
  5.6. TESTING WITH KUBERNETES ................................................................ 15
Event-Driven Ansible controller is a new way to enhance and expand automation by improving IT speed and agility while enabling consistency and resilience. Developed by Red Hat, this feature is designed for simplicity and flexibility.
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our technical content and encourage you to tell us what you think. If you’d like to add comments, provide insights, correct a typo, or even ask a question, you can do so directly in the documentation.

NOTE
You must have a Red Hat account and be logged in to the customer portal.

To submit documentation feedback from the customer portal, do the following:

1. Select the Multi-page HTML format.

2. Click the Feedback button at the top-right of the document.

3. Highlight the section of text where you want to provide feedback.

4. Click the Add Feedback dialog next to your highlighted text.

5. Enter your feedback in the text box on the right of the page and then click Submit.

We automatically create a tracking issue each time you submit feedback. Open the link that is displayed after you click Submit and start watching the issue or add more comments.
CHAPTER 1. EVENT-DRIVEN ANSIBLE CONTROLLER
OVERVIEW

Event-Driven Ansible is a highly scalable, flexible automation capability that works with event sources such as other software vendors’ monitoring tools. These tools watch IT solutions and identify events and automatically implement the documented changes or response in a rulebook to handle that event.

The following procedures form the user configuration:

- Setting up a new project
- Setting up a new decision environment
- Setting up a token to authenticate to Ansible Automation Platform Controller
- Setting up a rulebook activation
CHAPTER 2. PROJECTS

Projects are a logical collection of rulebooks.

2.1. SETTING UP A NEW PROJECT

Projects are a logical collection of rulebooks.

Prerequisites

- You are logged in to the Event-Driven Ansible controller Dashboard as a Content Consumer.
- You have set up a credential, if necessary. For more information, refer to the Credentials section of the automation controller documentation.
- You have an existing repository containing rulebooks that are integrated with playbooks contained in a repository to be used by automation controller.

Procedure

1. Log in to the Event-Driven Ansible controller Dashboard.

2. From the navigation panel, select Projects → Create project.

3. Insert the following:
   - **Name**
     Enter project name.
   - **Description**
     This field is optional.
   - **SCM type**
     Git is the only SCM type available for use.
   - **SCM URL**
     HTTP[S] protocol address of a repository, such as GitHub or GitLab.

   **NOTE**
   You cannot edit the SCM URL after you create the project.

   - **Credential**
     This field is optional. This is the token needed to utilize the SCM URL.

4. Select Create project.

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 the project.
CHAPTER 3. DECISION ENVIRONMENTS

Decision environments are a container image to run Ansible rulebooks. They create a common language for communicating automation dependencies, and provide a standard way to build and distribute the automation environment. The default decision environment is found in the Ansible-Rulebook.

To create your own decision environment refer to Building a custom decision environment for Event-Driven Ansible within Ansible Automation Platform.

3.1. SETTING UP A NEW DECISION ENVIRONMENT

The following steps describe how to import a decision environment into your Event-Driven Ansible controller Dashboard.

Prerequisites

- You are logged in to the Event-Driven Ansible controller Dashboard as a Content Consumer.
- You have set up a credential, if necessary. For more information, refer to the Credentials section of the Automation controller documentation.
- 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 the Event-Driven Ansible controller Dashboard.
2. From the navigation panel, select Decision Environments → Create decision environment.
3. Insert 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 utilize the decision environment image.
4. Select Create decision environment.

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

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

3.2. BUILDING A CUSTOM DECISION ENVIRONMENT FOR EVENT-DRIVEN ANSIBLE WITHIN ANSIBLE AUTOMATION PLATFORM
Refer to this section if you need a custom decision environment to provide a custom maintained or third-party event source plugin that is not available in the default decision environment.

**Prerequisites**

- Ansible Automation Platform $\geq 2.4$
- Event-Driven Ansible
- Ansible Builder $\geq 3.0$

**Procedure**

- Add the de-supported decision environment. This image is built from a base image provided by Red Hat called **de-minimal**.

**NOTE**

Red Hat recommends using de-minimal as the base image with Ansible Builder to build your custom decision environments.

The following is an example of the Ansible builder definition file using de-minimal as a base image to build a custom decision environment with the ansible.eda collection:

```yaml
version: 3
images:
  base_image:
    name: 'registry.redhat.io/ansible-automation-platform-24/de-minimal-rhel8:latest'
dependencies:
  galaxy:
    collections:
      - ansible.eda
  python_interpreter:
    package_system: "python39"
  options:
    package_manager_path: /usr/bin/microdnf
```

Additionally, if other python packages or RPM are needed, you can add the following to a single definition file:

```yaml
version: 3
images:
  base_image:
    name: 'registry.redhat.io/ansible-automation-platform-24/de-minimal-rhel8:latest'
dependencies:
  galaxy:
    collections:
      - ansible.eda
  python:
```

- six
- psutil

system:
  - iputils [platform: rpm]

python_interpreter:
  package_system: "python39"

options:
  package_manager_path: /usr/bin/microdnf
CHAPTER 4. SETTING UP A TOKEN

Automation controller must contain a project based on a repository with certain playbooks designed to work with the Event-Driven Ansible rulebooks. Automation controller must also have corresponding job templates set up based on the playbooks in that project.

4.1. SETTING UP A TOKEN TO AUTHENTICATE TO ANSIBLE AUTOMATION PLATFORM CONTROLLER

Prerequisites

- You are logged in to the Event-Driven Ansible controller Dashboard as a Content Consumer.
- You have created a user.
- You can log in to the Event-Driven Ansible controller Dashboard or you are added as a user in the organization.

Procedure

1. Navigate to the Event-Driven Ansible controller Dashboard.
2. From the top navigation panel, select your profile.
3. Go to User details.
4. Select Controller Tokens → Create controller token
5. Insert the following:
   - **Name**
     Insert the name.
   - **Description**
     This field is optional.
   - **Token**
     Create the token in automation controller. For more information on creating the token, refer to the Users - Tokens section of the Automation controller User Guide.

   **NOTE**

   The token must be in write-scope.

6. Select Create controller token.

After saving the new token, you are brought to the Controller Tokens tab where you can delete the token.
CHAPTER 5. RULEBOOK ACTIVATIONS

Rulebook activations are rulebooks that have been activated to run.

5.1. SETTING UP A RULEBOOK ACTIVATION

Prerequisites

- You are logged in to the Event-Driven Ansible controller Dashboard as a Content Consumer.
- You have set up a project.
- You have set up a decision environment.
- You have set up an authentication token.

Procedure

1. Navigate to the Event-Driven Ansible controller Dashboard.
2. From the navigation panel, select Rulebook Activations → Create rulebook activation.
3. Insert the following:
   
   **Name**
   - Insert the name.

   **Description**
   - This field is optional.

   **Project**
   - Projects are a logical collection of rulebooks.

   **Rulebook**
   - Rulebooks will be shown according to the project selected.

   **Decision environment**
   - Decision environments are a container image to run Ansible rulebooks.

   **Restart policy**
   - This is a policy to decide when to restart a rulebook.
     
     - Policies:
       1. Always: Restarts when a rulebook finishes
       2. Never: Never restarts a rulebook when it finishes
       3. On failure: Only restarts when it fails

   **Rulebook activation enabled?**
   - This automatically enables the rulebook activation to run.

   **Variables**
Pass extra command line variables to the playbook in the Variables section. These are the `-e` or `--extra-vars` command line parameters for ansible-rulebook. Provide key/value pairs using either YAML or JSON. Refer to the documentation for example syntax.

4. Select **Create rulebook activation**.

Your rulebook activation is now created and can be managed in the **Rulebook Activations** screen.

After saving the new rulebook activation, the rulebook activation’s details page is displayed. From there or the **Rulebook Activations** list view you can edit or delete it.

### 5.2. ENABLING AND DISABLING RULEBOOK ACTIVATIONS

1. Select the switch on the row level to enable or disable your chosen rulebook.

2. In the popup window, select **Yes, I confirm that I want to enable/disable these X rulebook activations**.

3. Select **Enable/Disable rulebook activation**.

### 5.3. RESTARTING RULEBOOK ACTIVATIONS

**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.

1. Click the **More Actions** icon to next to **Rulebook Activation enabled/disabled** toggle.

2. Select **Restart rulebook activation**.

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

4. Select **Restart rulebook activations**.

### 5.4. DELETING RULEBOOK ACTIVATIONS

1. Select the **More Actions** icon to next to the **Rulebook Activation enabled/disabled** toggle.

2. Select **Delete rulebook activation**.

3. In the popup window, select **Yes, I confirm that I want to delete these X rulebook activations**.

4. Select **Delete rulebook activations**.

### 5.5. ACTIVATING WEBHOOK RULEBOOKS

In Openshift environments, you can allow webhooks to reach an activation-job-pod over a given port by creating a Route that exposes that rulebook activation’s Kubernetes service.

**Prerequisites**

- You have created a rulebook activation in the Event-Driven Ansible controller Dashboard.
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:

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

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

```bash
curl -H "Content-Type: application/json" -X POST
  test-sync-bug-dynatrace.apps.aap-dt.ocp4.testing.ansible.com
  -d '{}'
```
NOTE
You do not need the port as it is specified on the Route (targetPort).

5.6. TESTING WITH KUBERNETES

With Kubernetes you can create an Ingress, or expose the port, but not for production.

Procedure

1. Run the following command to expose the port on the cluster for a given service:
   ```bash
   kubectl port-forward svc/<ACTIVATION_SVC_NAME> 5000:5000
   ```

2. Make the HTTP requests against the localhost:5000 to trigger the rulebook:
   ```bash
   curl -H "Content-Type: application/json" -X POST test-sync-bug-dynatrace.apps.aap-dt.ocp4.testing.ansible.com -d '{}'
   ```