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

Last Updated: 2024-10-10

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

Legal Notice

Abstract

Learn how to configure your Event-Driven Ansible controller to set up credentials, new projects, decision environments, tokens to authenticate to Ansible Automation Platform Controller, and rulebook activation.

Table of Contents

PREFACE

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.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

If you have a suggestion to improve this documentation, or find an error, you can contact technical support at https://access.redhat.com to open a request.

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 monitor 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 credentials
- 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

NOTE

API documentation for Event-Driven Ansible controller is available at https://<eda-server-host>/api/eda/v1/docs

CHAPTER 2. SETTING UP CREDENTIALS FOR EVENT-DRIVEN ANSIBLE CONTROLLER

Credentials are used by Event-Driven Ansible for authentication when launching rulebooks.

2.1. SETTING UP CREDENTIALS

Create a credential to use with a private repository (GitHub or GitLab) or a private container registry.

> **IMPORTANT**
>
> If you are using a GitHub or GitLab repository, use the **basic auth** method. Both SCM servers are officially supported. You can use any SCM provider that supports **basic auth**.

Procedure

1. Log in to the Event-Driven Ansible controller Dashboard.
2. From the navigation panel, select **Resources → Credentials**.
3. Click **Create credential**.
4. Insert the following:

   Name
   Insert the name.

   Description
   This field is optional.

   Credential type
   The options available are a GitHub personal access token, a GitLab personal access token, or a container registry.

   Username
   Insert the username.

   Token
   Insert a token that allows you to authenticate to your destination.

   > **NOTE**
   >
   > If you are using a container registry, the token field can be a token or a password, depending on the registry provider. If you are using the Ansible Automation Platform hub registry, insert the password for that in the token field.

5. Click **Create credential**.

After saving the credential, the credentials details page is displayed. From there or the **Credentials** list view, you can edit or delete it.

2.2. CREDENTIALS LIST VIEW

On the **Credentials** page, you can view the list of created credentials that you have created along with the **Type** of credential.

From the menu bar, you can search for credentials in the **Name** field.

You also have the following options in the menu bar:

- Choose which columns are shown in the list view by clicking **Manage columns**.
- Choose between a **List view** or a **Card view**, by clicking the icons.

2.3. EDITING A CREDENTIAL

Procedure

1. Edit the credential by using one of these methods:
   - From the **Credentials** list view, click the **Edit credential** icon next to the desired credential.
   - From the **Credentials** list view, select the name of the credential, click **Edit credential**.
2. Edit the appropriate details and click **Save credential**.

2.4. DELETING A CREDENTIAL

Procedure

1. Delete the credential by using one of these methods:
   - From the **Credentials** list view, click the **More Actions** icon ⋮ next to the desired credential and click **Delete credential**.
   - From the **Credentials** list view, select the name of the credential, click the **More Actions** icon ⋮ next to **Edit credential**, and click **Delete credential**.
2. In the pop-up window, select **Yes, I confirm that I want to delete this credential**
3. Click **Delete credential**.

You can delete multiple credentials at a time by selecting the checkbox next to each credential and clicking the **More Actions** icon ⋮ in the menu bar and then clicking **Delete selected credentials**.

CHAPTER 3. PROJECTS

Projects are a logical collection of rulebooks. They must be a git repository and only http protocol is supported. The rulebooks of a project must be located in the path defined for Event-Driven Ansible content in Ansible collections: **/extensions/eda/rulebooks** at the root of the project.

3.1. SETTING UP A NEW PROJECT

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, see the Setting up credentials section.
- 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.

   Options

   The Verify SSL option is enabled by default. Enabling this option verifies the SSL with HTTPS when the project is imported.

   > NOTE
   >
   > You can disable this option if you have a local repository that uses self-signed certificates.

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 it.

3.2. PROJECTS LIST VIEW

On the **Projects** page, you can view the projects that you have created along with the **Status** and the **Git hash**.

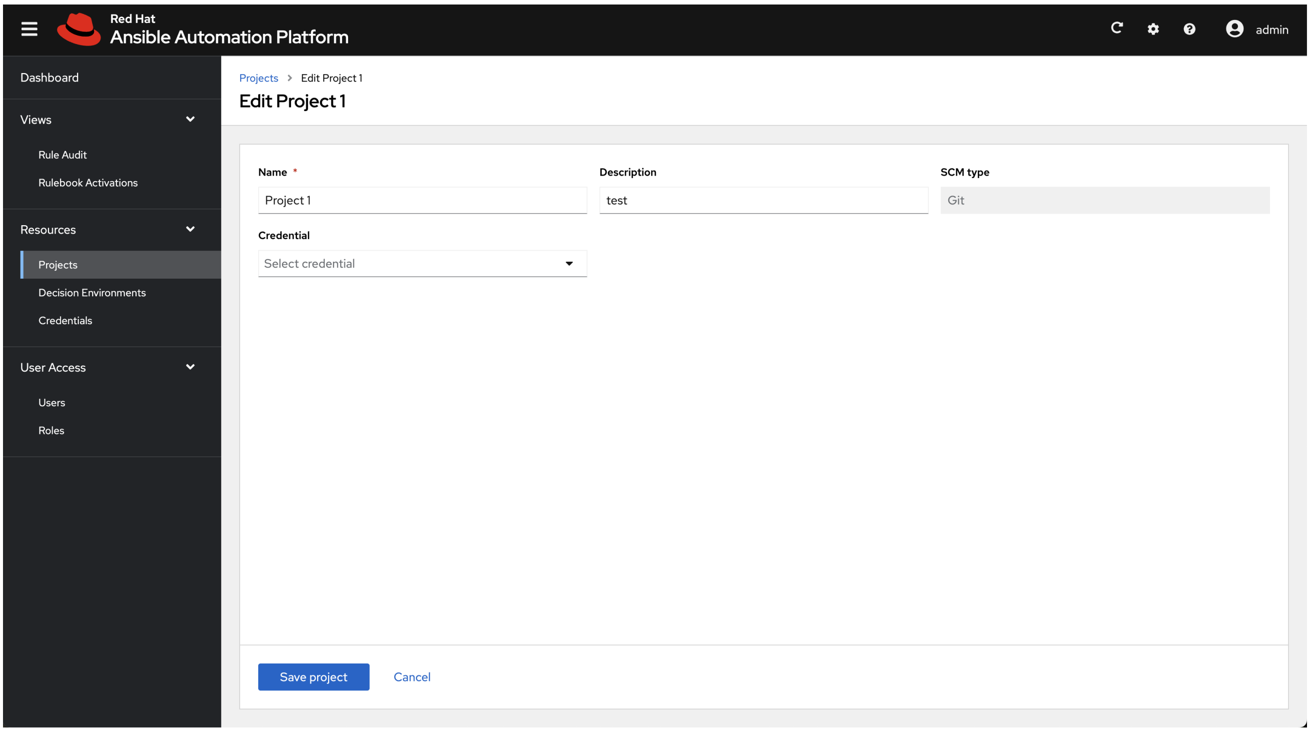
NOTE

If a rulebook changes in source control you can re-sync a project by selecting the sync icon next to the project from the **Projects** list view. The **Git hash** updates represent the latest commit on that repository. An activation must be restarted or recreated if you want to use the updated project.

3.3. EDITING A PROJECT

Procedure

1. From the **Projects** list view, select the **More Actions** icon ⋮ next to the desired project.
2. Select **Edit project**.
3. Enter the required changes and select **Save project**.



Edit Project 1

Name * Project 1

Description test

SCM type Git

Credential Select credential

3.4. DELETING A PROJECT

Procedure

1. From the **Projects** list view, select the **More Actions** icon ⋮ next to the desired project.
2. Select **Delete project**.
3. In the popup window, select **Yes, I confirm that I want to delete this project**.
4. Select **Delete project**.

CHAPTER 4. 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.

4.1. 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, see the Setting up credentials section.
- 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**.
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 Environments** list view, you can edit or delete it.

4.2. BUILDING A CUSTOM DECISION ENVIRONMENT FOR EVENT-DRIVEN ANSIBLE WITHIN ANSIBLE AUTOMATION PLATFORM

Use the following instructions 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 > = 2.4
- Event-Driven Ansible
- Ansible Builder > = 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 that uses **de-minimal** as a base image to build a custom decision environment with the ansible.eda collection:

```
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 you need other Python packages or RPMs, you can add the following to a single definition file:

```
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 5. SETTING UP AN AUTOMATION CONTROLLER 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.

5.1. SETTING UP A TOKEN TO AUTHENTICATE TO AUTOMATION 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. Go 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 about creating the token, see the Users - Tokens section of the Automation controller User Guide.

   NOTE

   Your token must be in write-scope and if you are the assigned user, you must have permission to execute job and workflow templates.

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 6. RULEBOOK ACTIVATIONS

A rulebook activation is a process running in the background defined by a decision environment executing a specific rulebook.

6.1. 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 automation controller token.

Procedure

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

   Name

   Insert the name.

   Description

   This field is optional.

   Project

   Projects are a logical collection of rulebooks.

   Rulebook

   Rulebooks are shown according to the project selected.

   Decision environment

   Decision environments are 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 a policy to decide when to restart a rulebook.

   - Policies:

     i. Always: Restarts when a rulebook finishes

     ii. Never: Never restarts a rulebook when it finishes

iii. On failure: Only restarts when it fails

Rulebook activation enabled?

This automatically enables the rulebook activation to run.

Variables

The variables for the rulebook are in a JSON/YAML format. The content would be equivalent to the file passed through the **--vars** flag of ansible-rulebook command.

4. Click **Create rulebook activation**.

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

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.

NOTE

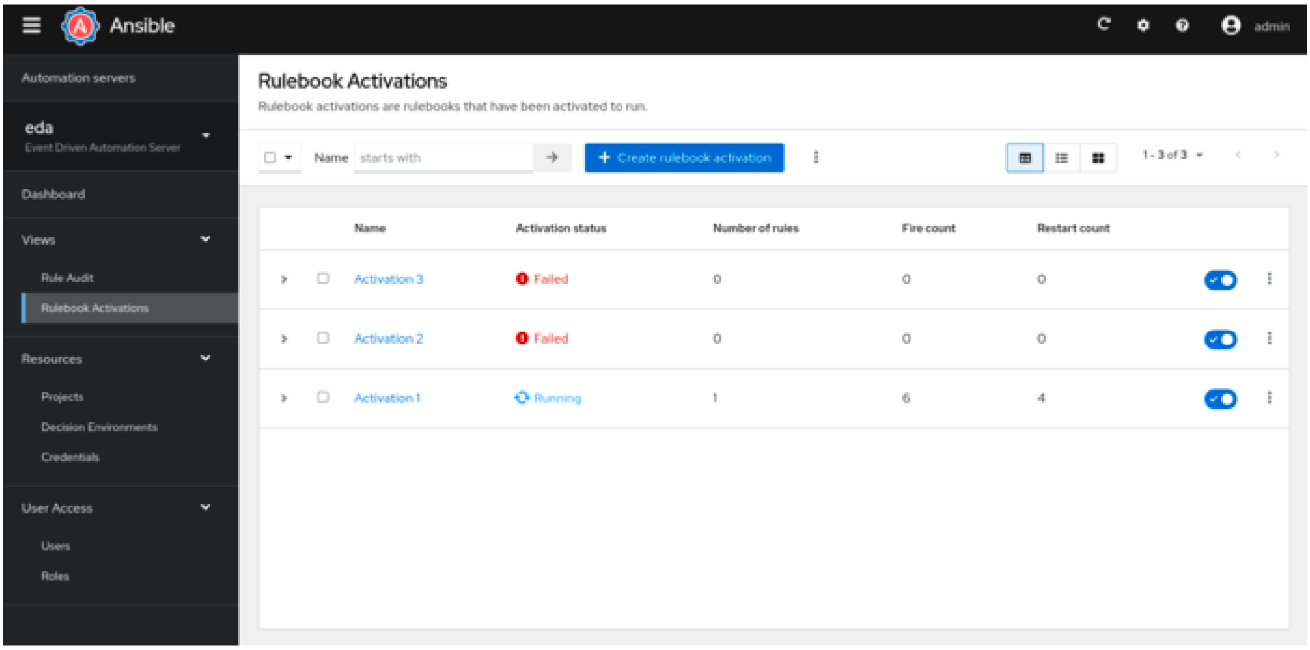
Occasionally, when a source plugin shuts down, it causes a rulebook to exit gracefully after a certain amount of time. When a rulebook activation shuts down, any tasks that are waiting to be performed will be canceled, and an info level message will be sent to the activation log. For more information,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

6.2. RULEBOOK ACTIVATION LIST VIEW

On the **Rulebook Activations** page, you can view the rulebook activations that you have created along with the **Activation status**, **Number of rules associated** with the rulebook, the **Fire count**, and **Restart count**.

If the **Activation 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.



Rulebook Activations

Rulebook activations are rulebooks that have been activated to run.

Name	Activation status	Number of rules	Fire count	Restart count
Activation 3	Failed	0	0	0
Activation 2	Failed	0	0	0
Activation 1	Running	1	6	4

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

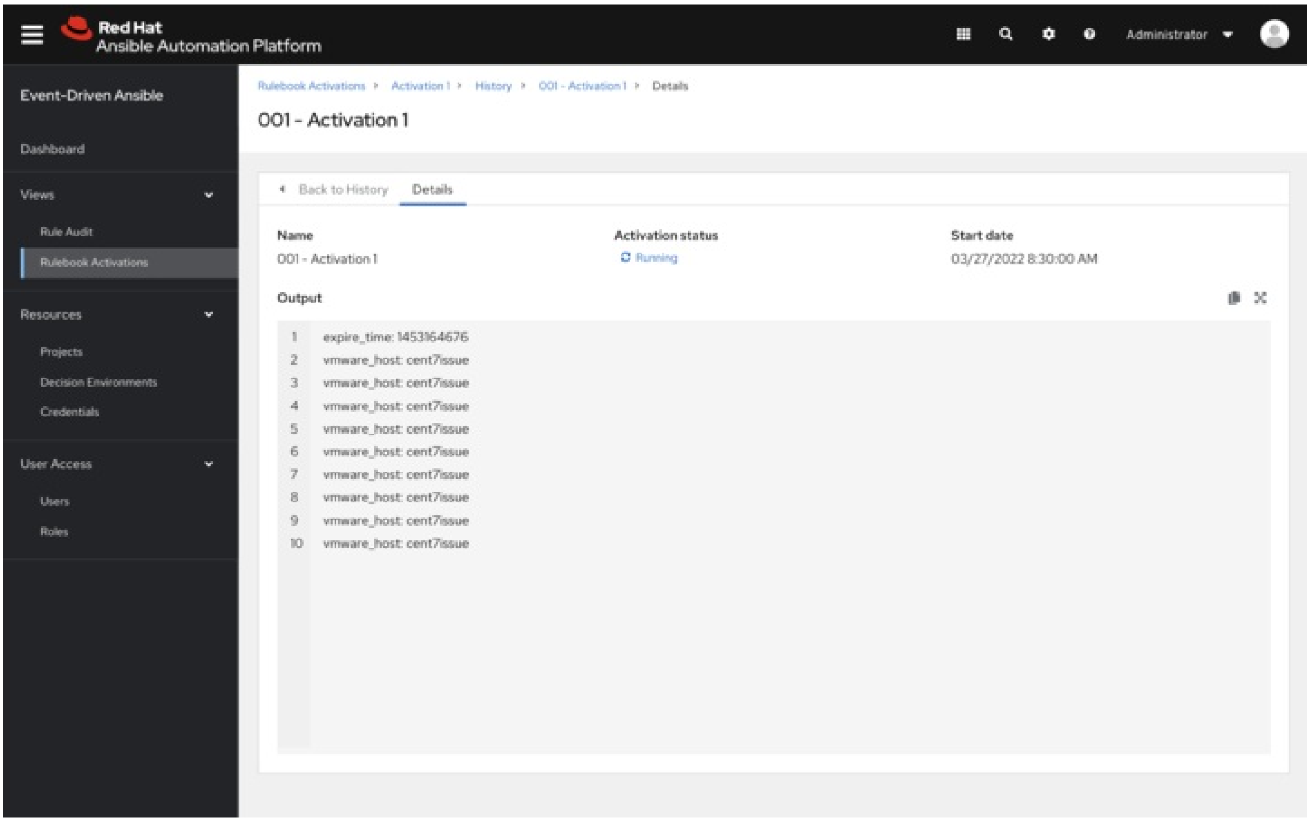
6.2.1. Viewing activation output

You can view the output of the activations in the **History** tab.

Procedure

1. Select the **History** tab to access the list of all the activation instances. An activation instance represents a single execution of the activation.

2. Then select the activation instance in question, this will show you the **Output** produced by that specific execution.



001 - Activation 1

Name: 001 - Activation 1

Activation status: Running

Start date: 03/27/2022 8:30:00 AM

Output

```
expire_time: 1453164676
vmware_host: cent7issue
vmware_host: cent7issue
vmware_host: cent7issue
vmware_host: cent7issue
vmware_host: cent7issue
vmware_host: cent7issue
vmware_host: cent7issue
vmware_host: cent7issue
vmware_host: cent7issue
```

To view events that came in and triggered an action, you can use the Rule Audit section in the Event-Driven Ansible controller Dashboard.

6.3. 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**.

6.4. 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. Select the **More Actions** icon ⋮ 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**.

6.5. DELETING RULEBOOK ACTIVATIONS

1. Select the **More Actions** icon ⋮ 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**.

6.6. 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:

```
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**:

```
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).

6.7. 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:

```
kubectl port-forward svc/<ACTIVATION_SVC_NAME> 5000:5000
```

2. Make the HTTP requests against the **localhost:5000** to trigger the rulebook:

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

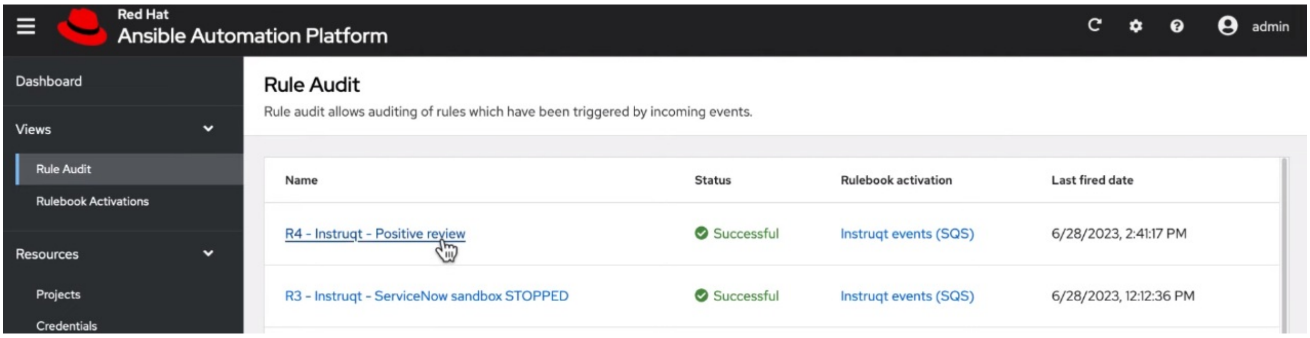
CHAPTER 7. RULE AUDIT

Rule audit allows the auditing of rules which have been triggered by all the rules that were activated at some point.

The **Rule Audit** list view shows you a list of every time an event came in that matched a condition within a rulebook and triggered an action. The list shows you rules within your rulebook and each heading matches up to a rule that has been executed.

7.1. VIEWING RULE AUDIT DETAILS

From the **Rule Audit** list view you can check the event that triggered specific actions.



Rule Audit

Rule audit allows auditing of rules which have been triggered by incoming events.

Name	Status	Rulebook activation	Last fired date
R4 - Instruqt - Positive review	✓ Successful	Instruqt events (SQS)	6/28/2023, 2:41:17 PM
R3 - Instruqt - ServiceNow sandbox STOPPED	✓ Successful	Instruqt events (SQS)	6/28/2023, 12:12:36 PM

Procedure

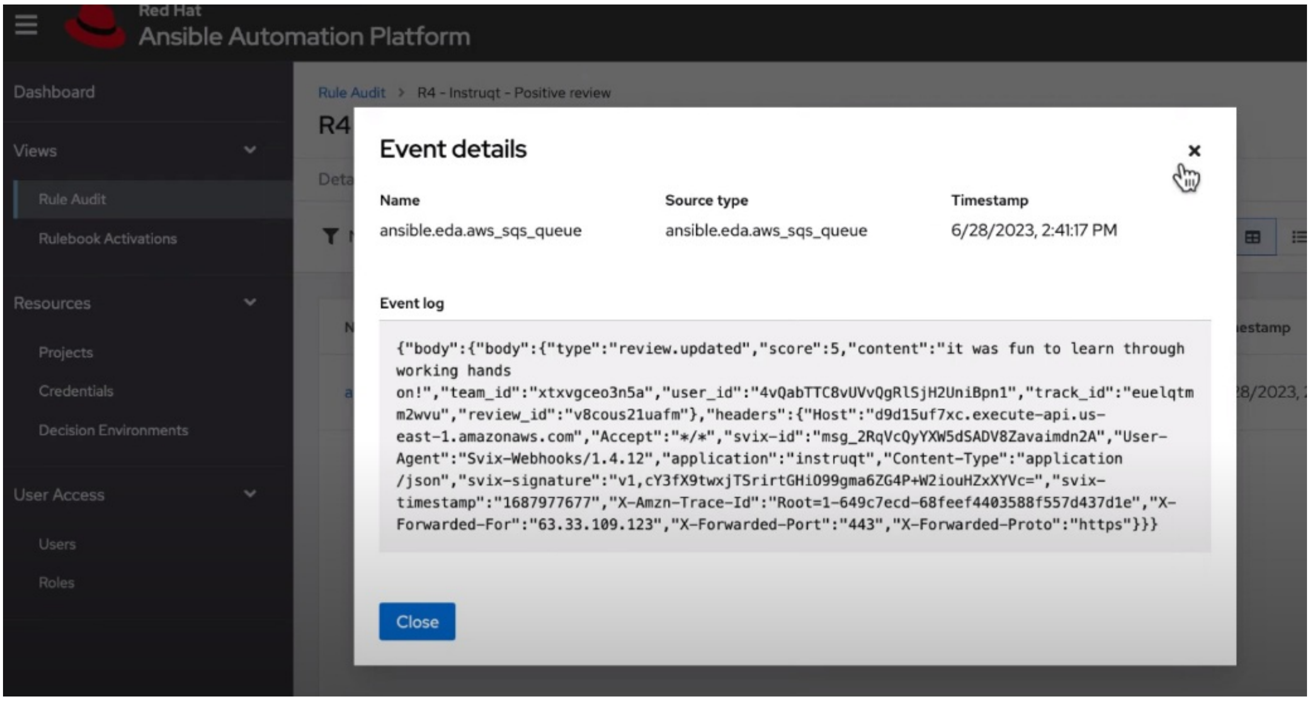
1. From the navigation panel select **Rule Audit**.
2. Select the desired rule, this brings you to the **Details** tab.

From here you can view when it was created, when it was last fired, and the rulebook activation that it corresponds to.

7.2. VIEWING RULE AUDIT EVENTS

Procedure

1. From the navigation panel select **Rule Audit**.
2. Select the desired rule, this brings you to the **Details** tab. To view all the events that triggered an action, select the **Events** tab. This shows you the event that triggered actions.
3. Select an event to view the **Event log**, along with the **Source type** and **Timestamp**.



Event details

Name: ansible.eda.aws_sqs_queue

Source type: ansible.eda.aws_sqs_queue

Timestamp: 6/28/2023, 2:41:17 PM

Event log

```
{"body":{"body":{"type":"review.updated","score":5,"content":"it was fun to learn through
working hands
on!","team_id":"xtxvgceo3n5a","user_id":"4vQabTTC8vUVvQgRlSjH2UniBpn1","track_id":"euelqtm
m2wvu","review_id":"v8cous21uafm"},"headers":{"Host":"d9d15uf7xc.execute-api.us-
east-1.amazonaws.com","Accept":"*/*","svix-id":"msg_2RqVcQyYXW5dSADV8Zavaimdn2A","User-
Agent":"Svix-Webhooks/1.4.12","application":"instruqt","Content-Type":"application
/json","svix-signature":"v1,cY3fX9twxjTSrirtGHiO99gma6ZG4P+W2iouHZxXYVc=","svix-
timestamp":"1687977677","X-Amzn-Trace-Id":"Root=1-649c7ecd-68feef4403588f557d437d1e","X-
Forwarded-For":"63.33.109.123","X-Forwarded-Port":"443","X-Forwarded-Proto":"https"}}}
```

7.3. VIEWING RULE AUDIT ACTIONS

Procedure

1. From the navigation panel select **Rule Audit**.
2. Select the desired rule, this brings you to the **Actions** tab.

From here you can view executed actions that were taken. Some actions are linked out to automation controller where you can view the output.

CHAPTER 8. PERFORMANCE TUNING FOR EVENT-DRIVEN ANSIBLE CONTROLLER

Event-Driven Ansible is a highly scalable, flexible automation capability. Event-Driven Ansible controller provides the interface in which Event-Driven Ansible automation performs. Tune your Event-Driven Ansible controller to optimize performance and scalability through:

- Characterizing your workload
- System level monitoring
- Performance troubleshooting

8.1. CHARACTERIZING YOUR WORKLOAD

In Event-Driven Ansible controller, your workload includes the number of rulebook activations and events being received. Consider the following factors to characterize your Event-Driven Ansible controller workload:

1. Number of simultaneous rulebook activations
2. Number of events received by Event-Driven Ansible controller

8.1.1. Modifying the number of simultaneous rulebook activations

By default, Event-Driven Ansible controller allows 12 rulebook activations to run simultaneously. If any more than 12 rulebook activations are created, the expected behavior is that subsequent rulebook activations wait in pending until there is an available rulebook activation worker. In this case, the rulebook activation status will display as “pending” even if there is enough free memory and CPU on your Event-Driven Ansible controller instance. To change this behavior, you must change the default maximum number of running rulebook activations.

Note: The value for **EDA_MAX_RUNNING_ACTIVATIONS** doesn’t change with the change in instance size, so it needs to be adjusted manually.

8.1.1.1. Modifying the number of simultaneous rulebook activations during Event-Driven Ansible controller installation

By default, Event-Driven Ansible controller allows 12 activations to run simultaneously. You can modify this default value during installation by using the following procedure:

Procedure

Provide a variable to the VM installer:

1. Navigate to the setup inventory file.
2. Add **automationedacontroller_max_running_activations** in the [all:vars] section. For example, **automationedacontroller_max_running_activations=16**.
3. Run the setup.

8.1.1.2. Modifying the number of simultaneous rulebook activations after Event-Driven Ansible controller installation

By default, Event-Driven Ansible controller allows 12 activations to run simultaneously. You can modify this default value after installation by using the following procedure:

Procedure

1. Navigate to the environment file at **/etc/ansible-automation-platform/eda**.
2. Choose the number of maximum running activations that you need. For example, **EDA_MAX_RUNNING_ACTIVATIONS = 16**
3. Use the following command to restart EDA services: **systemctl restart automation-eda-controller.target**

Resources

For more information about rulebook activations, see the Rulebook activations.

8.1.2. Modifying the default memory limit for each rulebook activation

Memory usage is based on the number of events that Event-Driven Ansible controller has to process. Each rulebook activation container has a 200MB memory limit. For example, with 4 CPU and 16GB of RAM, one rulebook activation container with an assigned 200MB memory limit can not handle more than 150,000 events per minute. If the number of parallel running rulebook activations is higher, then the maximum number of events each rulebook activation can process is reduced. If there are too many incoming events at a very high rate, the container can run out of memory trying to process the events. This will kill the container, and your rulebook activation will fail with a status code of 137.

To address this failure, you can increase the amount of memory allocated to rulebook activations in order to process a high number of events at a high rate by using one of the following procedures:

- Modifying the default memory limit for each rulebook activation during installation
- Modifying the default memory limit for each rulebook activation after installation

8.1.2.1. Modifying the default memory limit for each rulebook activation during installation

By default, each rulebook activation container has a 200MB memory limit. You can modify this default value during installation by using the following procedure:

Procedure

1. Navigate to the setup inventory file.
2. Add **automationedacontroller_podman_mem_limit** in the [all:vars] section. For example, **automationedacontroller_podman_mem_limit='400m'**.
3. Run the setup.

8.1.2.2. Modifying the default memory limit for each rulebook activation after installation

By default, each rulebook activation container has a 200MB memory limit. You can modify this default value after installation by using the following procedure:

Procedure

1. Navigate to the environment file at **/etc/ansible-automation-platform/eda**.

2. Modify the default container memory limit. For example, **EDA_PODMAN_MEM_LIMIT = '300m'**.

3. Restart the Event-Driven Ansible controller services using **systemctl restart automation-eda-controller.target**.

8.2. SYSTEM LEVEL MONITORING FOR EVENT-DRIVEN ANSIBLE CONTROLLER

After characterizing your workload to determine how many rulebook activations you are running in parallel and how many events you are receiving at any given point, you must consider monitoring your Event-Driven Ansible controller host at the system level. Using system level monitoring to review information about Event-Driven Ansible's performance over time helps when diagnosing problems or when considering capacity for future growth.

System level monitoring includes the following information:

- Disk I/O
- RAM utilization
- CPU utilization
- Network traffic

Higher CPU, RAM, or Disk utilization can affect the overall performance of Event-Driven Ansible controller. For example, a high utilization of any of these system level resources indicates that either the Event-Driven Ansible controller is running too many rulebook activations, or some of the individual rulebook activations are using a high volume of resources. In this case, you must increase your system level resources to support your workload.

8.3. PERFORMANCE TROUBLESHOOTING FOR EVENT-DRIVEN ANSIBLE CONTROLLER

Based on the default parameters within Event-Driven Ansible controller, you might encounter scenarios that pose challenges to completing your workload. The following section provides descriptions of these scenarios and troubleshooting guidance.

- My activation status displays as “running”, but it is not processing the events.
  - Ensure that you are using the correct event source in the rulebook activation. If the event you are expecting is coming from a source other than what is in the rulebook, Event-Driven Ansible controller will not process the event.
- My activation status displays as “running”, and Event-Driven Ansible controller is also receiving the events, but no actions are occuring.
  - Ensure that you have set the correct conditions for matching the event and taking actions in the rulebook activation.
- My activation keeps restarting in an infinite loop.
  - By default, the reset policy for rulebook activations is set to “on failure”. Change the estart policy using the following procedure:

1. Navigate to the Rulebook Activation screen.
2. Click the Restart Policy drop down menu.
3. Choose the appropriate value: “On Failure”, “Always”, “Never”.