



# Red Hat Ansible Automation Platform 2.5

## Using automation decisions

Configure and use Event-Driven Ansible controller to enhance and expand automation



## Red Hat Ansible Automation Platform 2.5 Using automation decisions

---

Configure and use Event-Driven Ansible controller to enhance and expand automation

## Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 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

<b>PREFACE</b> .....	<b>4</b>
<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b> .....	<b>5</b>
<b>CHAPTER 1. EVENT-DRIVEN ANSIBLE CONTROLLER OVERVIEW</b> .....	<b>6</b>
<b>CHAPTER 2. CREDENTIALS</b> .....	<b>7</b>
2.1. CREDENTIALS LIST VIEW	7
2.2. SETTING UP CREDENTIALS	7
2.3. EDITING A CREDENTIAL	8
2.4. DELETING A CREDENTIAL	8
<b>CHAPTER 3. CREDENTIAL TYPES</b> .....	<b>10</b>
3.1. CUSTOM CREDENTIAL TYPES	10
Input Configuration	10
Injector Configuration	11
3.2. CREATING A NEW CREDENTIAL TYPE	11
<b>CHAPTER 4. PROJECTS</b> .....	<b>14</b>
4.1. SETTING UP A NEW PROJECT	14
4.2. PROJECTS LIST VIEW	15
4.3. EDITING A PROJECT	15
4.4. DELETING A PROJECT	16
<b>CHAPTER 5. DECISION ENVIRONMENTS</b> .....	<b>17</b>
5.1. BUILDING A CUSTOM DECISION ENVIRONMENT FOR EVENT-DRIVEN ANSIBLE	17
5.2. SETTING UP A NEW DECISION ENVIRONMENT	18
<b>CHAPTER 6. SIMPLIFIED EVENT ROUTING</b> .....	<b>20</b>
6.1. EVENT STREAMS	20
6.2. CREATING AN EVENT STREAM CREDENTIAL	21
6.3. CREATING AN EVENT STREAM	22
6.4. CONFIGURING YOUR REMOTE SYSTEM TO SEND EVENTS	23
6.5. VERIFYING YOUR EVENT STREAMS WORK	24
6.6. REPLACING SOURCES AND ATTACHING EVENT STREAMS TO ACTIVATIONS	26
6.7. RESENDING WEBHOOK DATA FROM YOUR EVENT STREAM TYPE	29
6.8. CHECK THE RULE AUDIT FOR EVENTS ON YOUR NEW EVENT STREAM	30
<b>CHAPTER 7. RED HAT ANSIBLE AUTOMATION PLATFORM CREDENTIAL</b> .....	<b>31</b>
7.1. SETTING UP A RED HAT ANSIBLE AUTOMATION PLATFORM CREDENTIAL	31
<b>CHAPTER 8. RULEBOOK ACTIVATIONS</b> .....	<b>33</b>
8.1. SETTING UP A RULEBOOK ACTIVATION	34
8.2. RULEBOOK ACTIVATION LIST VIEW	36
8.2.1. Viewing activation output	36
8.3. ENABLING AND DISABLING RULEBOOK ACTIVATIONS	37
8.4. RESTARTING RULEBOOK ACTIVATIONS	37
8.5. DELETING RULEBOOK ACTIVATIONS	38
8.6. ACTIVATING WEBHOOK RULEBOOKS	38
8.7. TESTING WITH KUBERNETES	39
<b>CHAPTER 9. RULE AUDIT</b> .....	<b>40</b>
9.1. VIEWING RULE AUDIT DETAILS	40
9.2. VIEWING RULE AUDIT EVENTS	40

9.3. VIEWING RULE AUDIT ACTIONS	41
<b>CHAPTER 10. PERFORMANCE TUNING FOR EVENT-DRIVEN ANSIBLE CONTROLLER</b> .....	<b>42</b>
10.1. CHARACTERIZING YOUR WORKLOAD	42
10.1.1. Modifying the number of simultaneous rulebook activations	42
10.1.1.1. Modifying the number of simultaneous rulebook activations during Event-Driven Ansible controller installation	42
10.1.1.2. Modifying the number of simultaneous rulebook activations after Event-Driven Ansible controller installation	43
10.1.2. Modifying the default memory limit for each rulebook activation	43
10.1.2.1. Modifying the default memory limit for each rulebook activation during installation	43
10.1.2.2. Modifying the default memory limit for each rulebook activation after installation	43
10.2. SYSTEM LEVEL MONITORING FOR EVENT-DRIVEN ANSIBLE CONTROLLER	44
10.3. PERFORMANCE TROUBLESHOOTING FOR EVENT-DRIVEN ANSIBLE CONTROLLER	44
<b>CHAPTER 11. EVENT FILTER PLUGINS</b> .....	<b>46</b>
11.1. AUTHOR EVENT FILTERS	46
<b>CHAPTER 12. EVENT-DRIVEN ANSIBLE LOGGING STRATEGY</b> .....	<b>48</b>
12.1. LOGGING SAMPLES	48



## 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:

- [Credentials](#)
- [Credential types](#)
- [Projects](#)
- [Decision environments](#)
- [Simplified event routing](#)
- [Red Hat Ansible Automation Platform credential](#)
- [Rulebook activations](#)
- [Rule audit](#)
- [Performance tuning for Event-Driven Ansible controller](#)
- [Event filter plugins](#)
- [Event-Driven Ansible logging strategy](#)



## NOTE

- API documentation for Event-Driven Ansible controller is available at <https://<eda-server-host>/api/eda/v1/docs>
- To meet high availability demands, Event-Driven Ansible controller shares centralized [Redis \(REmote DIctionary Server\)](#) with the Ansible Automation Platform UI. When Redis is unavailable, you will not be able to create or sync projects, or enable rulebook activations.

## Additional resources

- For information on how to set user permissions for Event-Driven Ansible controller, see the following in the [Access management and authentication guide](#):
  1. [Adding roles for a user](#)
  2. [Roles](#)
- If you plan to use Event-Driven Ansible 2.5 with a 2.4 Ansible Automation Platform, see [Using Event-Driven Ansible 2.5 with Ansible Automation Platform 2.4](#).

## CHAPTER 2. CREDENTIALS

You can use credentials to store secrets that can be used for authentication purposes with resources, such as decision environments, rulebook activations and projects for Event-Driven Ansible controller, and projects for automation controller.

Credentials authenticate users when launching jobs against machines and importing project content from a version control system.

You can grant users and teams the ability to use these credentials without exposing the credential to the user. If a user moves to a different team or leaves the organization, you do not have to rekey all of your systems just because that credential was previously available.

### 2.1. CREDENTIALS LIST VIEW

When you log in to the Ansible Automation Platform and select **Automation Decisions** → **Infrastructure** → **Credentials**, the Credentials page has a pre-loaded **Decision Environment Container Registry** credential. When you create your own credentials, they will be added to this list view. .

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

You also have the following options in the menu bar:

- Choose how fields are shown in the list view by clicking the **Manage columns** icon. You have four options in which you can arrange your fields:
  - **Column** - Shows the column in the table.
  - **Description** - Shows the column when the item is expanded as a full width description.
  - **Expanded** - Shows the column when the item is expanded as a detail.
  - **Hidden** - Hides the column.
- Choose between a **List view** or a **Card view**, by clicking the icons.

### 2.2. SETTING UP CREDENTIALS

You can create a credential to use with a source plugin or a private container registry that you select. You can make your credential available to a team or individuals.

#### Procedure

1. Log in to the Ansible Automation Platform Dashboard.
2. From the navigation panel, select **Automation Decisions** → **Infrastructure** → **Credentials**.
3. Click **Create credential**.
4. Insert the following:

#### **Name**

Insert the name.

#### **Description**

This field is optional.

**Organization**

Click the list to select an organization or select **Default**.

**Credential type**

Click the list to select your Credential type.

**NOTE**

When you select the credential type, the **Type Details** section is displayed with fields that are applicable for the credential type you chose.

5. Complete the fields that are applicable to the credential type you selected.
6. 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.

## 2.3. EDITING A CREDENTIAL

You can edit existing credentials to ensure the appropriate level of access for your organization.



**Procedure**

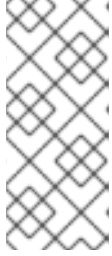
1. Edit the credential by using one of these methods:
  - From the **Credentials** list view, click the **Edit credential** icon next to the desired credential.
  - From the **Credentials** list view, select the name of the credential, click **Edit credential**.
2. Edit the appropriate details and click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

## 2.4. DELETING A CREDENTIAL

You can delete credentials if they are no longer needed for your organization.


**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**

**NOTE**

If your credential is still in use by other resources in your organization, a warning message is displayed letting you know that the credential cannot be deleted. Also, if your credential is being used in an event stream, you cannot delete it until the event stream is deleted or attached to a different credential. In general, avoid deleting a credential that is in use because it can lead to broken activations.

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. CREDENTIAL TYPES

Event-Driven Ansible controller comes with several built-in credential types that you can use for syncing projects, running rulebook activations, executing job templates through Automation Execution (automation controller), fetching images from container registries, and processing data through event streams.

These built-in credential types are not editable. So if you want credential types that support authentication with other systems, you can create your own credential types that can be used in your source plugins. Each credential type contains an input configuration and an injector configuration that can be passed to an Ansible rulebook to configure your sources.

For more information, see [Custom credential types](#).

### 3.1. CUSTOM CREDENTIAL TYPES

As a system administrator, you can define a custom credential type that works in ways similar to existing credential types in a standard format using a YAML or JSON-like definition.

Each credential type displays its own unique configurations in the Input Configuration field and the Injector Configuration field, if applicable. Custom credentials support Ansible extra variables as a means of injecting their authentication information.

You can attach one or more cloud, vault, and Red Hat Ansible Automation Platform credentials to a rulebook activation.



#### NOTE

- When creating a new credential type, you must avoid collisions in the **extra\_vars**.
- Extra variable names must not start with **EDA\_** because they are reserved.
- You must have System administrator (superuser) permissions to be able to create and edit a credential type and to be able to view the **Injector configuration** field.

When you customize your own credential types, they will display on the Credential Types page along with a list of built-in credential types.

Each credential type displays its own unique configurations in the Input Configuration and the Injector Configuration fields, if applicable. Both YAML and JSON formats are supported in the configuration fields.

#### Input Configuration

The Input configuration has two attributes:

- **fields** - a collection of properties for a credential type.
- **required** - a list of required fields.

Fields can have multiple properties, depending on the credential type you select.

**Table 3.1. Input Configuration Field Properties**

Fields	Description	Mandatory (Y/N)
<b>id</b>	Unique id of the field; must be a string type and stores the variable name	Yes
<b>type</b>	Can be string or boolean type	No, default is string
<b>label</b>	Used by the UI when rendering the UI element	Yes
<b>secret</b>	Will be encrypted	No, default false
<b>multiline</b>	If the field contains data from a file the multiline can be set to True	No, default false
<b>help_text</b>	The help text associated with this field	No

### Injector Configuration

You can use the Injector configuration field to take the fields from input configuration field and map them into **extra\_vars** that can be sent to ansible-rulebook when running the activation. The Injector currently only supports **extra\_vars**.

Injectors enable you to tailor the fields so that they can be injected into a rulebook as **extra\_vars**, which cannot have duplicate keys at the top level. If you have two sources in a rulebook that both require a parameter called username and password, the injectors, along with the rulebook, help you tailor the arguments for each source.

## 3.2. CREATING A NEW CREDENTIAL TYPE

You can create a credential type to use with a source plugin that you select based on the supported, default credential types. You can make your credential type available to a team or individuals.

### Procedure

1. Log in to the Ansible Automation Platform Dashboard.
2. From the navigation panel, select **Automation Decisions** → **Infrastructure** → **Credential Types**.
3. Click **Create credential type**.
4. Insert the following:

#### Name

Insert the name.

#### Description

This field is optional.

- In the **Input Configuration** field, specify an input schema that defines a set of ordered fields for that type. The format can be in YAML or JSON:

#### YAML

```
fields:
  - type: string
    id: username
    label: Username
  - type: string
    id: password
    label: Password
    secret: true
required:
  - username
  - password
```

View more YAML examples at the [YAML page](#).

#### JSON

```
{
  "fields": [
    {
      "type": "string",
      "id": "username",
      "label": "Username"
    },
    {
      "secret": true,
      "type": "string",
      "id": "password",
      "label": "Password"
    }
  ],
  "required": ["username", "password"]
}
```

View more JSON examples at [The JSON website](#).

- In the **Injector Configuration** field, enter environment variables or extra variables that specify the values a credential type can inject. The format can be in YAML or JSON (see examples in the previous step).

The following configuration in JSON format shows each field and how they are used:

```
{
  "extra_vars": {
    "some_extra_var": "{{ username }}:{{ password }}"
  }
}
```

- Click **Create credential type**.

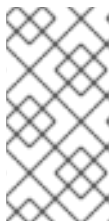
Your newly created credential type is displayed in the list of credential types:



## Credential Types

Name	Actions
Another new credential type	
New credential type	
new_cred_type	

8. Click the **Edit credential type** icon to modify the credential type options.

**NOTE**

On the **Edit** page, you can modify the details or delete the credential. If the **Delete** option is disabled, this means that the credential type is being used by a credential, and you must delete the credential type from all the credentials that use it before you can delete it.

**Verification**

- Verify that the newly created credential type can be selected from the **Credential Type** selection window when creating a new credential:

**Create New Credential**

Name \*  Description  Organization

Credential Type \*

- Microsoft Azure Resource Manager
- Network
- New credential type
- new\_cred\_type
- OpenShift or Kubernetes API Bearer Token
- OpenStack
- Red Hat Ansible Automation Platform

**Additional resources**

For information about how to create a new credential, see [Setting up credentials](#).

## CHAPTER 4. 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.



### IMPORTANT

To meet high availability demands, Event-Driven Ansible controller shares centralized [Redis \(REmote DIctionary Server\)](#) with the Ansible Automation Platform UI. When Redis is unavailable, you will not be able to create or sync projects.

### 4.1. SETTING UP A NEW PROJECT

You can set up projects to manage and store your rulebooks in Event-Driven Ansible controller.

#### Prerequisites

- You are logged in to the Ansible Automation Platform 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 Ansible Automation Platform Dashboard.
2. Navigate to **Automation Decisions** → **Projects**.
3. Click **Create project**.
4. Insert the following:

##### Name

Enter project name.

##### Description

This field is optional.

##### Source control type

Git is the only source control type available for use. This field is optional.

##### Source control URL

Enter Git, SSH, or HTTP[S] protocol address of a repository, such as GitHub or GitLab. This field is not editable.



### NOTE

This field accepts SSH private key or private key phrase. To enable the use of these private keys, your project URL must begin with **git@**.

**Proxy**

This is used to access access HTTP or HTTPS servers. This field is optional.

**Source control branch/tag/commit**

This is the branch to checkout. In addition to branches, you can input tags, commit hashes, and arbitrary refs. Some commit hashes and refs may not be available unless you also provide a custom refs spec. This field is optional.

**Source control refs spec**

A refs spec to fetch (passed to the Ansible git module). This parameter allows access to references via the branch field not otherwise available. This field is optional. For more information, see [Examples](#).

**Source control credential**

You must have this credential to utilize the source control URL. This field is optional.

**Content signature validation credential**

Enable content signing to verify that the content has remained secure when a project is synced. If the content has been tampered with, the job will not run. This field is optional.

**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.

5. Select **Create project**.

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

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.

## 4.2. 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.

## 4.3. EDITING A PROJECT

**Procedure**



1. From the **Projects** list view, select the **More Actions** icon  next to the desired project. The Edit page is displayed.

2. Enter the required changes and select **Save project**.

## 4.4. DELETING A PROJECT

If you need to delete a project, the Event-Driven Ansible controller interface provides multiple options.

### Procedure

1. To delete a project, complete one of the following:
  - From the **Projects** list view, select the checkbox next to the desired project, and click the **More Actions** icon  from the page menu.
  - From the **Projects** list view, click the **More Actions** icon  next to the desired project.
2. Select **Delete project**.
3. In the **Permanently delete projects** window, select **Yes, I confirm that I want to delete this project**.
4. Select **Delete project**.

## CHAPTER 5. DECISION ENVIRONMENTS

Decision environments are a container image to run Ansible Rulebook rulebooks. They create a common language for communicating automation dependencies, and give a standard way to build and distribute the automation environment. You can find the default decision environment in the [Ansible-Rulebook](#).

To create your own decision environment see [Building a custom decision environment for Event-Driven Ansible within Ansible Automation Platform](#).

### 5.1. BUILDING A CUSTOM DECISION ENVIRONMENT FOR EVENT-DRIVEN ANSIBLE

Decision Environments are execution environments tailored towards running Ansible Rulebooks.

Similar to execution environments that run Ansible rulebooks for automation controller, decision environments are designed to run rulebooks for Event-Driven Ansible controller.

You can create a custom decision environment for Event-Driven Ansible that provides a custom maintained or third-party event source plugin that is not available in the default decision environment.

#### Prerequisites

- Ansible Automation Platform > = 2.5
- 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** at [Ansible Automation Platform supported decision environment](#).



#### 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
```

## 5.2. 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 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](https://registry.redhat.io).

### Procedure

1. Log in to Ansible Automation Platform.
2. Navigate to **Automation Decisions** → **Decision Environments**.
3. Click **Create decision environment**.
4. Insert the following:

#### Name

Insert the name.

#### Description

This field is optional.

**Organization**

Select an organization to associate with the decision environment.

**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.

5. Select **Create decision environment**.

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.

## CHAPTER 6. SIMPLIFIED EVENT ROUTING

Simplified event routing enables Event-Driven Ansible controller to capture and analyze data from various remote systems using event streams. With event streams, you can send events from a remote system like GitHub or GitLab into Event-Driven Ansible controller. You can attach 1 or more event streams to an activation by swapping out sources in a rulebook.

Event streams are an easy way to connect your sources to your rulebooks. This capability lets you create a single endpoint to receive alerts from an event source and then use the events in multiple rulebooks.

### 6.1. EVENT STREAMS

Event streams can send events from remote systems to Event-Driven Ansible controller. In a typical set-up, a server sends data to an event stream over the internet to an Event-Driven Ansible event stream receiver. When the data comes over the internet, the request must be authenticated. Depending on the webhook vendor or remote system, the authentication method could differ.

Event-Driven Ansible controller supports 7 different event stream types.

**Table 6.1. Event Stream Types**

Type	Description	Vendors
<b>HMAC</b>	Hashed Message Authentication Code (HMAC). Uses a shared secret between Event-Driven Ansible controller and the vendors webhook server. This guarantees message integrity.	Github
<b>Basic Authentication</b>	Uses HTTP basic authentication.	Datadog, Dynatrace
<b>Token Authentication</b>	Uses Token Authentication. Usually the HTTP Header is <b>Authorization</b> but some vendors like Gitlab use <b>X-Gitlab-Token</b> .	Gitlab, ServiceNow
<b>OAuth2</b>	Uses Machine-to-Machine (M2M) mode with a grant type called <b>client credentials</b> . The token is opaque.	Dynatrace
<b>OAuth2 with JWT</b>	Uses M2M mode with a grant type called <b>client credentials</b> . The token is JSON Web Token (JWT).	Datadog
<b>ECDSA</b>	Elliptic Curve Digital Signature Algorithm	SendGrid, Twilio



Type	Description	Vendors
<b>Mutual TLS</b>	Needs the vendor's CA certificate to be present in our servers at startup. This supports non-repudiation.	PagerDuty

Event-Driven Ansible controller also supports 4 other specialized event streams that are based on the 7 basic event stream types:

- GitLab Event Stream
- GitHub Event Stream
- ServiceNow Event Stream
- Dynatrace Event Stream

These specialized types limit the parameters you use by adding default values. For example, the GitHub Event Stream is a specialization of the HMAC Event Stream with many of the fields already populated. After the GitHub Event Stream credential has been saved, the recommended defaults for the GitHub Event Stream are displayed.

## 6.2. CREATING AN EVENT STREAM CREDENTIAL

You must create an event stream credential first before you can use an event stream.

### Prerequisites

- Each event stream must have exactly one credential.

### Procedure

1. Log in to the Ansible Automation Platform Dashboard.
2. From the navigation panel, select **Automation Decisions** → **Infrastructure** → **Credentials**.
3. Click **Create credential**.
4. Insert the following:

#### **Name**

Insert the name.

#### **Description**

This field is optional.

#### **Organization**

Click the list to select an organization or select **Default**.

#### **Credential type**

Click the list to select your Credential type.

**NOTE**

When you select the credential type, the **Type Details** section is displayed with fields that are applicable for the credential type you selected.

**Type Details**

Add the requested information for the credential type you selected. For example, if you selected the GitHub Event Stream credential type, you are required to add an HMAC Secret (symmetrical shared secret) between Event-Driven Ansible controller and the remote server.

5. Click **Create credential**.

The Details page is displayed. From there or the **Credentials** list view, you can edit or delete it.

**6.3. CREATING AN EVENT STREAM**

You can create event streams that will be attached to a rulebook activation.

**Prerequisites**

- If you will be attaching your event stream to a rulebook activation, ensure that your activation has a decision environment and project already set up.
- If you plan to connect to automation controller to run your rulebook activation, ensure that you have created a Red Hat Ansible Automation Platform credential type in addition to the decision environment and project. For more information, see [Setting up a Red Hat Ansible Automation Platform credential](#).

**Procedure**

1. Log in to Ansible Automation Platform.
2. From the navigation panel, select **Automation Decisions → Event Streams**.
3. Click **Create event stream**.
4. Insert the following:

**Name**

Insert the name.

**Organization**

Click the list to select an organization or select **Default**.

**Event stream type**

Select the event stream type you prefer.

**NOTE**

This list displays at least 10 default event stream types that can be used to authenticate the connection coming from your remote server.

**Credentials**

Select a credential from the list, preferably the one you created for your event stream.

### Headers

Enter HTTP header keys, separated by commas, that you want to include in the event payload. To include all headers, leave the field empty.

### Forward events to rulebook activation

Use this option to enable or disable the capability of forwarding events to rulebook activations.



### NOTE

The event stream's event forwarding can be disabled for testing purposes while diagnosing connections and evaluating the incoming data. Disabling the **Forward events to rulebook activation** option allows you to test the event stream connection with the remote system, analyze the header and payload, and if necessary, diagnose credential issues. This ensures that events are not be forwarded to rulebook activations causing rules and conditions to be triggered inadvertently while you are in test mode. Some enterprises might have policies to change secrets and passwords at regular cadence. You can enable/disable this option anytime after the event stream is created.

## 5. Click **Create event stream**.

After creating your event stream, the following outputs occur:

- The Details page is displayed. From there or the Event Streams list view, you can edit or delete it. Also, the Event Streams page shows all of the event streams you have created and the following columns for each event: **Events received**, **Last event received**, and **Event stream type**. As the first two columns receive external data through the event stream, they are continuously updated to let you know they are receiving events from remote systems.
- If you disabled the event stream, the Details page is displayed with a warning message, **This event stream is disabled**.
- Your new event stream generates a URL that is necessary when you configure the webhook on the remote system that sends events.



### NOTE

After an event stream is created, the associated credential cannot be deleted until the event stream it is attached to is deleted.

## 6.4. CONFIGURING YOUR REMOTE SYSTEM TO SEND EVENTS

After you have created your event stream, you must configure your remote system to send events to Event-Driven Ansible controller. The method used for this configuration varies, depending on the vendor for the event stream credential type you select.

### Prerequisites

- The URL that was generated when you created your event stream
- Secrets or passwords that you set up in your event stream credential

## Procedure

The following example demonstrates how to configure webhooks in a remote system like GitHub to send events to Event-Driven Ansible controller. Each vendor will have unique methods for configuring your remote system to send events to Event-Driven Ansible controller.

1. Log in to your GitHub repository.
2. Click **Your profile name → Your repositories**



### NOTE

If you do not have a repository, click **New** to create a new one, select an owner, add a **Repository name**, and click **Create repository**.

1. Navigate to **Settings** (tool bar).
2. In the **General** navigation pane, select **Webhooks**.
3. Click **Add webhook**.
4. In the **Payload URL** field, paste the URL you saved when you created your event stream.
5. Select **application/json** in the **Content type** list.
6. Enter your **Secret**.
7. Click **Add webhook**.

After the webhook has been added, it attempts to send a test payload to ensure there is connectivity between the two systems (GitHub and Event-Driven Ansible controller). If it can successfully send the data you will see a green check mark next to the **Webhook URL** with the message, **Last delivery was successful**.

## 6.5. VERIFYING YOUR EVENT STREAMS WORK

Verify that you can use your event stream to connect to a remote system and receive data.

1. Log in to Ansible Automation Platform.
2. From the navigation panel, select **Automation Decisions → Event Streams**.
3. Select the event stream that you created to validate connectivity and ensure that the event stream sends data to the rulebook activation.
4. Verify that the events were received. You can see in the **Events received** field that the event was received. You can also see the header for the event stream that contains details about the event.

Event Streams > Test-event-stream > Details

Test-event-stream  Forward events to rulebook activation [Edit event stream](#)

Back to Event Streams Details Activations Team Access User Access

**⚠ This event stream is disabled.**  
Event streams that are disabled do not forward events to the rulebook activation where they are configured. To forward events to the rulebook activation, enable the forwarding of events.

Name	Event stream type	Organization
Test-event-stream	github	Default
Credential	URL	Events received
<a href="#">Test-event-stream-cred</a>	https://54.173.8.216/eda-event-st...	0
Created	Last modified	
9/20/2024, 5:50:27 PM	9/20/2024, 5:52:05 PM	

If you scroll down in the UI, you can also see the body of the payload with more information about the webhook.

```
X-Github-Hook-Installation-Target-Id: '626071641'
X-Github-Hook-Installation-Target-Type: repository
X-Hub-Signature: sha1=da9149ee1288c98272614af388bfd28d65982d5c
X-Hub-Signature-256: sha256=405246974ade9303ca3725851dacdec1993d8ccbf51d4749df33c81bcc1d3ecc
X-Request-Id: 0430cd80-c7ee-43ea-bbd2-e27a202aa583
```

**Body**

```
hook:
  active: true
  config:
    content_type: json
    insecure_ssl: '1'
    secret: '*****'
    url: https://aap-aap-hackathon-11.apps.hackathon.ocp4.testing.ansible.com/eda-event-streams/api/eda/v1/external_event_stream/9be5be26-5e86-469c-a361-382b65e1d2d0/post/
    created_at: '2024-09-05T19:26:45Z'
    delivers_url: https://api.github.com/repos/mkanoor/eda-project/hooks/500085791/deliveries
  events:
    - push
    id: 500085791
    last_response:
      code: null
      message: null
      status: unused
    name: web
    ping_url: https://api.github.com/repos/mkanoor/eda-project/hooks/500085791/pings
    test_url: https://api.github.com/repos/mkanoor/eda-project/hooks/500085791/test
    type: Repository
```

The **Header** and **Body** sections for the event stream are displayed on the Details page. They differ based on the vendor who is sending the event. The header and body can be used to check the attributes in the event payload, which will help you in writing conditions in your rulebook. For example:

5. Toggle the **Forward events to rulebook activation** option to enable you to push your events to a rulebook activation. This moves the event stream to production mode and makes it easy to attach to rulebook activations.

When this option is toggled off, your ability to forward events to a rulebook activation is disabled and the **This event stream is disabled** message is displayed.

## 6.6. REPLACING SOURCES AND ATTACHING EVENT STREAMS TO ACTIVATIONS

When you create rulebook activations, you can use event streams to swap out source mappings in rulebook activations and simplify routing from external sources to Event-Driven Ansible controller.

There are several key points to keep in mind regarding source mapping:

1. An event stream can only be used once in a rulebook source swap. If you have multiple sources in the rulebook, you can only replace each source once.
2. The source mapping happens only in the current rulebook activation. You must repeat this process for any other activations using the same rulebook.
3. The source mapping is valid only if the rulebook doesn't get modified. If the rulebook gets modified during the source mapping process, the source mapping would fail and it would have to be repeated.
4. If the rulebook is modified after the source mapping has been created and a **Restart** happens, the rulebook activation fails.

### Procedure

1. Log in to Ansible Automation Platform.
2. From the navigation panel, select **Automation Decisions** → **Rulebook Activations**.
3. Click **Create rulebook activation**.
4. Insert the following:

#### Name

Insert the name.

#### Description

This field is optional.

#### Organization

Enter your organization name or select Default from the list.

#### Project

Projects are a logical collection of rulebooks. This field is optional.



#### NOTE

Although this field is optional, selecting a project helps refine your list of rulebooks choices.

#### Rulebook

Rulebooks are shown according to the project selected. Select a rulebook.



#### NOTE

After you have selected a rulebook, the Event streams field is enabled. You can click the gear icon to display the Event streams mapping form.

## Event streams

All the event streams available and set up to forward events to rulebook activations are displayed. If you have not created any event streams, this field remains disabled.

Click the gear icon to display the Event streams mapping UI.

Complete the following fields:

### Rulebook source

A rulebook can contain multiple sources across multiple rulesets. You can map the same rulebook in multiple activations to multiple event streams. While managing event streams, unnamed sources are assigned temporary names (`__SOURCE {n}`) for identification purposes.

Select `__SOURCE_1` from the list.

### Event stream

Select your event stream name from the list.

Click **Save**.

Event streams can replace matching sources in a rulebook, and are server-side webhooks that enable you to connect various event sources to your rulebook activations. Source types that can be replaced with the event stream's source of type `ansible.eda.pg_listener` include `ansible.eda.webhook` and other compatible webhook source plugins. Replacing selected sources affects this activation only, and modifies the rulebook's source type, source name, and arguments. Filters, rules, conditions, and actions are all unaffected.

You can select which source you want to replace with a single event stream. If there are multiple sources in your rulebook, you can choose to replace each one of them with event streams, but you are not required to replace each one. The following image displays which sources can be replaced.

```

1 ---
2 - name: Github Webhook Ruleset
3   hosts: all
4   sources:
5     - name: my_github
6       ansible.eda.webhook:
7         port: 5555
8         host: 0.0.0.0
9       filters:
10        - ansible.eda.json_filter:
11          include_keys:
12            - payload
13            - repository
14            - name
15            - full_name
16            - url
17            - sender
18            - login
19            - login
20            - zen
21          exclude_keys:
22            - "*"
23        rules:
24          - name: Webhook event
25            condition: true
26            action:
27              debug:
28
29 - name: SNOW Webhook Ruleset
30   hosts: all
31   sources:
32     - name: my_snow
33       ansible.eda.webhook:
34         port: 5556
35         host: 0.0.0.0
36       filters:
37        - ansible.eda.json_filter:
38          include_keys:
39            - payload
40            - approval
41            - sys_class_name
42            - sys_created_by
43            - category
44            - urgency
45          exclude_keys:
46            - "*"
47        rules:
48          - name: Show webhook event
49            condition: true
50            action:
51              debug:

```

The items in pink demonstrate the sources that can be replaced: source type, source name, and arguments. The remaining items (filters, rules, and actions) are not replaced.

## 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 on credentials, see [Credentials](#).

## Decision environment

A decision environment is a container image used 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.

- Policies:
  - i. **Always:** This restarts the rulebook activation immediately, regardless of whether it ends successfully or not, and occurs no more than 5 times.
  - ii. **Never:** This never restarts a rulebook activation when the container process ends.
  - iii. **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.

- Levels:
  - i. **Error:** Logs that contain error messages that are displayed in the **History** tab of an activation.
  - ii. **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.
  - iii. **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?

This automatically enables the rulebook activation to run.

### Variables

The variables for the rulebook are in a 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.

5. Click **Create rulebook activation**.

After you create your rulebook activation, the **Details** page is displayed.

You can navigate to the Event streams page to confirm your events have been received.

## 6.7. RESENDING WEBHOOK DATA FROM YOUR EVENT STREAM TYPE

After you have replaced your sources with the event stream you created, you can now resend data from the event stream to ensure that it is attached to your rulebook activation. In the example shared earlier, the GitHub event stream was used. The following example demonstrates how to resend webhook data if you were using a GitHub event stream.

### Procedure

1. Go back to the **GitHub Webhook / Manage webhook** page.
2. Click the **Recent Deliveries** tab.
3. Click the **ellipsis**.
4. Click **Redeliver**. A **Redeliver payload?** window is displayed with a delivery message.
5. Click **Yes, redeliver this payload**.
6. Return to the Ansible Automation Platform to check your rule audit.

## 6.8. CHECK THE RULE AUDIT FOR EVENTS ON YOUR NEW EVENT STREAM

When events have been sent and received by Event-Driven Ansible controller, you can confirm that actions have been triggered by going to the Rule Audit page and viewing the event stream results.

### Procedure

1. Log in to Ansible Automation Platform.
2. From the navigation panel, select **Automation Decisions** → **Rule Audit**.  
If your rulebook activation received the event data from the event stream type you selected, the Rule Audit page displays the results similar to this image.

Name	Status	Rulebook activation	Last fired date
<a href="#">Webhook rule</a>	✔ Success	<a href="#">my-es-demo</a>	9/26/2024, 5:29:20 PM

## CHAPTER 7. RED HAT ANSIBLE AUTOMATION PLATFORM CREDENTIAL

The Red Hat Ansible Automation Platform credential type can connect to automation controller through the use of an automation controller URL and a username and password. After you have created this credential type, it can be attached to a rulebook and used to run rulebook activations.

### 7.1. SETTING UP A RED HAT ANSIBLE AUTOMATION PLATFORM CREDENTIAL

You can create a Red Hat Ansible Automation Platform credential type to run your rulebook activations.

#### Prerequisites

- You have created a user.
- You have obtained the URL and the credentials to access automation controller.

#### Procedure

1. Log in to the Ansible Automation Platform.
2. From the navigation panel, select **Automation Decisions** → **Infrastructure** → **Credentials**.
3. Click **Create credential**.
4. Insert the following:

##### Name

Insert the name.

##### Description

This field is optional.

##### Organization

Click the list to select an organization or select **Default**.

##### Credential type

Click the list and select **Red Hat Ansible Automation Platform**



#### NOTE

When you select the credential type, the **Type Details** section is displayed with fields that are applicable for the credential type you chose.

5. In the required Red Hat Ansible Automation Platform field, enter your automation controller URL.



## NOTE

For Event-Driven Ansible controller 2.5 with automation controller 2.4, use the following example: `https://<your_controller_host>`

For Ansible Automation Platform 2.5, use the following example:  
`https://<your_gateway_host>/api/controller/`

6. Enter a valid **Username** and **Password** or **Oauth Token**.
7. Click **Create credential**.

After you create this credential, you can use it for configuring your rulebook activations.

## CHAPTER 8. RULEBOOK ACTIVATIONS

A rulebook is a set of conditional rules that Event-Driven Ansible uses to perform IT actions in an event-driven automation model. Rulebooks are the means by which users tell Event-Driven Ansible which source to check for an event and when that event occurs what to do when certain conditions are met.

A rulebook specifies actions to be performed when a rule is triggered. A rule gets triggered when the events match the conditions for the rules. The following actions are currently supported:

- **run\_playbook** (only supported with ansible-rulebook CLI)
- **run\_module**
- **run\_job\_template**
- **run\_workflow\_template**
- **set\_fact**
- **post\_event**
- **retract\_fact**
- **print\_event**
- **shutdown**
- **debug**
- **none**

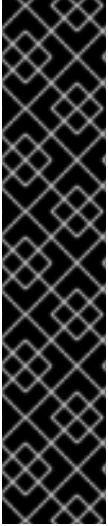
To view further details, see [Actions](#).

A rulebook activation is a process running in the background defined by a decision environment executing a specific rulebook. You can set up your rulebook activation by following [Setting up a rulebook activation](#).



### WARNING

Red Hat does not recommend the use of a non-supported source plugin with 1 postgres database. This can pose a potential risk to your use of Ansible Automation Platform.



## IMPORTANT

To meet high availability demands, Event-Driven Ansible controller shares centralized [Redis \(REmote DIctionary Server\)](#) with the Ansible Automation Platform UI. When Redis is unavailable, the following functions will not be available:

- Creating an activation, if **is\_enabled** is True
- Deleting an activation
- Enabling an activation, if not already enabled
- Disabling an activation, if not already disabled
- Restarting an activation

## 8.1. SETTING UP A RULEBOOK ACTIVATION

### Prerequisites

- You are logged in to the Ansible Automation Platform Dashboard as a Content Consumer.
- You have set up a project.
- You have set up a decision environment.

### Procedure

1. Log in to Ansible Automation Platform.
2. Navigate to the **Automation Decisions → Rulebook Activations**.
3. Click **Create rulebook activation**.
4. Insert the following:

#### Name

Insert the name.

#### Description

This field is optional.

#### Organization

Enter your organization name or select Default from the list.

#### Project

Projects are a logical collection of rulebooks. This field is optional.

#### Rulebook

Rulebooks are displayed according to the project 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**

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 the policy that determines how an activation should restart after the container process running the source plugin ends.

- Policies:
  - i. **Always:** This restarts the rulebook activation immediately, regardless of whether it ends successfully or not, and occurs no more than 5 times.
  - ii. **Never:** This never restarts a rulebook activation when the container process ends.
  - iii. **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.

- Levels:
  - i. **Error:** Logs that contain error messages that are displayed in the **History** tab of an activation.
  - ii. **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.
  - iii. **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?**

This automatically enables the rulebook activation to run.

**Variables**

The variables for the rulebook are in a 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.

5. 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, with either a **Pending**, **Running**, or **Failed** status. From there or the **Rulebook Activations** list view, you can restart 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 is sent to the activation log. For more information, see [Rulebooks](#).

## 8.2. RULEBOOK ACTIVATION LIST VIEW

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

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.

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.

### 8.2.1. 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 shows you the **Output** produced by that specific execution.

The screenshot shows the Red Hat Ansible Automation Platform interface. The left sidebar contains navigation options: Event-Driven Ansible, Dashboard, Views (Rule Audit, Rulebook Activations), Resources (Projects, Decision Environments, Credentials), and User Access (Users, Roles). The main content area displays the details for '001 - Activation 1'. It includes a breadcrumb trail: Rulebook Activations > Activation 1 > History > 001 - Activation 1 > Details. Below the breadcrumb, there is a table with columns for Name, Activation status, and Start date. The table shows one row: 001 - Activation 1, Running, 03/27/2022 8:30:00 AM. Below the table, there is an 'Output' section with a scrollable list of 10 lines of text: 1 expire\_time: 1453154676, 2 vmware\_host: cent7issue, 3 vmware\_host: cent7issue, 4 vmware\_host: cent7issue, 5 vmware\_host: cent7issue, 6 vmware\_host: cent7issue, 7 vmware\_host: cent7issue, 8 vmware\_host: cent7issue, 9 vmware\_host: cent7issue, 10 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.

## 8.3. ENABLING AND DISABLING RULEBOOK ACTIVATIONS

### 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.**


## 8.4. 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.

### 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**.

## 8.5. DELETING RULEBOOK ACTIVATIONS

### 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**.

## 8.6. 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.



### 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).

## 8.7. 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 '{}'
```

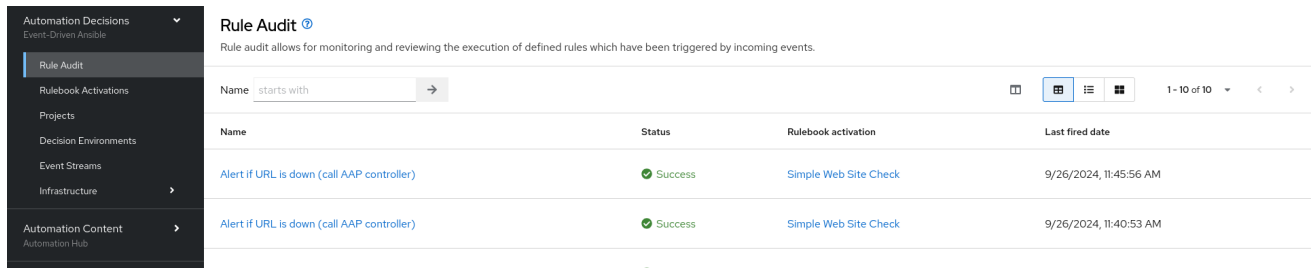
## CHAPTER 9. RULE AUDIT

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.

### 9.1. VIEWING RULE AUDIT DETAILS

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



The screenshot shows the 'Rule Audit' page in the Ansible Automation Platform. The left navigation panel is open, showing 'Automation Decisions' and 'Rule Audit' selected. The main content area displays a table of rule activations. The table has four columns: Name, Status, Rulebook activation, and Last fired date. Two rows are visible, both showing a 'Success' status for the rule 'Alert if URL is down (call AAP controller)' with the activation 'Simple Web Site Check' and a last fired date of '9/26/2024, 11:45:56 AM' and '9/26/2024, 11:40:53 AM' respectively.

Name	Status	Rulebook activation	Last fired date
Alert if URL is down (call AAP controller)	Success	Simple Web Site Check	9/26/2024, 11:45:56 AM
Alert if URL is down (call AAP controller)	Success	Simple Web Site Check	9/26/2024, 11:40:53 AM

#### Procedure

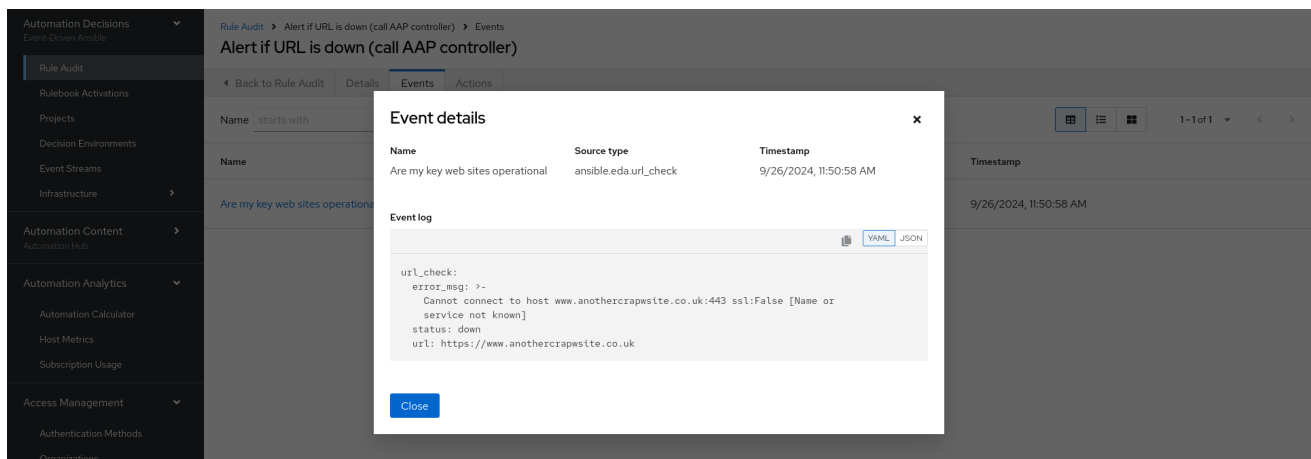
1. From the navigation panel select **Automation Decisions** → **Rule Audit**.
2. Select the desired rule, this brings you to the **Details** tab.

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

### 9.2. VIEWING RULE AUDIT EVENTS

#### Procedure

1. From the navigation panel, select **Automation Decisions** → **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**.



The screenshot shows the 'Rule Audit' page with the 'Events' tab selected. A modal window titled 'Event details' is open, displaying information for a specific event. The event name is 'Are my key web sites operational', the source type is 'ansible.eda.url\_check', and the timestamp is '9/26/2024, 11:50:58 AM'. The event log shows a failure: 'url\_check: error\_msg: - Cannot connect to host www.anothercrapsite.co.uk:443 ssl:False [Name or service not known] status: down url: https://www.anothercrapsite.co.uk'. The modal also has 'YAML' and 'JSON' tabs and a 'Close' button.

Name	Source type	Timestamp
Are my key web sites operational	ansible.eda.url_check	9/26/2024, 11:50:58 AM

```

url_check:
  error_msg: -
    Cannot connect to host www.anothercrapsite.co.uk:443 ssl:False [Name or
    service not known]
  status: down
  url: https://www.anothercrapsite.co.uk
  
```

## 9.3. VIEWING RULE AUDIT ACTIONS

### Procedure

1. From the navigation panel select **Automation Decisions → Rule Audit**.
2. Select the desired rule, then select the **Actions** tab.

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

## CHAPTER 10. 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

### 10.1. 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

#### 10.1.1. Modifying the number of simultaneous rulebook activations

By default, Event-Driven Ansible controller allows 12 rulebook activations to run simultaneously. If more than 12 rulebook activations are created, the expected behavior is that subsequent rulebook activations wait until there is an available rulebook activation worker. In this case, the rulebook activation status is displayed 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 **MAX\_RUNNING\_ACTIVATIONS** does not change when you modify the instance size, so it needs to be adjusted manually.

##### 10.1.1.1. 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.

### 10.1.1.2. 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/settings.yaml**.
2. Choose the number of maximum running activations that you need. For example, **MAX\_RUNNING\_ACTIVATIONS = 16**
3. Use the following command to restart Event-Driven Ansible services: **automation-eda-controller-service restart**

#### Resources

For more information about rulebook activations, see the [Rulebook activations](#).

### 10.1.2. 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

#### 10.1.2.1. 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.

#### 10.1.2.2. 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/settings.yaml`.
2. Modify the default container memory limit. For example, `PODMAN_MEM_LIMIT = '300m'`.
3. Restart the Event-Driven Ansible controller services using `automation-eda-controller-service restart`.

## 10.2. 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.

## 10.3. 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 occurring.
  - 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 restart



policy using the following procedure:

1. Navigate to **Automation Decisions** → **Rulebook Activations**.
2. Select the **Restart Policy** list to display the options.
3. Select the appropriate value: **On Failure, Always, Never**.

## CHAPTER 11. EVENT FILTER PLUGINS

Events sometimes have extra data that is unnecessary and might overwhelm the rule engine. Use event filters to remove that extra data so you can focus on what matters to your rules. Event filters might also change the format of the data so that the rule conditions can better match the data.

Events are defined as python code and distributed as collections. The default [eda collection](#) has the following filters:

Name	Description
<code>json_filter</code>	This filter includes and excludes keys from the event object
<code>dashes_to_underscores</code>	This filter changes the dashes in all keys in the payload to be underscore
<code>ansible.eda.insert_hosts_to_meta</code>	This filter is used to add host information into the event so that ansible-rulebook can locate it and use it
<code>ansible.eda.normalize_keys</code>	This filter is used if you want to change non alpha numeric keys to underscore

You can chain event filters one after the other, and the updated data is sent from one filter to the next. Event filters are defined in the rulebook after a source is defined. When the rulebook starts the source plugin it associates the correct filters and transforms the data before putting it into the queue.

### Example

```
sources:
- name: azure_service_bus
  ansible.eda.azure_service_bus:
    conn_str: "{{connection_str}}"
    queue_name: "{{queue_name}}"
filters:
- json_filter:
  include_keys: ['clone_url']
  exclude_keys: ['*_url', '_links', 'base', 'sender', 'owner', 'user']
- dashes_to_underscores:
```

In this example the data is first passed through the **json\_filter** and then through the **dashes\_to\_underscores** filter. In the event payload, keys can only contain letters, numbers, and underscores. The period (.) is used to access nested keys.

Since every event should record the origin of the event the filter **eda.builtin.insert\_meta\_info** is added automatically by ansible-rulebook to add the **source name, type**, and **received\_at**. The **received\_at** stores a date time in UTC ISO8601 format and includes the microseconds. The **uuid** stores the unique id for the event. The **meta key** is used to store metadata about the event and its needed to correctly report about the events in the aap-server.

### 11.1. AUTHOR EVENT FILTERS

Event filters are functions in a python module that perform transformations on the event data. They can remove, add, change, or move any data in the event data structure. Event filters take the event as the first argument and additional keyword arguments are provided by the configuration in the rulebook.

The basic structure follows:

```
# my_namespace.my_collection/extensions/eda/plugins/event_filter/my_filter.py
def main(event: dict, arg1, arg2):
    # Process event data here
    return event
```

You can use this filter in a rulebook by adding it to the filters list in an event source:

```
sources:
- name: azure_service_bus
  ansible.eda.azure_service_bus:
    conn_str: "{{connection_str}}"
    queue_name: "{{queue_name}}"
  filters:
- my_namespace.my_collection.my_filter:
    arg1: hello
    arg2: world
```

### Additional resources

See the event filter plugins in [ansible.eda collection](#) for more examples of how to author them.

## CHAPTER 12. EVENT-DRIVEN ANSIBLE LOGGING STRATEGY

Event-Driven Ansible offers an audit logging solution over its resources. Each supported create, read, update and delete (CRUD) operation is logged against rulebook activations, event streams, decision environments, projects, and activations. Some of these resources support further operations, such as sync, enable, disable, restart, start, and stop; for these operations, logging is supported as well. These logs are only retained for the lifecycle of its associated container. See the following sample logs for each supported logging operation.

### 12.1. LOGGING SAMPLES

When the following APIs are called for each operation, you see the following audit logs:

#### Rulebook activation

1. Create
  1. 2024-08-15 14:13:20,384 aap\_eda.api.views.activation INFO Action: Create / ResourceType: RulebookActivation / ResourceName: quick\_start\_project / ResourceID: 53 / Organization: Default
2. Read
  1. 2024-08-15 14:21:26,844 aap\_eda.api.views.activation INFO Action: Read / ResourceType: RulebookActivation / ResourceName: quick\_start\_activation / ResourceID: 1 / Organization: Default
3. Disable
  1. 2024-08-15 14:23:57,798 aap\_eda.api.views.activation INFO Action: Disable / ResourceType: RulebookActivation / ResourceName: quick\_start\_activation / ResourceID: 1 / Organization: Default
4. Enable
  1. 2024-08-15 14:24:16,472 aap\_eda.api.views.activation INFO Action: Enable / ResourceType: RulebookActivation / ResourceName: quick\_start\_activation / ResourceID: 1 / Organization: Default
5. Delete
  1. 2024-08-15 14:24:53,847 aap\_eda.api.views.activation INFO Action: Delete / ResourceType: RulebookActivation / ResourceName: quick\_start\_activation / ResourceID: 1 / Organization: Default
6. Restart
  - 2024-08-15 14:24:34,169 aap\_eda.api.views.activation INFO Action: Restart / ResourceType: RulebookActivation / ResourceName: quick\_start\_activation / ResourceID: 1 / Organization: Default

#### EventStream Logs

1. Create
  1. 2024-08-15 13:46:26,903 aap\_eda.api.views.webhook INFO Action: Create / ResourceType: EventStream / ResourceName: ZackTest / ResourceID: 1 / Organization: Default
2. Update
  1. 2024-08-15 13:56:17,440 aap\_eda.api.views.webhook INFO Action: Update / ResourceType: EventStream / ResourceName: ZackTest / ResourceID: 1 / Organization: Default
3. Read
  1. 2024-08-15 13:56:56,271 aap\_eda.api.views.webhook INFO Action: Read / ResourceType: EventStream / ResourceName: ZackTest / ResourceID: 1 / Organization: Default
4. List
  1. 2024-08-15 13:56:17,492 aap\_eda.api.views.webhook INFO Action: List / ResourceType: EventStream / ResourceName: \* / ResourceID: \* / Organization: \*
5. Delete
  1. 2024-08-15 13:57:13,124 aap\_eda.api.views.webhook INFO Action: Delete / ResourceType: EventStream / ResourceName: ZackTest / ResourceID: None / Organization: Default

#### Decision Environment

## 1. Create

1. 2024-08-15 14:10:53,311 aap\_eda.api.views.decision\_environment INFO Action: Create / Resource Type: DecisionEnvironment / ResourceName: quick\_start\_de / ResourceID: 86 / Organization: Default

## 2. Read

1. 2024-08-15 14:10:53,349 aap\_eda.api.views.decision\_environment INFO Action: Read / Resource Type: DecisionEnvironment / ResourceName: quick\_start\_de / ResourceID: 86 / Organization: Default

## 3. Update

2024-08-15 14:11:20,970 aap\_eda.api.views.decision\_environment INFO Action: Update / Resource Type: DecisionEnvironment / ResourceName: quick\_start\_de / ResourceID: 86 / Organization: Default

## 4. Delete

2024-08-15 14:11:42,369 aap\_eda.api.views.decision\_environment INFO Action: Delete / Resource Type: DecisionEnvironment / ResourceName: quick\_start\_de / ResourceID: None / Organization: Default

**Project**

## 1. Create

1. 2024-08-15 14:05:26,874 aap\_eda.api.views.project INFO Action: Create / Resource Type: Project / ResourceName: quick\_start\_project / ResourceID: 86 / Organization: Default

## 2. Read

1. 2024-08-15 14:05:26,913 aap\_eda.api.views.project INFO Action: Read / Resource Type: Project / ResourceName: quick\_start\_project / ResourceID: 86 / Organization: Default

## 3. Update

1. 2024-08-15 14:06:08,255 aap\_eda.api.views.project INFO Action: Update / Resource Type: Project / ResourceName: quick\_start\_project / ResourceID: 86 / Organization: Default

## 4. Sync

1. 2024-08-15 14:06:30,580 aap\_eda.api.views.project INFO Action: Sync / Resource Type: Project / ResourceName: quick\_start\_project / ResourceID: 86 / Organization: Default

## 5. Delete

1. 2024-08-15 14:06:49,481 aap\_eda.api.views.project INFO Action: Delete / Resource Type: Project / ResourceName: quick\_start\_project / ResourceID: 86 / Organization: Default

**Activation Start/Stop**

## 1. Start

1. 2024-08-15 14:21:29,076 aap\_eda.services.activation.activation\_manager INFO Requested to start activation 1, starting.

2024-08-15 14:21:29,093 aap\_eda.services.activation.activation\_manager INFO Creating a new activation instance for activation: 1

2024-08-15 14:21:29,104 aap\_eda.services.activation.activation\_manager INFO Starting container for activation instance: 1

## 2. Stop

1. eda-activation-worker-1 | 2024-08-15 14:40:52,547 aap\_eda.services.activation.activation\_manager INFO Stop operation requested for activation id: 2 Stopping activation.

eda-activation-worker-1 | 2024-08-15 14:40:52,550 aap\_eda.services.activation.activation\_manager INFO Activation 2 is already stopped.

eda-activation-worker-1 | 2024-08-15 14:40:52,550 aap\_eda.services.activation.activation\_manager INFO Activation manager activation id: 2

Activation restart scheduled for 1 second.

eda-activation-worker-1 | 2024-08-15 14:40:52,562 rq.worker INFO activation: Job OK  
(activation-2)