



Red Hat Ansible Automation Platform 2.4

Event-Driven Ansible Controller ユーザーガイド

Event-Driven Ansible Controller を設定して使用し、自動化を強化および拡張する方法

Red Hat Ansible Automation Platform 2.4 Event-Driven Ansible Controller ユーザーガイド

Event-Driven Ansible Controller を設定して使用し、自動化を強化および拡張する方法

法律上の通知

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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.

概要

このガイドは、Event-Driven Ansible Controller を設定して、新しいプロジェクト、決定環境、Ansible Automation Platform Controller に対して認証するためのトークン、およびルールブックアクティベーションをセットアップするのに役立ちます。

目次

はじめに	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
第1章 EVENT-DRIVEN ANSIBLE CONTROLLER の概要	5
第2章 EVENT-DRIVEN ANSIBLE CONTROLLER の認証情報のセットアップ	6
2.1. 認証情報のセットアップ	6
2.2. 認証情報のリストビュー	7
2.3. 認証情報の編集	7
2.4. 認証情報の削除	7
第3章 プロジェクト	8
3.1. 新しいプロジェクトの設定	8
3.2. PROJECTS リストビュー	9
3.3. プロジェクトの編集	9
3.4. プロジェクトの削除	9
第4章 決定環境	11
4.1. 新しい決定環境の設定	11
4.2. ANSIBLE AUTOMATION PLATFORM 内での EVENT-DRIVEN ANSIBLE のカスタム決定環境のビルド	12
第5章 AUTOMATION CONTROLLER トークンの設定	14
5.1. AUTOMATION CONTROLLER への認証のためのトークンのセットアップ	14
第6章 ルールブックアクティベーション	15
6.1. ルールブックアクティベーションの設定	15
6.2. ルールブックアクティベーションのリストビュー	16
6.3. ルールブックアクティベーションの有効化および無効化	18
6.4. ルールブックアクティベーションの再起動	18
6.5. ルールブックアクティベーションの削除	18
6.6. WEBHOOK ルールブックのアクティブ化	19
6.7. KUBERNETES を使用したテスト	20
第7章 ルール監査	21
7.1. ルール監査の詳細の表示	21
7.2. ルール監査イベントの表示	21
7.3. ルール監査アクションの表示	22
第8章 EVENT-DRIVEN ANSIBLE CONTROLLER のパフォーマンスチューニング	23
8.1. ワークロードの特徴付け	23
8.2. EVENT-DRIVEN ANSIBLE CONTROLLER のシステムレベルの監視	25
8.3. EVENT-DRIVEN ANSIBLE CONTROLLER のパフォーマンストラブルシューティング	25

はじめに

Event-Driven Ansible Controller は、一貫性と復元力を実現しながら IT の速度と俊敏性を向上させることで、自動化を強化および拡張する新しい方法です。Red Hat によって開発されたこの機能は、シンプルさと柔軟性を考慮して設計されています。

RED HAT ドキュメントへのフィードバック (英語のみ)

このドキュメントの改善に関するご意見がある場合や、エラーを発見した場合は、<https://access.redhat.com> から Technical Support チームに連絡してください。

第1章 EVENT-DRIVEN ANSIBLE CONTROLLER の概要

Event-Driven Ansible は、他のソフトウェアベンダーのモニタリングツールなどのイベントソースと連携する、拡張性が高く柔軟な自動化機能です。このようなツールは、IT ソリューションを監視してイベントを特定し、ルールブックに記述された変更や対応を自動的に実装して、そのイベントを処理します。

ユーザー設定は次の手順で行います。

- [認証情報のセットアップ](#)
- [新しいプロジェクトの設定](#)
- [新しい決定環境の設定](#)
- [Ansible Automation Platform Controller に対して認証するためのトークンの設定](#)
- [ルールブックアクティベーションの設定](#)



注記

Event-Driven Ansible Controller の API ドキュメントは、<https://<eda-server-host>/api/eda/v1/docs> で入手できます。

第2章 EVENT-DRIVEN ANSIBLE CONTROLLER の認証情報のセットアップ

認証情報は、ルールブックの起動時に、Event-Driven Ansible が認証するために使用されます。

2.1. 認証情報のセットアップ

プライベートリポジトリ (GitHub または GitLab) あるいはプライベートコンテナレジストリーで使用する認証情報を作成します。



重要

GitHub または GitLab リポジトリを使用している場合は、**basic auth** メソッドを使用します。両方の SCM サーバーが正式にサポートされています。**basic auth** をサポートする SCM プロバイダーを使用できます。

手順

1. Event-Driven Ansible Controller ダッシュボードにログインします。
2. ナビゲーションパネルから **Resources** → **Credentials** を選択します。
3. **Create credential** をクリックします。
4. 以下を入力します。

Name

名前を入力します。

Description

このフィールドは任意です。

認証情報タイプ

利用できるオプションは、GitHub パーソナルアクセストークン、GitLab パーソナルアクセストークン、またはコンテナレジストリーです。

Username

ユーザー名を挿入します。

Token

宛先への認証を可能にするトークンを挿入します。



注記

コンテナレジストリーを使用している場合、トークンフィールドは、レジストリープロバイダーに応じてトークンまたはパスワードにすることができます。Ansible Automation Platform ハブレジストリーを使用している場合は、そのパスワードを token フィールドに挿入します。

5. **Create credential** をクリックします。

認証情報を保存すると、認証情報の詳細ページが表示されます。詳細ページまたは **Credentials** リストビューから、認証情報を編集または削除できます。

2.2. 認証情報のリストビュー

Credentials ページで、認証情報の **Type** とともに作成した、作成済みの認証情報のリストを表示できます。

メニューバーの **Name** フィールドで、認証情報を検索できます。

メニューバーには、以下のオプションもあります。

- **Manage columns** をクリックして、リストビューに表示する列を選択します。
- アイコンをクリックして、**List view** または **Card view** のいずれかを選択します。

2.3. 認証情報の編集

手順

1. 以下のいずれかの方法を使用して、認証情報を編集します。
 - **Credentials** リストビューで、必要な認証情報の横にある **Edit credential** アイコンをクリックします。
 - **Credentials** リストビューで認証情報の名前を選択し、**Edit credential** をクリックします。
2. 適切な詳細を編集し、**Save credential** をクリックします。

2.4. 認証情報の削除

手順

1. 以下のいずれかの方法を使用して、認証情報を削除します。
 - **Credentials** リストビューで、必要な認証情報の横にある **More Actions** アイコン **⋮** をクリックし、**Delete credential** をクリックします。
 - **Credentials** リストビューで認証情報の名前を選択し、**Edit credential** の横にある **More Actions** アイコン **⋮** をクリックして、**Delete credential** をクリックします。
2. ポップアップウィンドウで、**Yes, I confirm to delete this credential** を選択します。
3. **Delete credential** をクリックします。

各認証情報の横にあるチェックボックスを選択し、メニューバーの **More Actions** アイコン **⋮** をクリックしてから、**Delete selected credentials** をクリックして、一度に複数の認証情報を削除できます。

第3章 プロジェクト

プロジェクトはルールブックの論理的なコレクションです。これは git リポジトリである必要があり、http プロトコルのみがサポートされます。プロジェクトのルールブックは、Ansible コレクション内の Event-Driven Ansible コンテンツ用に定義されたパス (プロジェクトのルートにある `/extensions/eda/rulebooks`) に配置する必要があります。

3.1. 新しいプロジェクトの設定

前提条件

- Event-Driven Ansible Controller ダッシュボードに Content Consumer としてログインしている。
- 必要に応じて、認証情報が設定されている。詳細は、[認証情報のセットアップ](#) セクションを参照してください。
- Automation Controller が使用する、リポジトリに含まれている Playbook と統合されたルールブックを含む既存のリポジトリがある。

手順

1. Event-Driven Ansible Controller ダッシュボードにログインします。
2. ナビゲーションパネルから、**Projects** → **Create project** を選択します。
3. 以下を入力します。

Name

プロジェクト名を入力します。

Description

このフィールドは任意です。

SCM type

Git のみ使用できます。

SCM URL

GitHub や GitLab などのリポジトリの HTTP[S] プロトコルアドレス。



注記

プロジェクトの作成後に SCM URL を編集することはできません。

Credential

このフィールドは任意です。これは、SCM URL を利用するために必要なトークンです。

4. **Create Project** を選択します。

プロジェクトが作成され、**Projects** 画面で管理できるようになります。

新しいプロジェクトを保存すると、プロジェクトの詳細ページが表示されます。詳細ページまたは **Projects** リストビューから、プロジェクトを編集または削除できます。

3.2. PROJECTS リストビュー

Projects ページでは、作成したプロジェクトを **ステータス** および **Gitハッシュ** とともに確認できます。



注記

ソース管理でルールブックが変更された場合は、**Projects** リストビューでプロジェクトの横にある同期アイコンを選択することで、プロジェクトを再同期できます。**Gitハッシュ** の更新は、そのリポジトリ上の最新のコミットを表します。更新されたプロジェクトを使用する場合は、アクティベーションを再開または再作成する必要があります。

3.3. プロジェクトの編集

手順

1. **Projects** リストビューで、目的のプロジェクトの横にある **More Actions** アイコン **⋮** を選択します。
2. **Edit project** を選択します。
3. 必要な変更を入力し、**Save project** を選択します。

The screenshot shows the 'Edit Project 1' interface in the Red Hat Ansible Automation Platform. The form contains the following fields:

Name *	Description	SCM type
Project 1	test	Git

Below the main form is a 'Credential' dropdown menu with the text 'Select credential'. At the bottom of the form are two buttons: 'Save project' and 'Cancel'.

3.4. プロジェクトの削除

手順

1. **Projects** リストビューで、目的のプロジェクトの横にある **More Actions** アイコン **⋮** を選択します。
2. **Delete project** を選択します。
3. ポップアップウィンドウで、**Yes, I confirm to delete this project** を選択します。

4. **Delete project** を選択します。

第4章 決定環境

決定環境は、Ansible ルールブックを実行するためのコンテナイメージです。決定環境は、自動化の依存関係を伝達するための共通言語を作成し、自動化環境をビルドおよび配布するための標準的な方法を提供します。デフォルトの決定環境は [Ansible-Rulebook](#) にあります。

独自の決定環境を作成するには、[Ansible Automation Platform 内での Event-Driven Ansible のカスタム決定環境のビルド](#) を参照してください。

4.1. 新しい決定環境の設定

次の手順では、Event-Driven Ansible Controller ダッシュボードに決定環境をインポートする方法を説明します。

前提条件

- Event-Driven Ansible Controller ダッシュボードに Content Consumer としてログインしている。
- 必要に応じて、認証情報が設定されている。詳細は、[認証情報のセットアップ](#) セクションを参照してください。
- 決定環境イメージをイメージリポジトリにプッシュしたか、[registry.redhat.io](#) で提供されている **de-supported** イメージを使用することを選択した。

手順

1. Event-Driven Ansible Controller ダッシュボードに移動します。
2. ナビゲーションパネルから **Decision Environments** を選択します。
3. 以下を入力します。

Name

名前を入力します。

Description

このフィールドは任意です。

Image

これは、コンテナレジストリー、イメージ名、およびバージョンタグを含む完全なイメージの場所です。

Credential

このフィールドは任意です。これは、決定環境イメージを利用するために必要なトークンです。

4. **Create decision environment** を選択します。

これで決定環境が作成され、**Decision Environments** 画面で管理できるようになります。

新しい決定環境を保存すると、決定環境の詳細ページが表示されます。詳細ページまたは **Decision Environment** リストビューから、決定環境を編集または削除できます。

4.2. ANSIBLE AUTOMATION PLATFORM 内での EVENT-DRIVEN ANSIBLE のカスタム決定環境のビルド

デフォルトの意思決定環境にはないカスタム管理やサードパーティーのイベントソースプラグインを提供するためにカスタム意思決定環境が必要な場合は、次の手順に従います。

前提条件

- Ansible Automation Platform 2.4 以降
- Event-Driven Ansible
- Ansible Builder 3.0 以降

手順

- **de-supported** 決定環境を追加します。このイメージは、**de-minimal** と呼ばれる Red Hat が提供するベースイメージからビルドされます。



注記

Red Hat は、Ansible Builder でベースイメージとして **de-minimal** を使用して、カスタム決定環境をビルドすることを推奨しています。

以下は、ansible.eda コレクションでカスタム決定環境をビルドするために、ベースイメージとして **de-minimal** を使用する Ansible Builder 定義ファイルの例です。

```
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
```

さらに、他の Python パッケージまたは RPM が必要な場合は、1つの定義ファイルに以下を追加します。

```
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
```

第5章 AUTOMATION CONTROLLER トークンの設定

Automation Controller には、Event-Driven Ansible ルールブックと連携するように設計された Playbook を含むリポジトリをベースとしたプロジェクトが含まれている必要があります。Automation Controller には、そのプロジェクト内の Playbook に基づいて設定された対応するジョブテンプレートも必要です。

5.1. AUTOMATION CONTROLLER への認証のためのトークンのセットアップ

前提条件

- Event-Driven Ansible Controller ダッシュボードに Content Consumer としてログインしている。
- ユーザーを作成している。
- Event-Driven Ansible Controller ダッシュボードにログインできるか、組織内のユーザーとして追加されている。

手順

1. Event-Driven Ansible Controller ダッシュボードに移動します。
2. 上部のナビゲーションパネルからプロフィールを選択します。
3. **User details** に移動します。
4. **Controller Tokens** → **Create controller token** を選択します。
5. 以下を入力します。

Name

名前を入力します。

Description

このフィールドは任意です。

Token

Automation Controller でトークンを作成します。トークンの作成の詳細は、Automation Controller ユーザーガイドの [ユーザー - トークン](#) セクションを参照してください。



注記

トークンは書き込みスコープ内にある必要があり、割り当てられたユーザーの場合は、ジョブおよびワークフローテンプレートを実行する権限を持っている必要があります。

6. **Create controller token** を選択します。

新しいトークンを保存すると、**Controller Tokens** タブが表示され、そこでトークンを削除できます。

第6章 ルールブックアクティベーション

ルールブックアクティベーションは、特定のルールブックを実行する決定環境によって定義されるバックグラウンドで実行されるプロセスです。

6.1. ルールブックアクティベーションの設定

前提条件

- Event-Driven Ansible Controller ダッシュボードに Content Consumer としてログインしている。
- プロジェクトを設定している。
- 決定環境を設定している。
- Automation Controller のトークンを設定している。

手順

1. Event-Driven Ansible Controller ダッシュボードに移動します。
2. ナビゲーションパネルから **Rulebook Activations** を選択します。
3. 以下を入力します。

Name

名前を入力します。

Description

このフィールドは任意です。

Project

プロジェクトはルールブックの論理的なコレクションです。

Rulebook

選択したプロジェクトに応じてルールブックが表示されます。

Decision environment

決定環境は、Ansible ルールブックを実行するためのコンテナイメージです。



注記

Event-Driven Ansible コントローラーでは、Decision environment のプルポリシーをカスタマイズできません。デフォルトでは、**always** ポリシーの動作に従います。アクティベーションが開始されるたびに、システムはイメージの最新バージョンを取得しようとします。

Restart policy

これは、ルールブックを再起動する条件を決定するためのポリシーです。

- Policies:
 - i. Always: ルールブック終了時に再起動します。

- ii. Never: 終了時にルールブックを再起動しません。
- iii. On failure: 失敗した場合にのみ再起動します。

Rulebook activation enabled?

これを選択すると、ルールブックアクティベーションが自動的に有効になります。

Variables

ルールブックの変数は JSON/YAML 形式です。内容は、ansible-rulebook コマンドの `--vars` フラグによって渡されるファイルと同等です。

4. Create rulebook activation をクリックします。

ルールブックアクティベーションが作成され、**Rulebook Activations** 画面で管理できるようになります。

新しいルールブックアクティベーションを保存すると、ルールブックアクティベーションの詳細ページが表示されます。詳細ページまたは **Rulebook Activations** リストビューから、ルールブックアクティベーションを編集または削除できます。



注記

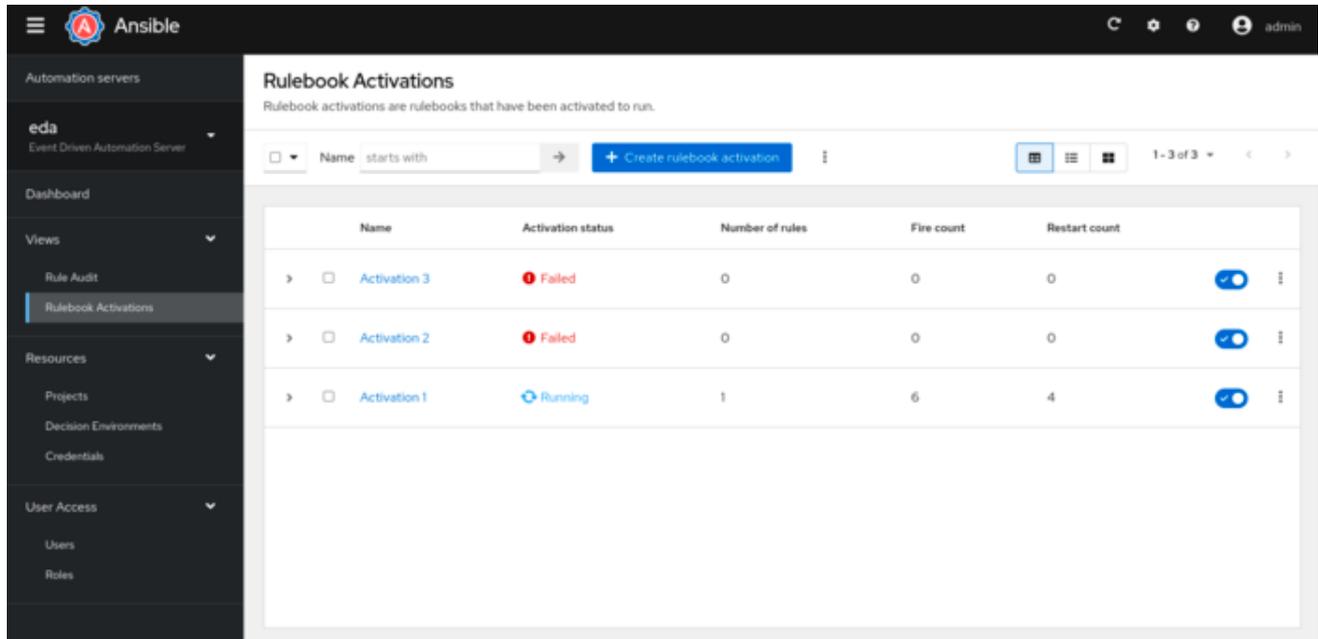
ソースプラグインのシャットダウンが原因で、ルールブックが正常に終了するまでに一定の時間かかることがあります。ルールブックアクティベーションがシャットダウンすると、待機しているすべてのタスクがキャンセルされ、info レベルのメッセージがアクティベーションログに送信されます。詳細は、[Rulebooks](#) を参照してください。

6.2. ルールブックアクティベーションのリストビュー

Rulebook Activations ページでは、作成したルールブックアクティベーションを、**アクティベーションステータス**、ルールブックに **関連付けられたルールの数**、**起動数**、および **再起動数** とともに確認できます。

Activation Status が **Running** の場合、ルールブックのアクティベーションがバックグラウンドで実行されており、ルールブックで宣言されたルールに従って必要なアクションが実行されていることを意味します。

Rulebook Activations リストビューからアクティベーションを選択すると、詳細を確認できます。



実行されたすべてのアクティベーションについて、**Details** タブと **History** タブを表示して、実行結果に関する詳細情報を取得できます。

6.2.1. アクティベーションの出力の表示

アクティベーションの出力は **History** タブで確認できます。

手順

1. **History** タブを選択して、すべてのアクティベーションインスタンスのリストにアクセスします。1つのアクティベーションインスタンスは、1回のアクティベーションの実行を表します。
2. 対象のアクティベーションインスタンスを選択すると、その特定の実行によって生成された **Output** が表示されます。

アクションをトリガーした受信イベントを表示するには、Event-Driven Ansible コントローラーダッシュボードの [Rule Audit](#) セクションを使用できます。

6.3. ルールブックアクティベーションの有効化および無効化

1. 行レベルのスイッチを選択して、選択したルールブックを有効または無効にします。
2. ポップアップウィンドウで、**Yes, I confirm that I want to enable/disable these X rulebook activations** 選択します。
3. **Enable/Disable rulebook activation** を選択します。

6.4. ルールブックアクティベーションの再起動



注記

ルールブックアクティベーションが現在有効であり、作成時に再起動ポリシーが **Always** に設定されていた場合にのみ、ルールブックアクティベーションを再起動できます。

1. **Rulebook Activation enabled/disabled** トグルの横にある **More Actions** アイコン **⋮** を選択します。
2. **Restart rulebook activation** を選択します。
3. ポップアップウィンドウで、**Yes, I confirm that I want to restart these X rulebook activations** を選択します。
4. **Restart rulebook activations** を選択します。

6.5. ルールブックアクティベーションの削除

1. **Rulebook Activation enabled/disabled** トグルの横にある **More Actions** アイコン  を選択します。
2. **Delete rulebook activation** を選択します。
3. ポップアップウィンドウで、**Yes, I confirm that I want to delete these X rulebook activations** を選択します。
4. **Delete rulebook activations** を選択します。

6.6. WEBHOOK ルールブックのアクティブ化

Openshift 環境では、ルールブックアクティベーションの Kubernetes サービスを公開するルートを作成することで、Webhook が特定のポートを介して activation-job-pod に到達できるようにすることが可能です。

前提条件

- Event-Driven Ansible Controller ダッシュボードでルールブックアクティベーションを作成している。

注記

特定の Webhook を含むルールブックの例を以下に示します。

```
- 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
```

手順

1. サービスを公開するためのルートを (OpenShift Container Platform 上に) 作成します。以下は、決定環境 Pod のポート 5000 で POST を要求する ansible-rulebook ソースのルートの例です。

```
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. ルートを作成したら、**ルート URL へのポスト** を使用してテストします。

```

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

```



注記

ポートはルート (targetPort) で指定されているため、必要ありません。

6.7. KUBERNETES を使用したテスト

Kubernetes を使用すると、Ingress の作成やポートの公開を行うことができます。ただし、これは実稼働環境用のものではありません。

手順

1. 次のコマンドを実行して、クラスター上の特定のサービスのポートを公開します。

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

2. **localhost:5000** に対して HTTP リクエストを実行して、ルールブックをトリガーします。

```

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

```

第7章 ルール監査

ルール監査では、ある時点においてアクティブ化されたすべてのルールによってトリガーされたルールを監査できます。

Rule Audit リストビューには、ルールブック内の条件と一致してアクションをトリガーしたイベントを受信した各タイミングのリストが表示されます。リストにはルールブック内のルールが表示され、各見出しに実行されたルール名が表示されます。

7.1. ルール監査の詳細の表示

Rule Audit リストビューから、特定のアクションをトリガーしたイベントを確認できます。

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

手順

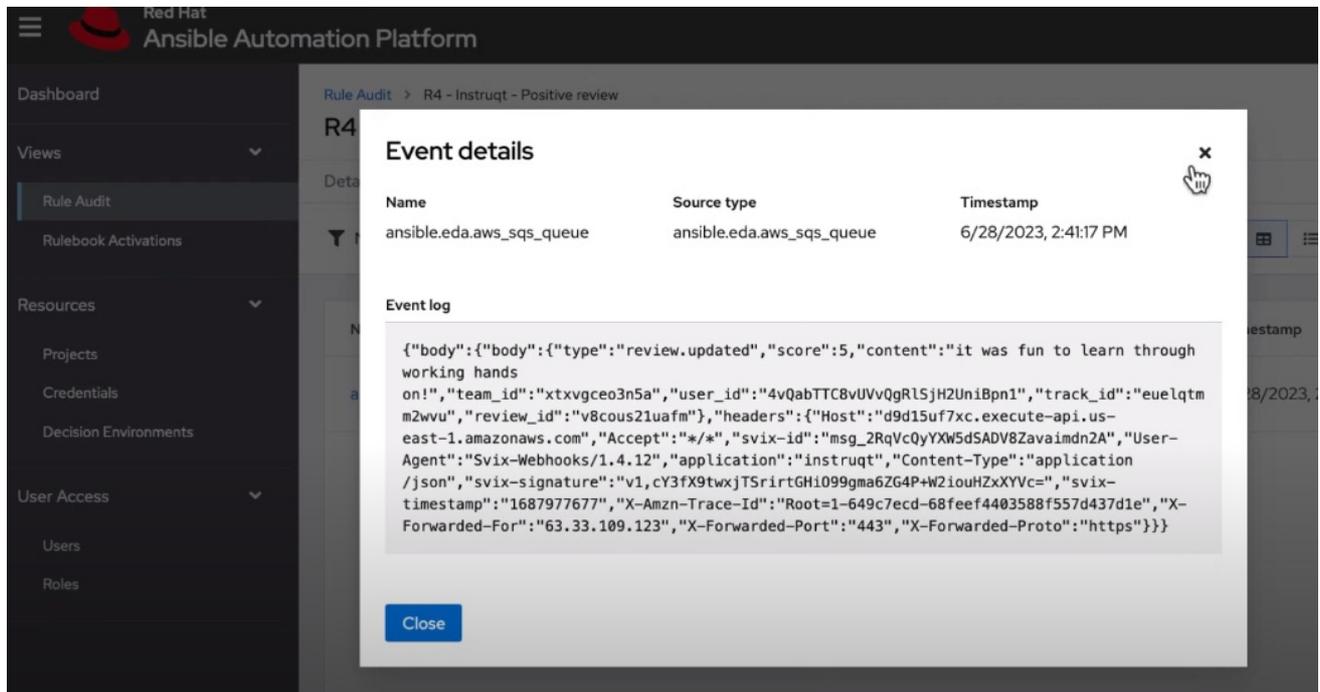
1. ナビゲーションパネルから **Rule Audit** を選択します。
2. 目的のルールを選択すると、**Details** タブが表示されます。

ここから、作成日時、最後に起動した日時、および対応するルールブックアクティベーションを確認できます。

7.2. ルール監査イベントの表示

手順

1. ナビゲーションパネルから **Rule Audit** を選択します。
2. 目的のルールを選択すると、**Details** タブが表示されます。アクションをトリガーしたすべてのイベントを表示するには、**Events** タブを選択します。選択すると、アクションをトリガーしたイベントが表示されます。
3. イベントを選択すると、**Event log** が **Source type** と **Timestamp** とともに表示されます。



7.3. ルール監査アクションの表示

手順

1. ナビゲーションパネルから **Rule Audit** を選択します。
2. 目的のルールを選択すると、**Actions** タブが表示されます。

ここから、実行されたアクションを確認できます。一部のアクションは Automation Controller にリンクされています。そこで出力を確認できます。

第8章 EVENT-DRIVEN ANSIBLE CONTROLLER のパフォーマンスチューニング

Event-Driven Ansible は、非常にスケーラブルで柔軟な自動化機能です。Event-Driven Ansible Controller は、Event-Driven Ansible による自動化を実行するインターフェイスを提供します。Event-Driven Ansible Controller をチューニングし、次の方法でパフォーマンスとスケーラビリティを最適化します。

- ワークロードの特徴付け
- システムレベルの監視
- パフォーマンスのトラブルシューティング

8.1. ワークロードの特徴付け

Event-Driven Ansible Controller では、ワークロードに、受信したルールブックアクティベーションとイベントの数が含まれます。Event-Driven Ansible Controller ワークロードを特徴付けるために、次の要素を考慮してください。

1. ルールブックの同時アクティベーションの数
2. Event-Driven Ansible Controller が受信したイベントの数

8.1.1. ルールブックの同時アクティベーションの数の変更

デフォルトでは、Event-Driven Ansible Controller は 12 のルールブックアクティベーションを同時に実行できます。12 を超えるルールブックアクティベーションが作成されると、ルールブックアクティベーションワーカーが利用可能になるまで、後続のルールブックアクティベーションが保留中として待機することが予想されます。この場合、Event-Driven Ansible Controller インスタンスに空きメモリーと CPU が十分にある場合でも、ルールブックアクティベーションのステータスは "pending" と表示されます。この動作を変更するには、実行中のルールブックアクティベーションのデフォルトの最大数を変更する必要があります。

注記: `EDA_MAX_RUNNING_ACTIVATIONS` の値は、インスタンスサイズが変更されても変更されないため、手動で調整する必要があります。

8.1.1.1. Event-Driven Ansible Controller のインストール時に、ルールブックの同時アクティベーションの数を変更する

デフォルトでは、Event-Driven Ansible Controller は 12 つのアクティベーションを同時に実行できません。以下の手順を使用して、インストール時にこのデフォルト値を変更できます。

手順

仮想マシンインストーラーに変数を指定します。

1. セットアップインベントリーファイルに移動します。
2. `[all:vars]` セクションに `automationedacontroller_max_running_activations` を追加します。たとえば、`automationedacontroller_max_running_activations=16` などです。
3. セットアップを実行します。

8.1.1.2. Event-Driven Ansible Controller のインストール後に、ルールブックの同時アクティベーションの数を変更する

デフォルトでは、Event-Driven Ansible Controller は 12 つのアクティベーションを同時に実行できます。以下の手順を使用して、インストール後にこのデフォルト値を変更できます。

手順

1. `/etc/ansible-automation-platform/eda` の環境ファイルに移動します。
2. 必要な実行中のアクティベーションの最大数を選択します。たとえば、`EDA_MAX_RUNNING_ACTIVATIONS = 16` などです。
3. `systemctl restart automation-eda-controller.target` コマンドを使用して、EDA サービスを再起動します。

関連情報

ルールブックアクティベーションの詳細は、[ルールブックアクティベーション](#) を参照してください。

8.1.2. ルールブックアクティベーションごとのデフォルトのメモリー制限の変更

メモリー使用量は、Event-Driven Ansible Controller が処理する必要のあるイベントの数に基づいています。各ルールブックアクティベーションコンテナには、200MB のメモリー制限があります。たとえば、CPU が 4 個、RAM が 16 GB の場合、割り当てられたメモリー制限が 200 MB のルールブックアクティベーションコンテナ 1 つでは、1 分あたり 150,000 件を超えるイベントを処理できません。並行して実行されるルールブックアクティベーションの数が多い場合は、各ルールブックアクティベーションが処理できるイベントの最大数は少なくなります。非常に高いレートでの受信イベントが多すぎる場合、コンテナはイベントを処理しようとしてメモリー不足になる可能性があります。これによりコンテナが強制終了され、ルールブックアクティベーションはステータスコード 137 で失敗します。

この障害に対処するには、次のいずれかの手順を使用して、ルールブックアクティベーションに割り当てられるメモリーの量を増やし、多数のイベントを高いレートで処理できるようにします。

- インストール中に、ルールブックアクティベーションごとにデフォルトのメモリー制限を変更する
- インストール後に、ルールブックアクティベーションごとにデフォルトのメモリー制限を変更する

8.1.2.1. インストール中に、ルールブックアクティベーションごとにデフォルトのメモリー制限を変更する

デフォルトでは、ルールブックアクティベーションコンテナごとに 200MB のメモリー制限があります。以下の手順を使用して、インストール時にこのデフォルト値を変更できます。

手順

1. セットアップインベントリーファイルに移動します。
2. `[all:vars]` セクションに `automationedacontroller_podman_mem_limit` を追加します。たとえば、`automationedacontroller_podman_mem_limit='400m'` などです。
3. セットアップを実行します。

8.1.2.2. インストール後に、ルールブックアクティベーションごとにデフォルトのメモリ制限を変更する

デフォルトでは、ルールブックアクティベーションコンテナごとに 200MB のメモリ制限があります。以下の手順を使用して、インストール後にこのデフォルト値を変更できます。

手順

1. `/etc/ansible-automation-platform/eda` の環境ファイルに移動します。
2. デフォルトのコンテナのメモリ制限を変更します。たとえば、`EDA_PODMAN_MEM_LIMIT = '300m'` などです。
3. `systemctl restart automation-eda-controller.target` を使用して、Event-Driven Ansible Controller サービスを再起動します。

8.2. EVENT-DRIVEN ANSIBLE CONTROLLER のシステムレベルの監視

ワークロードを特徴付けし、並行して実行しているルールブックのアクティベーションの数と、特定の時点で受信しているイベントの数を特定した後、システムレベルで Event-Driven Ansible Controller ホストを監視することを検討する必要があります。システムレベルのモニタリングを使用して、Event-Driven Ansible のパフォーマンスに関する情報を時間の経過とともに確認することで、問題を診断したり、将来的な容量の増加を検討したりする場合に役立ちます。

システムレベルの監視では、次の情報が含まれます。

- ディスク I/O
- RAM 使用率
- CPU の使用率
- ネットワークトラフィック

CPU、RAM、またはディスクの使用率が高いと、Event-Driven Ansible Controller の全体的なパフォーマンスに影響を与える可能性があります。たとえば、これらのシステムレベルリソースのうち、使用率が高い場合は、Event-Driven Ansible Controller が実行しているルールブックアクティベーションが多すぎるか、または個々のルールブックアクティベーションの一部で大量のリソースを使用していることを示しています。このような場合は、ワークロードをサポートするためにシステムレベルのリソースを増やす必要があります。

8.3. EVENT-DRIVEN ANSIBLE CONTROLLER のパフォーマンストラブルシューティング

Event-Driven Ansible Controller のデフォルトパラメーターに基づいて、ワークロードを完了するのに課題となるシナリオが発生する可能性があります。次のセクションでは、これらのシナリオおよびトラブルシューティングのガイダンスについて説明します。

- アクティベーションのステータスは "running" と表示されますが、イベントを処理していません。
 - ルールブックアクティベーションで正しいイベントソースを使用していることを確認します。ルールブック以外のソースから発生するイベントを想定している場合は、Event-Driven Ansible Controller はそのイベントを処理しません。

- アクティベーションステータスは "running" と表示され、Event-Driven Ansible Controller もイベントを受信していますが、アクションが実行されません。
 - ルールブックアクティベーションでイベントの一致とアクションの実行に対して、正しい条件を設定していることを確認します。
- アクティベーションは無限ループで再起動し続けます。
 - デフォルトでは、ルールブックアクティベーションのリセットポリシーは "on failure" に設定されています。以下の手順を使用して、Restart Policy を変更します。
 1. Rulebook Activation 画面に移動します。
 2. Restart Policy ドロップダウンメニューをクリックします。
 3. 適切な値 ("On Failure"、"Always"、"Never") を選択します。