



# Red Hat Ansible Automation Platform 2.4

## 이벤트 기반 **Ansible** 컨트롤러 사용자 가이드

이벤트 기반 Ansible 컨트롤러를 구성하고 사용하여 자동화를 개선하고 확장하는 방법  
알아보기



# Red Hat Ansible Automation Platform 2.4 이벤트 기반 Ansible 컨트롤러 사용자 가이드

---

이벤트 기반 Ansible 컨트롤러를 구성하고 사용하여 자동화를 개선하고 확장하는 방법 알아보기

## 법적 공지

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.

## 초록

인증 정보, 새 프로젝트, 의사 결정 환경, Ansible Automation Platform Controller에 인증할 토큰 및 롤북 활성화를 설정하기 위해 이벤트 기반 Ansible 컨트롤러를 구성하는 방법을 알아봅니다.

## 차례

머리말 .....	3
RED HAT 문서에 관한 피드백 제공 .....	4
1장. 이벤트 기반 ANSIBLE 컨트롤러 개요 .....	5
2장. 이벤트 기반 ANSIBLE 컨트롤러에 대한 인증 정보 설정 .....	6
2.1. 인증 정보 설정 .....	6
2.2. 인증 정보 목록 보기 .....	6
2.3. 인증 정보 편집 .....	7
2.4. 인증 정보 삭제 .....	7
3장. 프로젝트 .....	8
3.1. 새 프로젝트 만들기 .....	8
3.2. 프로젝트 목록 보기 .....	8
3.3. 프로젝트 편집 .....	9
3.4. 프로젝트 삭제 .....	9
4장. 의사 결정 환경 .....	10
4.1. 새 의사 결정 환경 설정 .....	10
4.2. ANSIBLE AUTOMATION PLATFORM 내에서 이벤트 기반 ANSIBLE을 위한 사용자 정의 의사 결정 환경 빌드 .....	10
5장. 자동화 컨트롤러 토큰 설정 .....	13
5.1. 자동화 컨트롤러에 인증할 토큰 설정 .....	13
6장. 룰북 활성화 .....	14
6.1. 룰북 활성화 설정 .....	14
6.2. 룰북 활성화 목록 보기 .....	15
6.3. 룰북 활성화 및 비활성화 .....	16
6.4. 룰북 활성화 다시 시작 .....	16
6.5. 룰북 활성화 삭제 .....	17
6.6. WEBHOOK 룰북 활성화 .....	17
6.7. KUBERNETES로 테스트 .....	18
7장. 규칙 감사 .....	19
7.1. 규칙 감사 세부 정보 보기 .....	19
7.2. 규칙 감사 이벤트 보기 .....	19
7.3. 규칙 감사 작업 보기 .....	20
8장. 이벤트 기반 ANSIBLE 컨트롤러의 성능 튜닝 .....	21
8.1. 워크로드 특성화 .....	21
8.2. 이벤트 기반 ANSIBLE 컨트롤러에 대한 시스템 수준 모니터링 .....	23
8.3. 이벤트 기반 ANSIBLE 컨트롤러의 성능 문제 해결 .....	23



## 머리말

이벤트 기반 Ansible 컨트롤러는 IT 속도 및 민첩성을 개선하고 일관성과 탄력성을 활성화하여 자동화를 개선 및 확장할 수 있는 새로운 방법입니다. Red Hat에서 개발한 이 기능은 단순성과 유연성을 위해 설계되었습니다.

## RED HAT 문서에 관한 피드백 제공

이 문서를 개선하기 위한 제안이 있거나 오류를 찾을 수 있는 경우 <https://access.redhat.com> 에서 기술 지원에 문의하여 요청을 열 수 있습니다.



## 1장. 이벤트 기반 ANSIBLE 컨트롤러 개요

이벤트 기반 Ansible은 확장성이 뛰어나고 유연한 자동화 기능으로 다른 소프트웨어 벤더의 모니터링 툴과 같은 이벤트 소스와 함께 작동합니다. 이러한 툴은 IT 솔루션을 모니터링하고 이벤트를 식별하고 해당 이벤트를 처리하기 위한 룰북에서 문서화된 변경 사항 또는 응답을 자동으로 구현합니다.

다음 절차는 사용자 구성을 형성합니다.

- 인증 정보 설정
- 새 프로젝트 만들기
- 새 의사 결정 환경 설정
- Ansible Automation Platform Controller에 인증할 토큰 설정
- 룰북 활성화 설정



### 참고

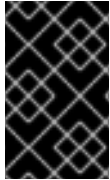
Event-Driven Ansible 컨트롤러에 대한 API 문서는 <https://<eda-server-host>/api/eda/v1/docs>에서 확인할 수 있습니다.

## 2장. 이벤트 기반 ANSIBLE 컨트롤러에 대한 인증 정보 설정

인증 정보는 룰북을 시작할 때 인증에 이벤트 기반 Ansible에서 사용됩니다.

### 2.1. 인증 정보 설정

개인 리포지토리(GitHub 또는 GitLab) 또는 프라이빗 컨테이너 레지스트리와 함께 사용할 인증 정보를 생성합니다.



#### 중요

GitHub 또는 GitLab 리포지토리를 사용하는 경우 기본 인증 방법을 사용합니다. 두 SCM 서버가 모두 공식적으로 지원됩니다. 기본 인증을 지원하는 모든 **SCM** 공급자를 사용할 수 있습니다.

#### 절차

1. 이벤트 기반 Ansible 컨트롤러 대시보드에 로그인합니다.
2. 탐색 패널에서 리소스 자격증명을 선택합니다.
3. 인증 정보 생성을 클릭합니다.
4. 다음을 삽입합니다.

#### 이름

이름을 삽입합니다.

#### 설명

이 필드는 선택 사항입니다.

#### 인증 정보 유형

사용 가능한 옵션은 GitHub 개인 액세스 토큰, GitLab 개인 액세스 토큰 또는 컨테이너 레지스트리입니다.

#### 사용자 이름

사용자 이름을 삽입합니다.

#### 토큰

대상에 인증할 수 있는 토큰을 삽입합니다.



#### 참고

컨테이너 레지스트리를 사용하는 경우 토큰 필드는 레지스트리 공급자에 따라 토큰 또는 암호일 수 있습니다. Ansible Automation Platform 허브 레지스트리를 사용하는 경우 토큰 필드에 암호를 삽입합니다.

5. 인증 정보 생성을 클릭합니다.

인증 정보를 저장하면 인증 정보 세부 정보 페이지가 표시됩니다. 인증 정보 목록 보기 또는 인증 정보 목록 뷰에서 편집하거나 삭제할 수 있습니다.

### 2.2. 인증 정보 목록 보기

인증 정보 페이지에서 인증 정보 유형 과 함께 생성한 생성된 인증 정보 목록을 볼 수 있습니다.

메뉴 모음에서 이름 필드에서 인증 정보를 검색할 수 있습니다.

메뉴 표시줄에도 다음과 같은 옵션이 있습니다.

- 열 관리를 클릭하여 목록 뷰에 표시되는 열을 선택합니다. Choose which columns are shown in the list view by clicking **Manage columns**.
- 아이콘을 클릭하여 목록 보기 또는 카드 보기 중에서 선택합니다.

## 2.3. 인증 정보 편집

### 절차

1. 다음 방법 중 하나를 사용하여 인증 정보를 편집합니다.
  - 인증 정보 목록 보기에서 원하는 인증 정보 옆에 있는 **자격 증명 편집** 아이콘을 클릭합니다.
  - 인증 정보 목록 뷰에서 인증 정보 이름을 선택하고 인증 정보 **편집**를 클릭합니다.
2. 적절한 세부 정보를 편집하고 **Save Credential** 을 클릭합니다.

## 2.4. 인증 정보 삭제

### 절차

1. 다음 방법 중 하나를 사용하여 인증 정보를 삭제합니다.
  - 인증 정보 목록 보기에서 원하는 인증 정보 옆에 있는 **More Actions** (추가 작업) 아이콘을 클릭하고 **Delete credential** 을 클릭합니다.
  - 인증 정보 목록 뷰에서 인증 정보의 이름을 선택하고 **Edit Credential** (자격 증명 편집) 옆에 있는 **More Actions** (추가 작업) 아이콘을 클릭하고 자격 증명 삭제를 클릭합니다.
2. 팝업 창에서 **Yes**를 선택하고 이 인증 정보를 삭제하고 싶습니다.
3. 인증 정보 삭제를 클릭합니다.

각 인증 정보 옆에 있는 확인란을 선택하고 메뉴 표시줄에서 **More Actions** 아이콘을 클릭한 다음 선택한 인증 정보 삭제를 클릭하여 한 번에 여러 인증 정보를 삭제할 수 있습니다.

## 3장. 프로젝트

프로젝트는 룰북의 논리 컬렉션입니다. git 리포지토리어야 하며 http 프로토콜만 지원됩니다. 프로젝트의 룰북은 Ansible 컬렉션의 이벤트 기반 Ansible 콘텐츠에 대해 정의된 경로( /extensions/eda/rulebooks )에 있어야 합니다.

### 3.1. 새 프로젝트 만들기

사전 요구 사항

- 이벤트 기반 Ansible 컨트롤러 대시보드에 콘텐츠 소비자로 로그인되어 있습니다.
- 필요한 경우 인증 정보를 설정했습니다. 자세한 내용은 [인증 정보 설정](#) 섹션을 참조하십시오.
- 자동화 컨트롤러에서 사용할 리포지토리에 포함된 플레이북과 통합된 룰북이 포함된 기존 리포지토리가 있습니다.

절차

1. 이벤트 기반 Ansible 컨트롤러 대시보드에 로그인합니다.
2. 탐색 패널에서 프로젝트를 생성을 선택합니다.
3. 다음을 삽입합니다.

이름

프로젝트 이름을 입력합니다.

설명

이 필드는 선택 사항입니다.

SCM 유형

Git은 사용할 수 있는 유일한 SCM 유형입니다.

SCM URL

리포지토리의 프로토콜 주소(예: GitHub 또는 GitLab)입니다.



참고

프로젝트를 생성한 후에는 SCM URL을 편집할 수 없습니다.

인증 정보

이 필드는 선택 사항입니다. SCM URL을 사용하는 데 필요한 토큰입니다.

4. 프로젝트를 생성을 선택합니다.

이제 프로젝트가 생성되고 프로젝트 화면에서 관리할 수 있습니다.

새 프로젝트를 저장하면 프로젝트의 세부 정보 페이지가 표시됩니다. 프로젝트 목록 보기 또는 프로젝트 목록 보기에서 편집하거나 삭제할 수 있습니다.

### 3.2. 프로젝트 목록 보기

프로젝트 페이지에서 Status 및 Git 해시와 함께 생성한 프로젝트를 볼 수 있습니다.



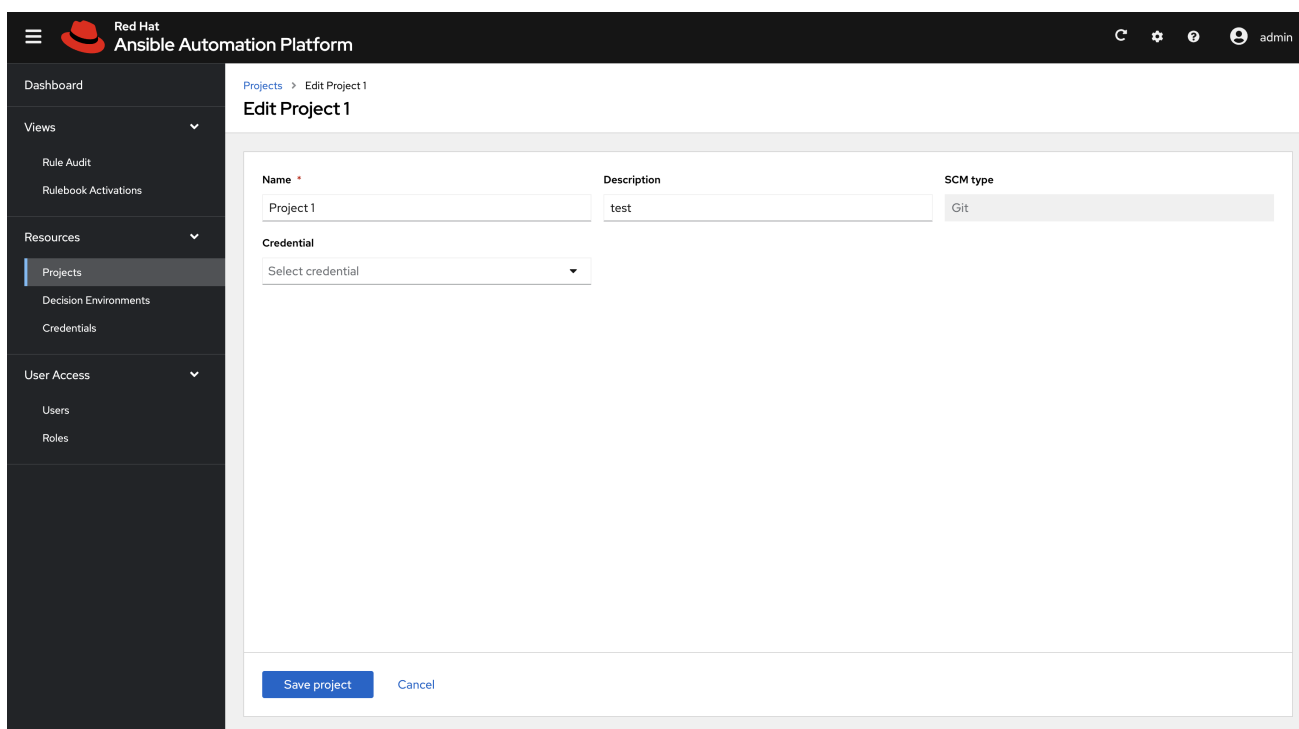
## 참고

소스 제어에서 룰북이 변경되면 프로젝트 목록 뷰에서 프로젝트 옆에 있는 동기화 아이콘을 선택하여 프로젝트를 다시 동기화할 수 있습니다. Git 해시 업데이트는 해당 리포지토리의 최신 커밋을 나타냅니다. 업데이트된 프로젝트를 사용하려면 활성화를 다시 시작하거나 다시 생성해야 합니다.

### 3.3. 프로젝트 편집

#### 절차

1. 프로젝트 목록 보기에서 원하는 프로젝트 옆에 있는 **More Actions** 아이콘을 선택합니다.
2. 프로젝트 편집을 선택합니다.
3. 필요한 변경 사항을 입력하고 **Save project** 를 선택합니다.



### 3.4. 프로젝트 삭제

#### 절차

1. 프로젝트 목록 보기에서 원하는 프로젝트 옆에 있는 **More Actions** 아이콘을 선택합니다.
2. 프로젝트 삭제를 선택합니다.
3. 팝업 창에서 **Yes**를 선택하고 이 프로젝트를 삭제하고 싶습니다.
4. 프로젝트 삭제를 선택합니다.

## 4장. 의사 결정 환경

의사 결정 환경은 Ansible 룰북을 실행하는 컨테이너 이미지입니다. 자동화 종속 항목을 전달하는 공통 언어를 생성하고 자동화 환경을 빌드하고 배포하는 표준 방법을 제공합니다. 기본 의사 결정 환경은 [Ansible-Rulebook](#)에서 확인할 수 있습니다.

고유한 의사 결정 환경을 생성하려면 [Ansible Automation Platform](#) 내에서 [이벤트 기반 Ansible](#)을 위한 사용자 정의 의사 결정 환경 빌드를 참조하십시오.

### 4.1. 새 의사 결정 환경 설정

다음 단계에서는 의사 결정 환경을 이벤트 기반 Ansible 컨트롤러 대시보드로 가져오는 방법을 설명합니다.

사전 요구 사항

- 이벤트 기반 Ansible 컨트롤러 대시보드에 콘텐츠 소비자로 로그인되어 있습니다.
- 필요한 경우 인증 정보를 설정했습니다. 자세한 내용은 [인증 정보 설정](#) 섹션을 참조하십시오.
- 의사 결정 환경 이미지를 이미지 리포지토리로 내보내거나 [registry.redhat.io](#)에서 제공하는 이미지 사용 중단을 선택했습니다.

절차

1. 이벤트 기반 Ansible 컨트롤러 대시보드로 이동합니다.
2. 탐색 패널에서 의사 결정 환경을 선택합니다.
3. 다음을 삽입합니다.

이름

이름을 삽입합니다.

설명

이 필드는 선택 사항입니다.

Image

컨테이너 레지스트리, 이미지 이름, 버전 태그를 포함한 전체 이미지 위치입니다.

인증 정보

이 필드는 선택 사항입니다. 이는 의사 결정 환경 이미지를 활용하는 데 필요한 토큰입니다.

4. 의사 결정 환경 생성을 선택합니다.

이제 의사 결정 환경이 생성되고 의사 결정 환경에서 관리할 수 있습니다.

새 의사 결정 환경을 저장하면 의사 결정 환경의 세부 정보 페이지가 표시됩니다. 의사 결정 환경 목록 보기에서 편집 또는 삭제할 수 있습니다.

### 4.2. ANSIBLE AUTOMATION PLATFORM 내에서 이벤트 기반 ANSIBLE을 위한 사용자 정의 의사 결정 환경 빌드

기본 의사 결정 환경에서 사용할 수 없는 사용자 지정 유지 관리 또는 타사 이벤트 소스 플러그인을 제공하는 사용자 정의 의사 결정 환경이 필요한 경우 다음 지침을 사용합니다.

## 사전 요구 사항

- Ansible Automation Platform > = 2.4
- 이벤트 기반 Ansible
- Ansible Builder > = 3.0

## 절차

- 지원되지 않는 의사 결정 환경을 추가합니다. 이 이미지는 Red Hat에서 제공하는 기본 이미지에서 빌드됩니다. **de-minimal**.



## 참고

Red Hat은 Ansible Builder에서 기본 이미지로 **de-minimal** 을 사용하여 사용자 정의 의사 결정 환경을 빌드하는 것이 좋습니다.

다음은 **de-minimal** 을 기본 이미지로 사용하여 **ansible.eda** 컬렉션을 사용하여 사용자 정의 의사 결정 환경을 빌드하는 Ansible 빌더 정의 파일의 예입니다.

```
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
```

또한 다른 Python 패키지 또는 RPM이 필요한 경우 단일 정의 파일에 다음을 추가할 수 있습니다.

```
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
```



## 5장. 자동화 컨트롤러 토큰 설정

자동화 컨트롤러에는 이벤트 기반 Ansible 룰북에서 작동하도록 설계된 특정 플레이북이 있는 리포지토리를 기반으로 하는 프로젝트가 포함되어야 합니다. 자동화 컨트롤러에는 해당 프로젝트의 플레이북을 기반으로 해당 작업 템플릿이 설정되어 있어야 합니다.

### 5.1. 자동화 컨트롤러에 인증할 토큰 설정

사전 요구 사항

- 이벤트 기반 Ansible 컨트롤러 대시보드에 콘텐츠 소비자로 로그인되어 있습니다.
- 사용자를 생성했습니다.
- 이벤트 기반 Ansible 컨트롤러 대시보드에 로그인하거나 조직에서 사용자로 추가할 수 있습니다.

절차

1. 이벤트 기반 Ansible 컨트롤러 대시보드로 이동합니다.
2. 상단 탐색 패널에서 프로필을 선택합니다.
3. 사용자 세부 정보로 이동합니다.
4. 컨트롤러 토큰 →을 선택합니다.
5. 다음을 삽입합니다.

이름

이름을 삽입합니다.

설명

이 필드는 선택 사항입니다.

토큰

자동화 컨트롤러에서 토큰을 생성합니다. 토큰 생성에 대한 자세한 내용은 [자동화 컨트롤러 사용자 가이드의 사용자 - 토큰](#) 섹션을 참조하십시오.



참고

토큰은 쓰기 범위에 있어야 하며 할당된 사용자인 경우 작업 및 워크플로우 템플릿을 실행할 수 있는 권한이 있어야 합니다.

6. 컨트롤러 토큰 생성을 선택합니다.

새 토큰을 저장한 후 토큰을 삭제할 수 있는 컨트롤러 토큰 탭으로 가져옵니다.

## 6장. 룰북 활성화

룰북 활성화는 특정 룰북을 실행하는 의사 결정 환경에 의해 정의된 백그라운드에서 실행되는 프로세스입니다.

### 6.1. 룰북 활성화 설정

사전 요구 사항

- 이벤트 기반 Ansible 컨트롤러 대시보드에 콘텐츠 소비자로 로그인되어 있습니다.
- 프로젝트를 설정했습니다.
- 의사 결정 환경을 설정했습니다.
- 자동화 컨트롤러 토큰을 설정했습니다.

절차

1. 이벤트 기반 Ansible 컨트롤러 대시보드로 이동합니다.
2. 탐색 패널에서 Rulebook Activations 를 선택합니다.
3. 다음을 삽입합니다.

이름

이름을 삽입합니다.

설명

이 필드는 선택 사항입니다.

프로젝트

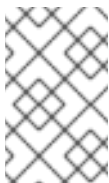
프로젝트는 룰북의 논리 컬렉션입니다.

룰북

룰북은 선택한 프로젝트에 따라 표시됩니다.

의사 결정 환경

의사 결정 환경은 Ansible 룰북을 실행하는 컨테이너 이미지입니다.



참고

이벤트 기반 Ansible 컨트롤러에서는 의사 결정 환경의 가져오기 정책을 사용자 지정할 수 없습니다. 기본적으로 always 정책의 동작을 따릅니다. 활성화가 시작될 때마다 시스템은 최신 버전의 이미지를 가져오려고 합니다.

재시작 정책

이 정책은 룰북을 다시 시작할 시기를 결정하는 정책입니다.

- 정책:
  - i. Always: 룰북이 완료되면 다시 시작
  - ii. never: 완료 시 룰북을 재시작하지 마십시오.

## iii. 실패 시: 실패하는 경우에만 재시작

룰북 활성화가 활성화됩니까?

이렇게 하면 룰북 활성화가 자동으로 활성화됩니다.

변수

룰북의 변수는 JSON/YAML 형식입니다. 콘텐츠는 `ansible-rulebook` 명령의 `--vars` 플래그를 통해 전달되는 파일과 동일합니다.

## 4. 룰북 활성화 생성을 클릭합니다.

이제 룰북 활성화가 생성되고 규칙북 활성화 페이지에서 관리할 수 있습니다.

새 룰북 활성화를 저장하면 룰북 활성화의 세부 정보 페이지가 표시됩니다. 규칙북 활성화 목록 보기에서 편집하거나 삭제할 수 있습니다.



## 참고

경우에 따라 소스 플러그인이 종료되면 일정 시간 후에 룰북이 정상적으로 종료됩니다. 룰북 활성화가 종료되면 수행 대기 중인 작업이 취소되고 정보 수준 메시지가 활성화 로그로 전송됩니다. 자세한 내용은 [규칙북](#) 을 참조하십시오.

## 6.2. 룰북 활성화 목록 보기

규칙북 활성화 페이지에서 활성화 상태, 룰북 과 관련된 규칙 수, Fire count, 재시작 수와 함께 생성한 룰북 활성화를 볼 수 있습니다.

활성화 상태가 Running 이면 룰북 활성화가 백그라운드에서 실행 중이고 룰북에 선언된 규칙에 따라 필요한 작업을 실행합니다.

규칙북 활성화 목록 보기에서 활성화를 선택하여 자세한 정보를 볼 수 있습니다.

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

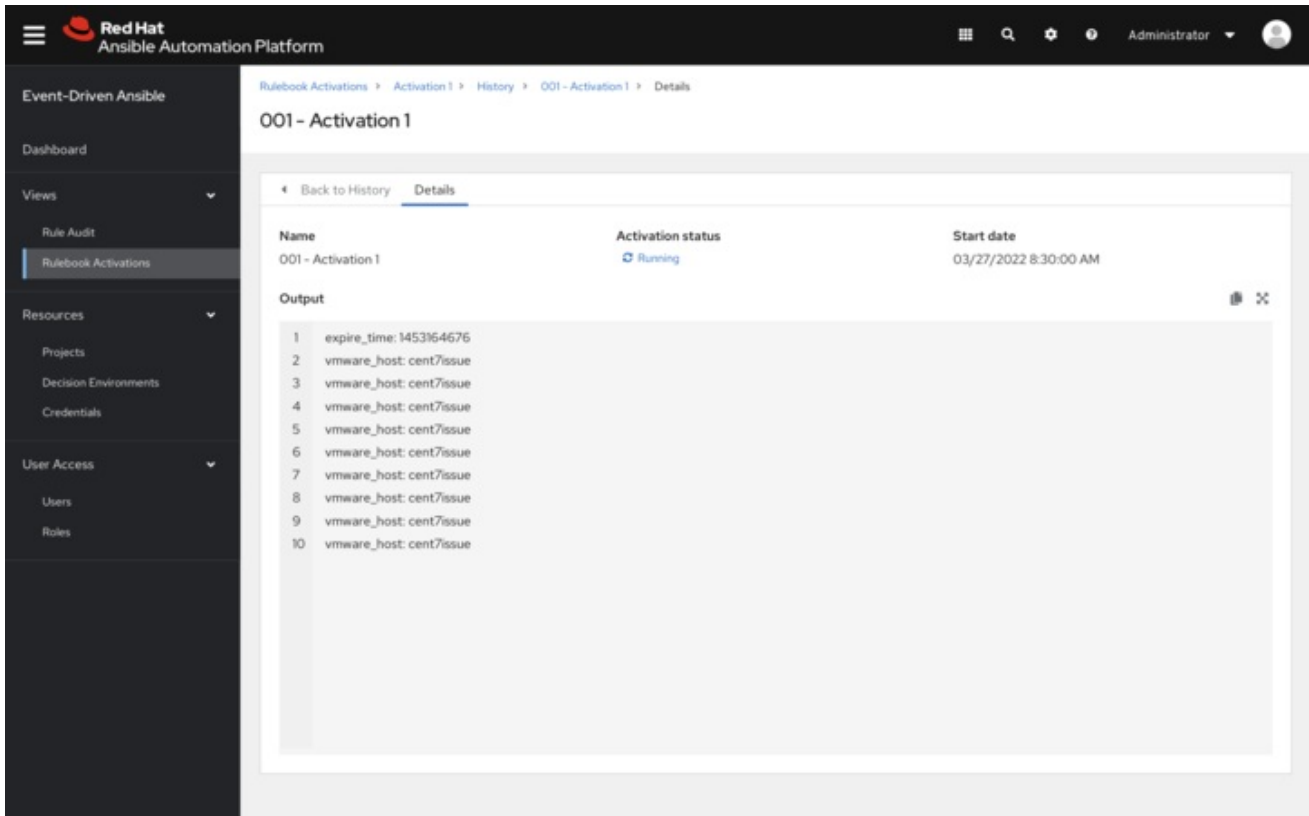
실행된 모든 활성화의 경우 세부 정보 및 기록 탭을 보고 발생한 사항에 대한 자세한 정보를 얻을 수 있습니다.

## 6.2.1. 활성화 출력 보기

활성화의 출력은 기록 탭에서 볼 수 있습니다.

절차

1. 내역 탭을 선택하여 모든 활성화 인스턴스 목록에 액세스합니다. 활성화 인스턴스는 활성화의 단일 실행을 나타냅니다.
2. 그런 다음 문제의 활성화 인스턴스를 선택하면 해당 특정 실행에 의해 생성된 출력이 표시됩니다.

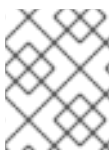


작업이 시작되어 트리거된 이벤트를 보려면 이벤트 기반 Ansible 컨트롤러 대시보드에서 **규칙 감사** 섹션을 사용할 수 있습니다.

### 6.3. 룰북 활성화 및 비활성화

1. 행 수준에서 스위치를 선택하여 선택한 룰북을 활성화하거나 비활성화합니다.
2. 팝업 창에서 **Yes**를 선택하여 이러한 X 룰북 활성화를 활성화/비활성화합니다.
3. **Enable/Disable rulebook activation** 을 선택합니다.

### 6.4. 룰북 활성화 다시 시작



참고

현재 활성화되어 있는 경우에만 룰북 활성화를 다시 시작할 수 있으며 재시작 정책이 생성된 경우 Always 로 설정되었습니다.

1. Rulebook Activation enabled/disabled 토글 옆에 있는 **More Actions** 아이콘을 선택합니다.
2. 룰북 활성화 재시작 을 선택합니다.

3. 팝업 창에서 **Yes, I confirm that I want to restart these X rulebook activations.**를 선택합니다.
4. 룰북 활성화 재시작 을 선택합니다.

## 6.5. 룰북 활성화 삭제

1. Rulebook Activation enabled/disabled 토글 옆에 있는 **More Actions** 아이콘을 선택합니다.
2. 룰북 활성화 삭제를 선택합니다.
3. 팝업 창에서 예 를 선택하고 이러한 X 룰북 활성화를 삭제하고 싶습니다.
4. 룰북 활성화 삭제를 선택합니다.

## 6.6. WEBHOOK 룰북 활성화

Openshift 환경에서는 룰북 활성화의 Kubernetes 서비스를 노출하는 경로를 생성하여 Webhook가 지정된 포트를 통해 activation-job-pod에 도달하도록 허용할 수 있습니다.

사전 요구 사항

- 이벤트 기반 Ansible 컨트롤러 대시보드에서 룰북 활성화를 생성했습니다.



참고

다음은 지정된 Webhook가 있는 룰북의 예입니다.

```
- 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
```

절차

1. 서비스를 노출할 경로(OpenShift Container Platform)를 생성합니다. 다음은 의사 결정 환경 Pod에서 포트 5000에 POST가 예상되는 ansible-rulebook 소스의 경로 예제입니다.

```
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

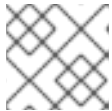
```

2. 경로를 생성할 때 경로 URL에 대한 Post로 테스트합니다.

```

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

```



#### 참고

Route(targetPort)에 지정되어 있으므로 포트가 필요하지 않습니다.

## 6.7. KUBERNETES로 테스트

Kubernetes를 사용하면 Ingress를 생성하거나 포트를 노출할 수 있지만 프로덕션에는 사용할 수 없습니다.

### 절차

1. 다음 명령을 실행하여 지정된 서비스의 클러스터에 포트를 노출합니다.

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

2. localhost:5000에 대해 HTTP 요청을 만들어 불복을 트리거합니다.

```

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

```

## 7장. 규칙 감사

규칙 감사를 사용하면 특정 시점에서 활성화된 모든 규칙에 의해 트리거된 규칙을 감사할 수 있습니다.

규칙 감사 목록 보기는 이벤트가 룰북 내의 조건과 일치하고 작업을 트리거할 때마다 목록을 표시합니다. 이 목록에는 룰북 내의 규칙이 표시되고 각 제목이 실행된 규칙과 일치합니다.

### 7.1. 규칙 감사 세부 정보 보기

규칙 감사 목록 뷰에서 특정 작업을 트리거한 이벤트를 확인할 수 있습니다.

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

#### 절차

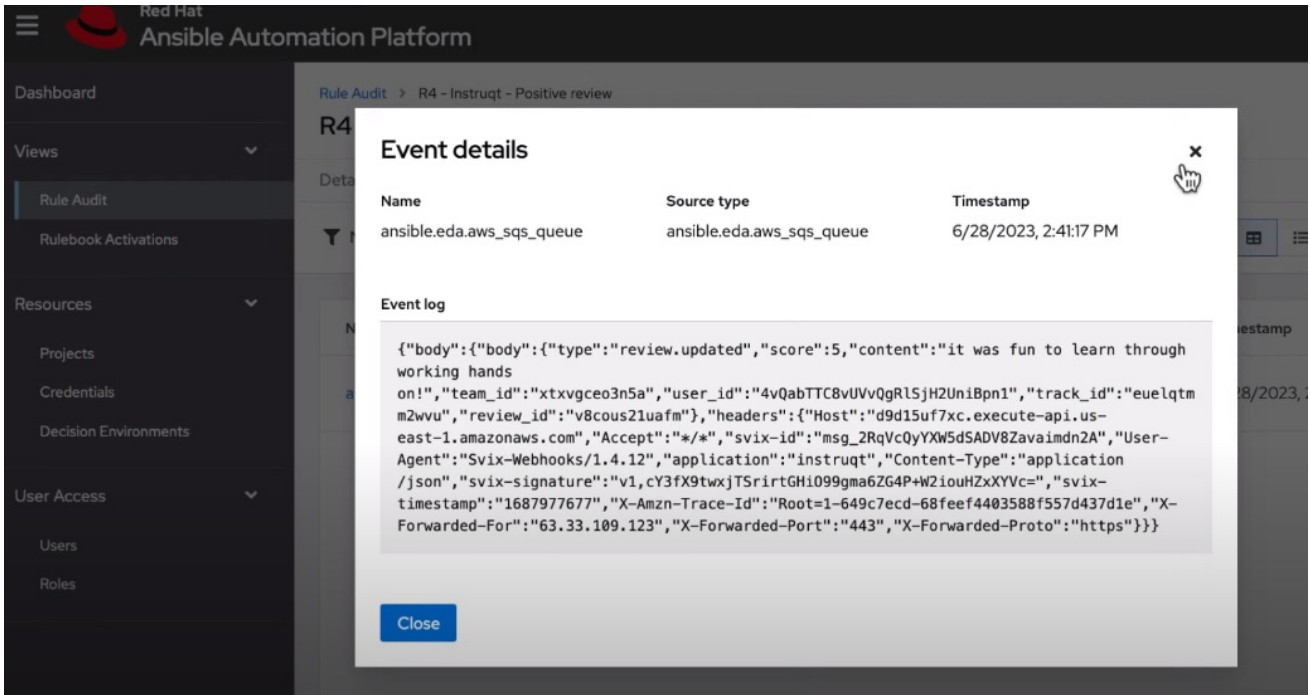
1. 탐색 패널에서 규칙 감사를 선택합니다.
2. 원하는 규칙을 선택하면 세부 정보 탭으로 이동합니다.

여기에서는 생성 시기, 마지막으로 실행된 시기, 룰북 활성화를 확인할 수 있습니다.

### 7.2. 규칙 감사 이벤트 보기

#### 절차

1. 탐색 패널에서 규칙 감사를 선택합니다.
2. 원하는 규칙을 선택하면 세부 정보 탭으로 이동합니다. 작업을 트리거한 모든 이벤트를 보려면 이벤트 탭을 선택합니다. 작업을 트리거한 이벤트가 표시됩니다.
3. 소스 유형 및 Timestamp 와 함께 이벤트 로그를 볼 이벤트를 선택합니다.



### 7.3. 규칙 감사 작업 보기

#### 절차

1. 탐색 패널에서 규칙 감사를 선택합니다.
2. 원하는 규칙을 선택하면 작업 탭으로 이동합니다.

여기에서 실행된 작업을 볼 수 있습니다. 일부 작업은 출력을 볼 수 있는 자동화 컨트롤러에 연결됩니다.



## 8장. 이벤트 기반 ANSIBLE 컨트롤러의 성능 튜닝

이벤트 기반 Ansible은 확장성이 뛰어나고 유연한 자동화 기능입니다. 이벤트 기반 Ansible 컨트롤러는 이벤트 기반 Ansible 자동화가 수행하는 인터페이스를 제공합니다. 다음을 통해 이벤트 기반 Ansible 컨트롤러를 조정하여 성능 및 확장성을 최적화합니다.

- 워크로드 특성화
- 시스템 수준 모니터링
- 성능 문제 해결

### 8.1. 워크로드 특성화

이벤트 기반 Ansible 컨트롤러에서 워크로드에는 룰북 활성화 및 수신되는 이벤트 수가 포함됩니다. 이벤트 기반 Ansible 컨트롤러 워크로드를 특성화하려면 다음 요인을 고려하십시오.

1. 동시 룰북 활성화 수
2. 이벤트 기반 Ansible 컨트롤러에서 수신하는 이벤트 수

#### 8.1.1. 동시 룰북 활성화 수 수정

기본적으로 이벤트 기반 Ansible 컨트롤러는 12개의 룰북 활성화를 동시에 실행할 수 있습니다. 12개 이상의 룰북 활성화가 생성되는 경우, 사용 가능한 룰북 활성화 작업자가 있을 때까지 후속 룰북 활성화가 보류 중이어야 합니다. 이 경우 이벤트 기반 Ansible 컨트롤러 인스턴스에 사용 가능한 메모리와 CPU가 충분한 경우에도 룰북 활성화 상태가 "보류 중"으로 표시됩니다. 이 동작을 변경하려면 기본 실행 중인 룰북 활성화 수를 변경해야 합니다.

참고: `EDA_MAX_RUNNING_ACTIVATIONS` 의 값은 인스턴스 크기 변경 시 변경되지 않으므로 수동으로 조정해야 합니다.

##### 8.1.1.1. 이벤트 기반 Ansible 컨트롤러 설치 중 동시 룰북 활성화 수 수정

기본적으로 이벤트 기반 Ansible 컨트롤러는 12개의 활성화를 동시에 실행할 수 있습니다. 다음 절차에 따라 설치 중에 이 기본값을 수정할 수 있습니다.

절차

VM 설치 프로그램에 변수를 제공합니다.

1. 설정 인벤토리 파일로 이동합니다.
2. `[all:vars]` 섹션에 `automationedacontroller_max_running_activations` 를 추가합니다. 예를 들어 `automationedacontroller_max_running_activations=16`.
3. 설정을 실행합니다.

##### 8.1.1.2. 이벤트 기반 Ansible 컨트롤러 설치 후 동시 룰북 활성화 수 수정

기본적으로 이벤트 기반 Ansible 컨트롤러는 12개의 활성화를 동시에 실행할 수 있습니다. 다음 절차에 따라 설치 후 이 기본값을 수정할 수 있습니다.

절차

1. `/etc/ansible-automation-platform/eda`의 환경 파일로 이동합니다.
2. 필요한 최대 실행 중인 활성화 수를 선택합니다. 예를 들어 `EDA_MAX_RUNNING_ACTIVATIONS = 16`
3. 다음 명령을 사용하여 EDA 서비스를 다시 시작합니다. `systemctl restart automation-eda-controller.target`

## Resources

롤북 활성화에 대한 자세한 내용은 규칙북 활성화를 참조하십시오.

[https://access.redhat.com/documentation/en-us/red\\_hat\\_ansible\\_automation\\_platform/2.4/html-single/event-driven\\_ansible\\_controller\\_user\\_guide/index#eda-rulebook-activations](https://access.redhat.com/documentation/en-us/red_hat_ansible_automation_platform/2.4/html-single/event-driven_ansible_controller_user_guide/index#eda-rulebook-activations)

### 8.1.2. 각 롤북 활성화에 대한 기본 메모리 제한 수정

메모리 사용량은 이벤트 기반 Ansible 컨트롤러에서 처리해야 하는 이벤트 수를 기반으로 합니다. 각 롤북 활성화 컨테이너에는 200MB 메모리 제한이 있습니다. 예를 들어 CPU 4GB와 16GB의 RAM을 사용하면 할당된 200MB 메모리 제한이 있는 하나의 롤북 활성화 컨테이너는 분당 150,000개 이상의 이벤트를 처리할 수 없습니다. 병렬 실행 롤북 활성화 수가 높으면 각 롤북 활성화가 처리할 수 있는 최대 이벤트 수가 줄어듭니다. 매우 높은 속도로 들어오는 이벤트가 너무 많으면 컨테이너는 이벤트를 처리하려는 메모리가 부족할 수 있습니다. 그러면 컨테이너가 종료되고 롤북 활성화가 137개의 상태 코드와 함께 실패합니다.

이 오류를 해결하기 위해 다음 절차 중 하나를 사용하여 높은 속도로 많은 이벤트를 처리하기 위해 롤북 활성화에 할당된 메모리 양을 늘릴 수 있습니다.

- 설치 중 각 롤북 활성화의 기본 메모리 제한 수정
- 설치 후 각 롤북 활성화의 기본 메모리 제한 수정

#### 8.1.2.1. 설치 중 각 롤북 활성화의 기본 메모리 제한 수정

기본적으로 각 롤북 활성화 컨테이너에는 200MB 메모리 제한이 있습니다. 다음 절차에 따라 설치 중에 이 기본값을 수정할 수 있습니다.

##### 절차

1. 설정 인벤토리 파일로 이동합니다.
2. `[all:vars]` 섹션에 `automationedacontroller_podman_mem_limit`를 추가합니다. 예를 들어 `automationedacontroller_podman_mem_limit='400m'`입니다.
3. 설정을 실행합니다.

#### 8.1.2.2. 설치 후 각 롤북 활성화의 기본 메모리 제한 수정

기본적으로 각 롤북 활성화 컨테이너에는 200MB 메모리 제한이 있습니다. 다음 절차에 따라 설치 후 이 기본값을 수정할 수 있습니다.

##### 절차

1. `/etc/ansible-automation-platform/eda`의 환경 파일로 이동합니다.

2. 기본 컨테이너 메모리 제한을 수정합니다. 예를 들어 `EDA_PODMAN_MEM_LIMIT = '300m'` 입니다.
3. `systemctl restart automation-eda-controller.target` 을 사용하여 Event-Driven Ansible 컨트롤러 서비스를 다시 시작합니다.

## 8.2. 이벤트 기반 ANSIBLE 컨트롤러에 대한 시스템 수준 모니터링

병렬로 실행 중인 룰북 활성화 수와 지정된 시점에서 수신하는 이벤트 수를 결정하기 위해 워크로드를 지정 한 후 시스템 수준에서 이벤트 기반 Ansible 컨트롤러 호스트를 모니터링하는 것이 좋습니다. 시스템 수준 모니터링을 사용하여 시간이 지남에 따라 이벤트 기반 Ansible의 성능에 대한 정보를 검토하면 문제를 진단하거나 향후 성장을 위한 용량을 고려할 때 도움이 됩니다.

시스템 수준 모니터링에는 다음 정보가 포함됩니다.

- 디스크 I/O
- RAM 사용
- CPU 사용
- 네트워크 트래픽

CPU, RAM 또는 디스크 사용률이 높으면 Event-Driven Ansible 컨트롤러의 전반적인 성능에 영향을 미칠 수 있습니다. 예를 들어 이러한 시스템 수준 리소스의 높은 사용률은 Event-Driven Ansible 컨트롤러가 너무 많은 룰북 활성화를 실행 중이거나 일부 개별 룰북 활성화 중 하나가 많은 리소스를 사용하고 있음을 나타냅니다. 이 경우 워크로드를 지원하기 위해 시스템 수준 리소스를 늘려야 합니다.

## 8.3. 이벤트 기반 ANSIBLE 컨트롤러의 성능 문제 해결

이벤트 기반 Ansible 컨트롤러 내의 기본 매개변수를 기반으로 워크로드를 완료하는 데 어려움이 있는 시나리오가 발생할 수 있습니다. 다음 섹션에서는 이러한 시나리오 및 문제 해결 지침을 설명합니다.

- 내 활성화 상태는 "실행 중"으로 표시되지만 이벤트를 처리하지는 않습니다.
  - 룰북 활성화에서 올바른 이벤트 소스를 사용하고 있는지 확인합니다. 예상되는 이벤트가 룰북에 있는 소스 이외의 소스에서 제공되는 경우 이벤트 기반 Ansible 컨트롤러에서 이벤트를 처리하지 않습니다.
- 내 활성화 상태가 "실행 중"으로 표시되고 이벤트 기반 Ansible 컨트롤러도 이벤트를 수신하지만 작업이 발생하지 않습니다.
  - 이벤트 일치 및 룰북 활성화에서 작업을 수행하는 데 필요한 올바른 조건을 설정해야 합니다.
- 활성화는 무한 루프로 다시 시작됩니다.
  - 기본적으로 룰북 활성화를 위한 재설정 정책은 "실패 시"로 설정됩니다. 다음 절차를 사용하여 `estart` 정책을 변경합니다.
    1. Rulebook 활성화 화면으로 이동합니다.
    2. Restart Policy 드롭다운 메뉴를 클릭합니다.
    3. "실패 시", "항상", "Never"를 선택합니다.

