



# Red Hat Ansible Automation Platform 2.4

## Red Hat Ansible Automation Platform Creator Guide

Ansible을 사용하여 자동화 콘텐츠 생성 방법 알아보기



# Red Hat Ansible Automation Platform 2.4 Red Hat Ansible Automation Platform Creator Guide

---

Ansible을 사용하여 자동화 콘텐츠 생성 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

이 가이드에서는 개발자가 Ansible을 사용하여 자동화를 위한 콘텐츠를 생성하는 방법을 배울 수 있도록 지원합니다.

<b>차례</b>	
RED HAT 문서에 관한 피드백 제공 .....	3
1장. 머리말 .....	4
2장. 콘텐츠 작성자 워크플로 및 자동화 실행 환경 소개 .....	5
2.1. 콘텐츠 워크플로 정보 .....	5
2.2. 아키텍처 개요 .....	5
3장. ANSIBLE 개념 이해 .....	6
3.1. 사전 요구 사항 .....	6
3.2. ANSIBLE 플레이북 정보 .....	6
3.3. ANSIBLE 역할 정보 .....	6
3.4. 콘텐츠 컬렉션 정보 .....	6
3.5. 실행 환경 정보 .....	7
4장. 틀 및 구성 요소 .....	8
4.1. ANSIBLE 빌더 정보 .....	8
4.2. 자동화 콘텐츠 사용 .....	8
4.3. 자동화 허브 정보 .....	8
4.4. ANSIBLE 명령줄 인터페이스 정보 .....	8
4.5. 추가 리소스 .....	9
5장. 개발 환경 설정 .....	10
5.1. ANSIBLE 빌더 설치 .....	10
5.2. RPM에서 RHEL에 자동화 콘텐츠 설치 .....	10
5.3. 기본 자동화 실행 환경 다운로드 .....	11
6장. 콘텐츠 생성 .....	12
6.1. 플레이북 생성 .....	12
6.2. 컬렉션 생성 .....	12
6.3. 역할 생성 .....	13
6.4. 자동화 실행 환경 생성 .....	14
7장. 기존 콘텐츠 마이그레이션 .....	16
7.1. 가상 환경을 자동화 실행 환경으로 마이그레이션 .....	16
7.2. ANSIBLE CORE 버전 간 마이그레이션 .....	17
8장. 자동화 콘텐츠로 콘텐츠 실행 .....	18
8.1. 자동화 콘텐츠로 ANSIBLE 플레이북 실행 .....	18
9장. CONCLUSION .....	21



## 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 **docs-product** 구성 요소를 사용하여 Ansible Automation Platform Jira 프로젝트에 문제를 생성하십시오.

## 1장. 머리말

자동화 실행 환경을 사용하여 Red Hat Ansible Automation Platform 내에서 콘텐츠 자동화

실행 환경을 재현 가능하고 이식 가능하며 일관성 있고 공유할 수 있는 컨테이너 이미지로 사용할 수 있습니다. Ansible Automation Platform 작업 런타임 환경의 모든 종속 항목을 컬렉션 형태로 시스템 종속 항목, Python 종속 항목, Ansible 버전 및 Ansible 콘텐츠에서 제어합니다.



## 2장. 콘텐츠 작성자 워크플로 및 자동화 실행 환경 소개

### 2.1. 콘텐츠 워크플로 정보

Red Hat Ansible Automation Platform 2.0 이전에는 자동화 콘텐츠 개발자에게 이를 관리하기 위해 자체 자동화가 필요한 많은 Python 가상 환경이 필요할 수 있습니다. 이러한 수준의 복잡성을 줄이기 위해 Ansible Automation Platform 2.0은 가상 환경에서 벗어나 자동화 실행 환경이라는 컨테이너를 사용하여 쉽게 빌드하고 관리할 수 있으며 팀 및 조직 전반에서 공유될 수 있습니다.

자동화 컨트롤러가 자동화 실행 환경 사용으로 전환하면 자동화 콘텐츠 탐색 및 Ansible Builder와 같은 툴을 통해 자체 개발 시스템 내에서 이러한 자동화 실행 환경을 로컬로 활용할 수 있습니다.

#### 추가 리소스

- 자동화 콘텐츠 탐색을 사용하는 방법에 대한 자세한 내용은 [Automation Content Navigator Creator Guide](#) 를 참조하십시오.
- Ansible 빌더에 대한 자세한 내용은 [실행 환경 생성 및 사용](#) 을 참조하십시오.

### 2.2. 아키텍처 개요

다음 목록은 Ansible Automation Platform 2.0에서 사용할 수 있는 툴의 정렬 및 사용과 활용 방법을 보여줍니다.

- 자동화 콘텐츠로 이동하여 현재 Ansible Automation Platform 1.2에서 사용할 수 있습니다.
- 자동화 콘텐츠 탐색기 + 다운로드된 자동화 실행 환경 - 랩탑/워크스테이션에서 직접 사용
- 자동화 콘텐츠 탐색 및 + 다운로드된 자동화 실행 환경 + 자동화 컨트롤러 - 로컬로 푸시/실행하기 위한 → 원격으로
- 자동화 콘텐츠 탐색기 + 자동화 컨트롤러 + Ansible Builder + Layered 사용자 지정 EE - 자동화 작업 실행 방법에 대한 활용된 콘텐츠를 보다 효과적으로 제어 가능

## 3장. ANSIBLE 개념 이해

자동화 개발자는 다음 Ansible 개념을 검토하여 Ansible 개발 프로젝트를 시작하기 전에 성공적인 Ansible 플레이북 및 자동화 실행 환경을 생성합니다.

### 3.1. 사전 요구 사항

- Ansible이 설치되어 있어야 합니다. Ansible 설치에 대한 자세한 내용은 Ansible 설명서의 [Ansible 설치](#)를 참조하십시오.

### 3.2. ANSIBLE 플레이북 정보

플레이북은 사용자가 읽을 수 있는 특정 명령 집합 또는 "플레이"가 포함된 단일 대상 또는 대상 그룹에서 실행되도록 보내는 YAML로 작성된 파일입니다.

플레이북을 사용하여 원격 시스템에 대한 구성 및 배포는 물론 롤링 업데이트와 관련된 시퀀스 다중 계층 롤아웃을 관리할 수 있습니다. 플레이북을 사용하여 모니터링 서버 및 로드 밸런서와 상호 작용하여 다른 호스트에 작업을 위임합니다. 작성되고 나면 자동화를 위해 기업 전체에서 플레이북을 반복적으로 사용할 수 있습니다.

### 3.3. ANSIBLE 역할 정보

역할은 알려진 파일 구조를 사용하여 Ansible의 자동화 콘텐츠를 번들링하는 방법이며 관련 vars, 파일, 작업, 처리기 및 기타 아티팩트를 자동으로 로드하는 것입니다. 수백 개의 작업으로 대규모 플레이북을 생성하는 대신 역할을 사용하여 작업을 더 작고 개별 작업 단위로 분할할 수 있습니다.

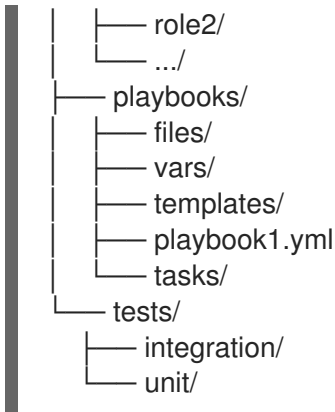
Ansible Galaxy에서 인프라 프로비저닝, 애플리케이션 배포 및 매일 수행하는 모든 작업을 찾을 수 있습니다. 유형별로 검색을 필터링하고 역할을 선택합니다. 관심 있는 역할을 찾으면 Ansible과 함께 제공되는 **ansible-galaxy** 명령을 사용하여 다운로드할 수 있습니다.

```
$ ansible-galaxy role install username.rolename
```

### 3.4. 콘텐츠 컬렉션 정보

Ansible Content Collection은 자동화를 위해 바로 사용할 수 있는 툴킷입니다. 플레이북, 역할, 모듈 및 플러그인과 같은 여러 유형의 콘텐츠가 모두 한 곳에 포함됩니다. 다음 다이어그램에서는 컬렉션의 기본 구조를 보여줍니다.

```
collection/
├── docs/
├── galaxy.yml
├── meta/
│   └── runtime.yml
├── plugins/
│   ├── modules/
│   │   └── module1.py
│   ├── inventory/
│   ├── lookup/
│   ├── filter/
│   └── .../
├── README.md
├── roles/
│   └── role1/
```



Red Hat Ansible Automation Platform에서 자동화 허브는 Ansible 인증 콘텐츠 컬렉션의 소스 역할을 합니다.

### 3.5. 실행 환경 정보

자동화 실행 환경은 Ansible 제어 노드 역할을 하는 일관되고 공유 가능한 컨테이너 이미지입니다. 자동화 실행 환경은 외부 종속 항목이 있는 Ansible 콘텐츠를 공유하는 문제를 줄입니다.

자동화 실행 환경에는 다음이 포함됩니다.

- Ansible Core
- Ansible Runner
- Ansible 컬렉션
- Python 라이브러리
- 시스템 종속 항목
- 사용자 정의 사용자 요구 사항

Ansible Builder를 사용하여 자동화 실행 환경을 정의하고 생성할 수 있습니다.

#### 추가 리소스

- Ansible 빌더에 대한 자세한 내용은 [실행 환경 생성 및 사용](#)을 참조하십시오.

## 4장. 툴 및 구성 요소

자동화 실행 환경을 생성하는 데 사용할 Red Hat Ansible Automation Platform 툴 및 구성 요소에 대해 자세히 알아보십시오.

### 4.1. ANSIBLE 빌더 정보

Ansible Builder는 다양한 Ansible 컬렉션 또는 사용자가 정의한 메타데이터를 사용하여 자동화 실행 환경을 빌드하는 프로세스를 자동화하는 명령줄 툴입니다.

Ansible Builder가 개발되기 전에 Red Hat Ansible Automation Platform 사용자는 필요한 모든 종속 항목이 포함된 사용자 지정 가상 환경 또는 컨테이너를 생성할 때 종속성 문제 및 오류를 실행할 수 있습니다.

이제 Ansible Builder를 사용하면 컬렉션, 타사 Python 요구 사항 및 시스템 수준 패키지와 같은 자동화 실행 환경에 포함할 콘텐츠를 지정하는 사용자 지정 가능한 자동화 실행 환경 정의 파일을 쉽게 생성할 수 있습니다. 이를 통해 필요한 모든 요구 사항과 종속 항목을 충족하여 작업을 실행할 수 있습니다.



#### 참고

Red Hat은 현재 자동화 자동화 실행 환경을 빌드할 때 자체 컨테이너 이미지를 제공하기로 선택한 사용자를 지원하지 않습니다.

### 4.2. 자동화 콘텐츠 사용

Automation Content navigator는 텍스트 기반 사용자 인터페이스를 사용하는 명령줄 콘텐츠에 중점을 둔 툴입니다. 자동화 콘텐츠를 사용하여 다음을 수행할 수 있습니다.

- 작업 및 플레이북을 시작하고 조사합니다.
- 저장된, 완료된 플레이북 및 작업 실행 아티팩트를 JSON 형식으로 공유합니다.
- 자동화 실행 환경을 찾아 인트로스펙션합니다.
- 파일 기반 인벤토리를 찾습니다.
- Ansible 모듈을 렌더링하고 플레이북에서 사용할 수 있는 예제를 추출합니다.
- 사용자 인터페이스에서 자세한 명령 출력을 확인합니다.

### 4.3. 자동화 허브 정보

Automation Hub는 Red Hat 및 기술 파트너가 지원하는 콘텐츠를 신속하게 찾아서 사용하여 가장 까다로운 환경에 대한 추가 지원을 제공할 수 있는 환경을 제공합니다.

높은 수준에서 자동화 허브는 참여 및 인증된 지원 콘텐츠를 제공하는 모든 파트너에 대한 개요를 제공합니다.

중앙 보기에서 사용자는 각 파트너에 대해 자세히 살펴보고 컬렉션을 확인할 수 있습니다.

또한 사용 가능한 모든 컬렉션에 대한 검색 가능한 개요를 사용할 수 있습니다.

### 4.4. ANSIBLE 명령줄 인터페이스 정보

명령줄에서 Ansible을 사용하는 것은 자주 반복되지 않는 작업을 실행하는 데 유용합니다. 반복된 작업을 처리하는 데 권장되는 방법은 플레이북을 작성하는 것입니다.

명령줄에서 Ansible에 대한 애드혹 명령은 다음 구조를 따릅니다.

```
$ ansible [pattern] -m [module] -a "[module options]"
```

## 4.5. 추가 리소스

- Ansible을 명령줄 툴로 사용하는 방법에 대한 자세한 내용은 [Ansible 사용자 가이드](#)의 [명령줄 툴 작업을 참조하십시오](#).
- 자동화 허브에 콘텐츠를 [업로드하려면](#) [Ansible Automation Platform 제품 설명서](#)의 [자동화 허브에 콘텐츠 업로드](#)를 참조하십시오.

## 5장. 개발 환경 설정

이 섹션의 절차에 따라 자동화 실행 환경을 생성하기 위해 개발 환경을 설정할 수 있습니다.

### 5.1. ANSIBLE 빌더 설치

#### 사전 요구 사항

- Podman 컨테이너 런타임을 설치했습니다.
- 호스트에 유효한 서브스크립션이 연결되어 있습니다. 이렇게 하면 **ansible-builder** 를 설치하는 데 필요한 서브스크립션 전용 리소스에 액세스하고 **ansible-builder** 에 필요한 리포지토리가 자동으로 활성화되어 있는지 확인할 수 있습니다. 자세한 내용은 [Red Hat Ansible Automation Platform 서브스크립션 연결](#)을 참조하십시오.

#### 절차

1. 터미널에서 다음 명령을 실행하여 Ansible Automation Platform 리포지토리를 활성화합니다.

```
# dnf install --enablerepo=ansible-automation-platform-2.4-for-rhel-9-x86_64-rpms ansible-builder
```

### 5.2. RPM에서 RHEL에 자동화 콘텐츠 설치

RPM에서 RHEL(Red Hat Enterprise Linux)에 자동화 콘텐츠를 설치할 수 있습니다.

#### 사전 요구 사항

- RHEL 8.6 이상을 설치했습니다.
- Red Hat Subscription Manager에 시스템을 등록하셨습니다.



#### 참고

현재 Red Hat Ansible Automation Platform 환경과 일치하는 navigator만 설치해야 합니다.

#### 절차

1. Red Hat Ansible Automation Platform SKU 연결:

```
$ subscription-manager attach --pool=<sku-pool-id>
```

2. 다음 명령을 사용하여 자동화 콘텐츠를 설치합니다.

V.2.4 for RHEL 8 for x86\_64

```
$ sudo dnf install --enablerepo=ansible-automation-platform-2.4-for-rhel-8-x86_64-rpms ansible-navigator
```

V.2.4 for RHEL 9 for x86-64

```
$ sudo dnf install --enablerepo=ansible-automation-platform-2.4-for-rhel-9-x86_64-rpms
ansible-navigator
```

## 검증

- 자동화 콘텐츠 탐색 설치를 확인합니다.

```
$ ansible-navigator --help
```

다음 예제에서는 설치에 성공하는 방법을 보여줍니다.

```
$ ansible-navigator --help
usage: ansible-navigator [-h] [--version] [--cdcp COLLECTION_DOC_CACHE_PATH] [--ce CONTAINER_ENGINE] [--dc DISPLAY_COLOR] [--ecmd EDITOR_COMMAND]
                        [--econ EDITOR_CONSOLE] [--ee EXECUTION_ENVIRONMENT] [--eei EXECUTION_ENVIRONMENT_IMAGE]
                        [--eev EXECUTION_ENVIRONMENT_VOLUME_MOUNTS [EXECUTION_ENVIRONMENT_VOLUME_MOUNTS ...]] [--la LOG_APPEND] [--lf LOG_FILE]
                        [--ll LOG_LEVEL] [-m MODE] [--osc4 OSC4] [--penv PASS_ENVIRONMENT_VARIABLE [PASS_ENVIRONMENT_VARIABLE ...]]
                        [--pp PULL_POLICY] [--senv SET_ENVIRONMENT_VARIABLE [SET_ENVIRONMENT_VARIABLE ...]]
                        {subcommand} --help ...

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit

<... output truncated ...>

Subcommands:
  {subcommand} --help
  collections           Explore available collections
  config                Explore the current ansible configuration
  doc                   Review documentation for a module or plugin
  images                Explore execution environment images
  inventory             Explore an inventory
  replay                Explore a previous run using a playbook artifact
  run                   Run a playbook
  welcome               Start at the welcome page
```

## 5.3. 기본 자동화 실행 환경 다운로드

Ansible Automation Platform 2.0과 함께 제공되는 기본 이미지는 Red Hat Ecosystem Catalog(registry.redhat.io)에서 호스팅됩니다.

### 사전 요구 사항

- 유효한 Red Hat Ansible Automation Platform 서브스크립션이 있어야 합니다.

### 절차

1. registry.redhat.io에 로그인

```
$ podman login registry.redhat.io
```

2. 레지스트리에서 기본 이미지를 가져옵니다.

```
$ podman pull registry.redhat.io/aap/<image name>
```

## 6장. 콘텐츠 생성

Red Hat Ansible Automation Platform에서 사용할 콘텐츠 개발에 대해 자세히 알아보려면 *크리에이터 가이드*의 이 섹션에 있는 지침을 사용하십시오.

### 6.1. 플레이북 생성

플레이북에는 하나 이상의 플레이가 포함되어 있습니다. 기본 플레이에는 다음 섹션이 포함되어 있습니다.

- **name:** 플레이북의 전반적인 기능에 대한 간략한 설명으로, 모든 사용자가 읽고 구성하는 데 도움이 됩니다.
- **hosts:** Ansible이 실행할 대상을 식별합니다.
- **become 문:** 이 선택적 문을 **true/yes** 로 설정하여 become 플러그인 (예: **sudo, su, pexec, doas, pbrun, dzdo, ksu**) 을 사용하여 권한 에스컬레이션을 활성화할 수 있습니다.
- **tasks:** 플레이의 각 호스트에 대해 실행되는 목록 작업입니다.

#### 플레이북의 예

```
- name: Set Up a Project and Job Template
hosts: host.name.ip
become: true

tasks:
  - name: Create a Project
    ansible.controller.project:
      name: Job Template Test Project
      state: present
      scm_type: git
      scm_url: https://github.com/ansible/ansible-tower-samples.git

  - name: Create a Job Template
    ansible.controller.job_template:
      name: my-job-1
      project: Job Template Test Project
      inventory: Demo Inventory
      playbook: hello_world.yml
      job_type: run
      state: present
```

### 6.2. 컬렉션 생성

Ansible Galaxy CLI 툴을 사용하여 고유한 컬렉션을 로컬로 생성할 수 있습니다. **collection** 하위 명령을 사용하여 컬렉션별 명령을 활성화할 수 있습니다.

#### 사전 요구 사항

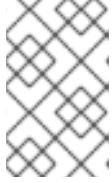
- 개발 환경에 Ansible-core 버전 2.15 이상이 설치되어 있어야 합니다.

#### 절차



1. 터미널에서 네임스페이스 루트 디렉토리를 원하는 위치로 이동합니다. 단순화를 위해 `COLLECTIONS_PATH`의 경로 여야 하지만 필수는 아닙니다.
2. `my_namespace` 및 `my_collection_name` 을 고유한 값으로 교체하여 다음 명령을 실행합니다.

```
$ ansible-galaxy collection init <my_namespace>.<my_collection_name>
```



### 참고

galaxy.ansible.com 또는 console.redhat.com/ansible/automation-hub의 **My Content** 탭에서 네임스페이스에 업로드할 수 있는 적절한 권한이 있는지 확인합니다.

이전 명령은 네임스페이스 인수에서 라는 디렉토리를 만들고(아직 없는 경우) 컬렉션 이름으로 디렉토리를 생성합니다. 해당 디렉토리의 내부는 default 또는 "skeleton" Collection이 됩니다. 여기에서 역할 또는 플러그인을 추가하고 자체 컬렉션 개발을 시작할 수 있습니다.

컬렉션 개발자는 실행 환경과 관련하여 Ansible Builder에서 적절한 메타데이터를 제공하여 콘텐츠에 대한 요구 사항을 선언할 수 있습니다.

컬렉션의 요구 사항은 다음과 같은 방법으로 인식할 수 있습니다.

- Python 또는 **bindep** 요구 사항 파일을 참조하는 file **meta/execution-environment.yml**.
- **requirements.txt** 라는 파일은 Python 종속 항목에 대한 정보를 포함하며 경우에 따라 컬렉션의 루트 수준에서 찾을 수 있습니다.
- 시스템 수준 종속성을 포함하는 **bindep.txt** 라는 파일은 종종 컬렉션의 루트 수준에서 찾을 수 있습니다.
- 이러한 파일 중 컬렉션 **build\_ignore** 에 있는 경우 Ansible Builder는 이러한 파일을 선택하지 않습니다. **build\_ignore** 섹션은 빌드 아티팩트에 포함되지 않아야 하는 모든 파일 또는 디렉토리를 필터링합니다.

컬렉션 유지 관리자는 **introspect** 명령을 사용하여 ansible-builder에서 예상 요구 사항을 인식하는지 확인할 수 있습니다.

```
$ ansible-builder introspect --sanitize ~/.ansible/collections/
```

### 추가 리소스

- 컬렉션 생성에 대한 자세한 내용은 Ansible 개발자 가이드에서 [컬렉션 생성](#) 을 참조하십시오.

## 6.3. 역할 생성

Ansible Galaxy CLI 툴을 사용하여 역할을 생성할 수 있습니다. **roles** 하위 명령에서 역할별 명령에 액세스할 수 있습니다.

```
ansible-galaxy role init <role_name>
```

컬렉션 외부의 독립 실행형 역할은 계속 지원되지만 Ansible Automation Platform에서 제공하는 모든 기능을 활용하기 위해 컬렉션 내에 새 역할을 생성합니다.

### 절차

1. 터미널에서 컬렉션 내의 **roles** 디렉터리로 이동합니다.
2. 컬렉션 내에 **role\_name** 이라는 역할을 생성합니다.

```
$ ansible-galaxy role init my_role
```

이제 컬렉션에 역할 디렉터리 내에 **my\_role** 역할이 포함됩니다.

```
~/ansible/collections/ansible_collections/<my_namespace>/<my_collection_name>
...
├── roles/
│   └── my_role/
│       ├── .travis.yml
│       ├── README.md
│       ├── defaults/
│       │   └── main.yml
│       ├── files/
│       ├── handlers/
│       │   └── main.yml
│       ├── meta/
│       │   └── main.yml
│       ├── tasks/
│       │   └── main.yml
│       ├── templates/
│       ├── tests/
│       │   ├── inventory
│       │   └── test.yml
│       └── vars/
│           └── main.yml
```

3. **--role-skeleton** 인수를 사용하여 사용자 지정 역할 skeleton 디렉터리를 제공할 수 있습니다. 이를 통해 조직은 새로운 역할에 맞게 표준화된 템플릿을 생성할 수 있습니다.

```
ansible-galaxy role init my_role --role-skeleton ~/role_skeleton
```

이렇게 하면 **~/role\_skeleton**의 콘텐츠를 **my\_role**에 복사하여 **my\_role**이라는 역할이 생성됩니다. **role\_skeleton**의 내용은 역할 디렉터리 내에서 유효한 파일 또는 폴더일 수 있습니다.

### 추가 리소스

- 역할 생성에 대한 자세한 내용은 Ansible Galaxy 설명서에서 [역할 생성](#)을 참조하십시오.

## 6.4. 자동화 실행 환경 생성

자동화 실행 환경 정의 파일은 다음을 지정합니다.

- Ansible 버전
- Python 버전(기본값: 시스템 Python)
- 필요한 Python 라이브러리 세트
- 제로 이상의 콘텐츠 컬렉션 (선택 사항)

- 이러한 특정 컬렉션의 Python 종속 항목

환경에 컬렉션 집합을 지정하는 개념은 종속성을 해결 및 설치하는 것입니다. 컬렉션 자체는 자동화 실행 환경을 생성하는 머신에 설치할 필요가 없습니다.

자동화 실행 환경은 이 정의에서 빌드되며 컨테이너 이미지가 생성됩니다. 이러한 이미지 생성과 관련된 단계를 알아보려면 [Ansible Builder 설명서](#)를 참조하십시오.

## 7장. 기존 콘텐츠 마이그레이션

다음 섹션에서는 **awx-manage** 명령을 사용하여 Red Hat Ansible Automation Platform 2.0 및 자동화 컨트롤러 4.0으로 업그레이드한 후 마이그레이션 프로세스의 추가 단계를 지원하는 방법을 알아봅니다. 또한 Ansible 버전 간 마이그레이션에 대해 자세히 알아보십시오.

### 7.1. 가상 환경을 자동화 실행 환경으로 마이그레이션

Red Hat Ansible Automation Platform 2.0 및 자동화 컨트롤러 4.0으로 업그레이드되면 다음 섹션을 사용하여 마이그레이션 프로세스의 추가 단계를 지원합니다.

#### 7.1.1. 사용자 정의 가상 환경 나열

**awx-manage** 명령을 사용하여 자동화 컨트롤러 인스턴스의 가상 환경을 나열할 수 있습니다.

##### 절차

1. 자동화 컨트롤러 인스턴스에 SSH를 사용하여 다음을 실행합니다.

```
$ awx-manage list_custom_venvs
```

검색된 가상 환경 목록이 표시됩니다.

```
# Discovered virtual environments:
/var/lib/awx/venv/testing
/var/lib/venv/new_env
```

```
To export the contents of a virtual environment, re-run while supplying the path as an argument:
awx-manage export_custom_venv /path/to/venv
```

#### 7.1.2. 사용자 정의 가상 환경과 연결된 오브젝트 보기

**awx-manage** 명령을 사용하여 사용자 지정 가상 환경과 관련된 조직, 작업 및 인벤토리 소스를 확인합니다.

##### 절차

1. 자동화 컨트롤러 인스턴스에 SSH를 사용하여 다음을 실행합니다.

```
$ awx-manage custom_venv_associations /path/to/venv
```

연결된 오브젝트 목록이 나타납니다.

```
inventory_sources:
- id: 15
  name: celery
job_templates:
- id: 9
  name: Demo Job Template @ 2:40:47 PM
- id: 13
  name: elephant
organizations
```

```
- id: 3
  name: alternating_bongo_meow
- id: 1
  name: Default
  projects: []
```

### 7.1.3. 내보낼 사용자 정의 가상 환경 선택

**awx-manage export\_custom\_venv** 명령을 사용하여 내보낼 사용자 지정 가상 환경을 선택합니다.

#### 절차

1. 자동화 컨트롤러 인스턴스에 SSH를 사용하여 다음을 실행합니다.

```
$ awx-manage export_custom_venv /path/to/venv
```

이 명령의 출력은 지정된 가상 환경에 있는 항목의 **pip** 동결 을 표시합니다. 이 정보는 Ansible Builder가 새 자동화 실행 환경 이미지를 생성하는 데 사용할 **requirements.txt** 파일에 복사할 수 있습니다.

```
numpy==1.20.2
pandas==1.2.4
python-dateutil==2.8.1
pytz==2021.1
six==1.16.0
```

To list all available custom virtual environments run:  
awx-manage list\_custom\_venvs



#### 참고

출력을 줄이기 위해 **awx-manage list\_custom\_venvs** 를 실행할 때 **-q** 플래그를 전달합니다.

## 7.2. ANSIBLE CORE 버전 간 마이그레이션

Ansible Core 버전 간에 마이그레이션하려면 최신 버전에서 작동하도록 Ansible 인프라의 플레이북, 플러그인 및 기타 부분을 업데이트해야 합니다. 이 프로세스를 수행하려면 Ansible Core의 후속 버전에 대한 업데이트에 대해 변경 사항이 검증되어야 합니다. 이전 버전의 Ansible에서 Ansible-core 2.15로 마이그레이션하려는 경우 먼저 버전 뒤에 있는 Ansible 버전의 요구 사항을 충족하고 연속적인 업데이트를 2.15로 충족하는지 확인해야 합니다.

### 7.2.1. Ansible 포트 지정 가이드

Ansible *포트 지정 가이드* 는 연속 Ansible 버전 간에 동작 변경에 대한 정보를 제공하는 일련의 문서입니다. Ansible 버전에서 최신 버전으로 마이그레이션할 때 가이드를 참조하십시오.

### 7.2.2. 추가 리소스

- Ansible 2.8과 [Ansible 2.9](#) 간의 동작 변경은 Ansible 2.9를 참조하십시오.
- Ansible 2.9 및 [Ansible 2.10](#) 간의 동작 변경 사항은 Ansible 2.10을 참조하십시오.

## 8장. 자동화 콘텐츠로 콘텐츠 실행

자동화 실행 환경이 빌드되었으므로 자동화 콘텐츠 탐색기를 사용하여 자동화 컨트롤러에서 해당 콘텐츠를 실행하는 것과 동일한 방식으로 콘텐츠가 실행되는지 확인할 수 있습니다.

### 8.1. 자동화 콘텐츠로 ANSIBLE 플레이북 실행

콘텐츠 작성자는 자동화 콘텐츠로 Ansible 플레이북을 실행하고 각 플레이와 작업의 결과를 대화형으로 조사하여 플레이북을 확인하거나 문제를 해결할 수 있습니다. 실행 환경 내에서 및 실행 환경 없이 Ansible 플레이북을 실행하여 문제를 비교하고 해결할 수도 있습니다.

#### 8.1.1. 자동화 콘텐츠 탐색에서 플레이북 실행

자동화 콘텐츠 탐색기 텍스트 기반 사용자 인터페이스를 사용하여 Ansible 플레이북을 실행하여 작업 실행을 따르고 각 작업의 결과를 조사할 수 있습니다.

##### 사전 요구 사항

- 플레이북입니다.
- **localhost** 또는 인벤토리 플러그인을 사용하지 않는 경우 유효한 인벤토리 파일입니다.

##### 절차

1. 자동화 콘텐츠 시작

```
$ ansible-navigator
```

2. 플레이북을 실행합니다.

```
$ :run
```

3. 선택 사항: **ansible-navigator**를 입력하여 **simple-playbook.yml -i inventory.yml** 을 실행하여 플레이북을 실행합니다.
4. 인벤토리 및 기타 명령줄 매개 변수를 확인하거나 추가합니다.

```
INVENTORY OR PLAYBOOK NOT FOUND, PLEASE CONFIRM THE FOLLOWING
-----
Path to playbook: /home/ansible-navigator_demo/simple_playbook.yml
Inventory source: /home/ansible-navigator-demo/inventory.yml
Additional command line parameters: Please provide a value (optional)
-----
Submit Cancel
```

5. 탭을 클릭하여 제출하고 Enter를 누릅니다. 실행 중인 작업이 표시되어야 합니다.

PLAY NAME	OK	CHANGED	UNREACHABLE	FAILED	SKIPPED	IGNORED	IN PROGRESS	TASK COUNT	PROGRESS
all	6	0	0	6	0	0	0	12	COMPLETE

6. 플레이 옆에 있는 숫자를 입력하여 플레이 결과에 단계별로 입력하거나 9 위의 숫자에 대해 **<number>**를 입력합니다.

RESULT	HOST	NUMBER	CHANGED	TASK	TASK ACTION	DURATION
3 OK	node-0	3	False	Gathering Facts	gather_facts	1s
4 OK	node-1	4	False	Gathering Facts	gather_facts	1s
5 OK	node-2	5	False	Gathering Facts	gather_facts	1s
6 FAILED	main-0	6	False	Gather the package facts	ansible.builtin.package_facts	1s
7 FAILED	infra-0	7	False	Gather the package facts	ansible.builtin.package_facts	1s
8 FAILED	lb-0	8	False	Gather the package facts	ansible.builtin.package_facts	1s
9 FAILED	node-0	9	False	Gather the package facts	ansible.builtin.package_facts	1s
10 FAILED	node-1	10	False	Gather the package facts	ansible.builtin.package_facts	1s
11 FAILED	node-2	11	False	Gather the package facts	ansible.builtin.package_facts	0s

자동화 콘텐츠 탐색에 사용할 수 있는 색상이 있는 경우 실패한 작업이 빨간색으로 표시됩니다.

- 작업 결과를 검토하는 작업 옆에 숫자를 입력하거나 9 위의 숫자에 대해 `<number>`를 입력합니다.

```
PLAY [all:6] *****
TASK [Gather the package facts] *****
FAILED: [main-0] Could not detect a supported package manager from the following list: ['apt', 'apk', 'rpm', 'portage', 'pkg']
0 ---
1 duration: 1.339719
2 end: '2021-06-10T18:52:32.968770'
3 event_loop: null
4 host: main-0
5 ignore_errors: null
6 play: all
```

- 선택 사항: `type: doc` 은 문제 해결을 돕기 위해 작업에 사용된 모듈 또는 플러그인에 대한 문서를 표시합니다.

```
ANSIBLE.BUILTIN.PACKAGE_FACTS (MODULE)
0 ---
1 doc:
2 author:
3 - Matthew Jones (@matburt)
4 - Brian Coca (@bcoca)
5 - Adam Miller (@maxamillion)
6 collection: ansible.builtin
7 description:
8 - Return information about installed packages as facts.
<... output omitted ...>
11 module: package_facts
12 notes:
13 - Supports C(check_mode).
14 options:
15 manager:
16 choices:
17 - auto
18 - rpm
19 - apt
20 - portage
21 - pkg
22 - pacman
<... output truncated ...>
```

추가 리소스

- [ansible-playbook](#)
- [Ansible Playbook](#)

### 8.1.2. 자동화 콘텐츠 탐색 아티팩트 파일을 사용하여 플레이북 결과 검토

Automation Content navigator는 플레이북 실행 결과를 JSON 아티팩트 파일에 저장합니다. 이 파일을 사용하여 플레이북 결과를 다른 사용자와 공유하고, 보안 또는 규정 준수 이유로 저장하거나, 나중에 검토 및 문제를 해결할 수 있습니다. 플레이북 실행을 검토하려면 아티팩트 파일만 있으면 됩니다. 플레이북 자체 또는 인벤토리 액세스 권한에 액세스할 필요가 없습니다.

### 사전 요구 사항

- 플레이북 실행에서 자동화 콘텐츠를 탐색하는 아티팩트 JSON 파일입니다.

### 절차

- 아티팩트 파일을 사용하여 자동화 콘텐츠를 시작합니다.

```
$ ansible-navigator replay simple_playbook_artifact.json
```

1. 플레이북이 실행될 때 일치하는 플레이북 결과를 검토합니다.

PLAY NAME	OK	CHANGED	UNREACHABLE	FAILED	SKIPPED	IGNORED	IN PROGRESS	TASK COUNT	PROGRESS
all	12	0	0	0	25	0	0	37	COMPLETE

이제 플레이북을 실행한 후와 같이 결과를 검토하기 위해 플레이 및 작업 옆에 있는 숫자를 입력하여 결과를 검토할 수 있습니다.

### 추가 리소스

- [ansible-playbook](#)
- [Ansible Playbook](#)



## 9장. CONCLUSION

이제 특정 자동화 요구 사항에 맞는 자동화 실행 환경을 사용자 정의하고 컨테이너 레지스트리를 통해 공유 및 사용할 수 있습니다.