Red Hat Ansible Automation Platform 2.4

Red Hat Ansible Automation Platform 创建指南

了解如何使用 Ansible 创建自动化内容

Last Updated: 2024-07-03

Red Hat Ansible Automation Platform 2.4 Red Hat Ansible Automation Platform 创建指南

了解如何使用 Ansible 创建自动化内容

法律通告

摘要

本指南帮助开发人员了解如何使用 Ansible 为自动化创建内容。

目录

对红帽文档提供反馈

如果您对本文档有任何改进建议，或发现了任何错误，请通过 https://access.redhat.com 联系技术支持，以使用 **docs-product** 组件在 Ansible Automation Platform JIRA 项目中创建一个问题。

第 1 章 前言

使用自动化执行环境在 Red Hat Ansible Automation Platform 中自动化内容

您可以将执行环境用作可重复生成的、可移植、一致和可共享的容器镜像。它们以集合的形式从系统依赖项、Python 依赖项、Ansible 版本和 Ansible 内容控制 Ansible Automation Platform 作业运行时环境的所有依赖项。

第 2 章 内容创建器工作流和自动化执行环境简介

2.1. 关于内容工作流

在 Red Hat Ansible Automation Platform 2.0 之前，自动化内容开发人员可能需要多个 Python 虚拟环境来管理它们。为降低这种复杂性，Ansible Automation Platform 2.0 正在脱离虚拟环境并使用容器（称为自动化执行环境），因为它们易于构建和管理，并且更容易在团队和机构间共享。

随着自动化控制器使用自动化执行环境，自动化内容导航器和 Ansible Builder 等工具可确保您可以在您自己的开发系统中本地利用这些自动化执行环境。

其他资源

- 有关使用自动化内容导航器的更多信息，请参阅 Automation Content Navigator Creator 指南。
- 如需有关 Ansible Builder 的更多信息，请参阅 创建和恢复执行环境。

2.2. 架构概述

下表显示了 Ansible Automation Platform 2.0 上可用的工具的安排和使用，以及如何使用它们：

- 现在，Ansible Automation Platform 1.2 中使用自动内容导航程序(automation content navigator)
- 自动化内容导航器 + 下载的自动化执行环境 - 直接在笔记本电脑/工作站上使用
- 自动化内容导航器 + 下载的自动化执行环境 + 自动化控制器 - 用于本地推送/执行 → 远程
- 自动化内容导航器 + 自动化控制器 + Ansible Builder + 分层自定义 EE - 提供对已利用内容的更多控制，了解如何执行自动化作业

第 3 章 了解 ANSIBLE 概念

作为自动化开发人员，请在开始 Ansible 开发项目之前，回顾以下 Ansible 概念以创建成功的 Ansible playbook 和自动化执行环境。

3.1. 先决条件

- Ansible 已经安装。有关安装 Ansible 的详情，请参考 Ansible 文档中的安装 Ansible。

3.2. 关于 ANSIBLE PLAYBOOK

Playbook 是使用 YAML 编写的文件，其中包含特定组人类可读的指令或 "plays"，供您发送到在单个目标或目标组上运行。

Playbook 可用于管理远程机器的配置和部署，以及涉及滚动更新的多层部署顺序。使用 playbook 将操作委派给其他主机，并在此过程中与监控服务器和负载平衡器交互。编写完 playbook 后，您可以在企业内重复使用 playbook 实现自动化。

3.3. 关于 ANSIBLE 角色

角色是 Ansible 捆绑自动化内容的方法，以及利用已知文件结构自动加载相关变量、文件、任务、处理程序和其他工件。除了创建具有数百个任务的大型 playbook 外，您可以使用角色将任务划分为更小、分散的工作单元。

您可以查找用于调配基础架构、部署应用以及 Ansible Galaxy 上每天执行的所有任务的角色。根据 **Type** 过滤您的搜索并选择 **Role**。找到感兴趣的角色后，您可以使用 Ansible 捆绑的 **ansible-galaxy** 命令下载角色：

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

3.4. 关于内容集合

Ansible 内容集合是用于自动化的现成工具包。它包括几种类型的内容，如 playbook、角色、模块和插件。下图显示了集合的基本结构：

```
collection/
├── docs/
├── galaxy.yml
├── meta/
│   └── runtime.yml
├── plugins/
│   ├── modules/
│   │   └── module1.py
│   ├── inventory/
│   ├── lookup/
│   ├── filter/
│   └── .../
├── README.md
├── roles/
│   ├── role1/
│   ├── role2/
│   └── .../
├── playbooks/
```

```
│   ├── files/
│   ├── vars/
│   ├── templates/
│   ├── playbook1.yml
│   └── tasks/
└── tests/
    ├── integration/
    └── unit/
```

在 Red Hat Ansible Automation Platform 中，自动化中心充当 Ansible 认证内容集合的来源。

3.5. 关于执行环境

自动化执行环境是一致且可共享的容器镜像，充当 Ansible 控制节点。自动化执行环境减少了共享具有外部依赖项的 Ansible 内容的挑战。

自动化执行环境包括：

- Ansible 内核
- Ansible Runner
- Ansible Collections
- Python 库
- 系统依赖项
- 自定义用户需求

您可以使用 Ansible Builder 定义并创建自动化执行环境。

其他资源

- 如需有关 Ansible Builder 的更多信息，请参阅 创建和恢复执行环境。

第 4 章 工具和组件

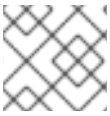
了解更多有关您在创建自动化执行环境中使用的红帽 Ansible 自动化平台工具和组件。

4.1. 关于 ANSIBLE BUILDER

Ansible Builder（Ansible 构建器）是一个命令行工具，它通过使用各种 Ansible Collections 中定义的元数据以及用户自动构建自动化执行环境的过程。

在开发 Ansible Builder 之前，Red Hat Ansible Automation Platform 用户可以在创建自定义虚拟环境或容器（包含所有必要的依赖项）时遇到依赖项问题和错误。

现在，使用 Ansible Builder 时，您可以轻松地创建一个可自定义的自动化执行环境定义文件，该文件指定了自动化执行环境中包含的内容，如集合、第三方 Python 要求和系统级别软件包。这可让您满足所有必要的要求和依赖项来获取作业运行。



注意

红帽目前不支持在构建自动化执行环境时选择提供自己的容器镜像的用户。

4.2. 使用自动化内容导航器

自动化内容导航器是一种命令行以内容创建者为导向的工具，具有基于文本的用户界面。您可以使用自动化内容导航器进行：

- 启动和观察作业和 playbook。
- 以 JSON 格式共享存储、完成的 playbook 和作业运行工件。
- 浏览和内省自动化执行环境.
- 浏览基于文件的清单。
- 呈现 Ansible 模块文档，并提取您可以在 playbook 中使用的示例。
- 查看用户界面的详细命令输出。

4.3. 关于自动化中心

Automation Hub 为红帽订阅的用户提供了一个途径，以便快速查找和使用红帽及我们的技术合作伙伴支持的内容，以便为最要求的环境提供额外的保证。

在高级别上，自动化中心为所有合作伙伴提供参与并提供经过认证的支持内容的概述。

从中央视图中，用户可以深入了解每个合作伙伴并签出集合。

此外，也可搜索所有可用集合的概述。

4.4. 关于 ANSIBLE 命令行界面

在命令行中使用 Ansible 是运行您不经常重复的任务的有用方法。处理重复任务的建议方法是编写 playbook。

命令行中 Ansible 的临时命令遵循以下结构：

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

4.5. 其他资源

- 有关如何将 Ansible 用作命令行工具的更多信息，请参阅 Ansible *用户指南*中的使用命令行工具。
- 要将内容上传到自动化中心，请参阅 Ansible Automation Platform 产品文档中的将内容 上传到自动化中心。

第 5 章 设置您的开发环境

您可以按照本节中的步骤设置开发环境，以创建自动化执行环境。

5.1. 安装 ANSIBLE BUILDER

先决条件

- 已安装 Podman 容器运行时。
- 您已在主机上附加了有效的订阅。这样，您可以访问安装 **ansible-builder** 所需的仅订阅资源，并确保自动启用 **ansible-builder** 所需的存储库。如需更多信息，请参阅附加 Red Hat Ansible Automation Platform 订阅。

流程

1. 在终端中，运行以下命令来激活 Ansible Automation Platform 仓库：

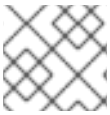
```
# dnf install --enablerepo=ansible-automation-platform-2.4-for-rhel-9-x86_64-rpms ansible-builder
```

5.2. 从 RPM 在 RHEL 上安装自动化内容导航器

您可以从 RPM 在 Red Hat Enterprise Linux (RHEL)上安装自动化内容导航程序。

先决条件

- 已安装 RHEL 8.6 或更高版本。
- 使用 Red Hat Subscription Manager 注册了您的系统。



注意

确保您只安装与当前 Red Hat Ansible Automation Platform 环境匹配的导航程序。

流程

1. 附加 Red Hat Ansible Automation Platform SKU。

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

2. 使用以下命令安装自动化内容导航器：
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
```

验证

- 验证您的自动化内容导航器安装：

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

以下示例演示了成功安装：

```
$ 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. 下载基本自动化执行环境

Ansible Automation Platform 2.0 附带的基础镜像托管在红帽生态系统目录(registry.redhat.io)上。

先决条件

- 您有有效的 Red Hat Ansible Automation Platform 订阅。

流程

1. 登录到 registry.redhat.io

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

2. 从 registry 中拉取基础镜像

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

第 6 章 创建内容

使用*创建器指南*一节中的指南，了解更多有关开发 Red Hat Ansible Automation Platform 中使用的内容的信息。

6.1. 创建 PLAYBOOK

Playbook 包含一个或多个 play。基本 play 包括以下部分：

- Name：对 playbook 整体功能的简短描述，有助于保持所有用户的可读性和组织性。
- Hosts：识别要针对 Ansible 运行的目标。
- Become 声明：这个可选声明可以被设置为 **true**/**yes** 来使用一个 become 插件 (如 **sudo**, **su**, **pfexec**, **doas**, **pbrun**, **dzdo**, **ksu**) 进行权限升级。
- Tasks：这是对 play 中每一主机执行的列表操作。

playbook 示例

```
- 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. 创建集合

您可以使用 Ansible Galaxy CLI 工具在本地创建自己的集合。您可以使用 **collection** 子命令激活特定于集合的命令。

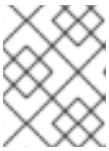
先决条件

- 开发环境中已安装了 Ansible 核版本 2.15 或更新版本。

流程

1. 在终端中，前往您希望命名空间根目录的位置。为简单起见，这应当是 COLLECTIONS_PATH 中的路径，但这不是必须的。
2. 运行以下命令，将 **my_namespace** 和 **my_collection_name** 替换为您自己的值：

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



注意

在 galaxy.ansible.com 或 console.redhat.com/ansible/automation-hub 上的 **My Content** 选项卡下检查，确保您有上传到命名空间的适当权限

前面的 命令将从命名空间参数（如果尚未存在）创建名为 的目录，然后在 下创建一个具有集合名称的目录。该目录内为默认值或"框架"集合。在这里，您可以添加角色或插件，并开始开发自己的集合。

对于执行环境，集合开发人员可以通过在 Ansible Builder 中提供适当的元数据来声明其内容的要求。

集合的要求可以通过以下方法识别：

- 文件 **meta/execution-environment.yml**，它引用 Python 或 **bindep** 要求文件。
- 名为 **requirements.txt** 的文件，其中包含有关 Python 依赖项的信息，有时可在集合的根目录中找到。
- 名为 **bindep.txt** 的文件，其中包含系统级依赖项，有时可在集合的根目录中找到。
- 如果其中任何文件位于 Collection 的 **build_ignore** 中，Ansible Builder 不会在这些文件中获取。**build_ignore** 部分过滤不应包含在构建工件中的任何文件或目录。

集合维护者可以使用 introspection 命令验证 ansible-builder 是否识别他们期望的要求：

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

其他资源

- **有关创建集合的更多信息，请参阅 Ansible *开发人员 指南中的创建 集合*。**

6.3. 创建角色

您可以使用 Ansible Galaxy CLI 工具创建角色。您可以从 roles 子命令访问特定于角色的命令。

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

仍然支持集合之外的独立角色，但在集合内创建新的角色，以利用 Ansible Automation Platform 必须提供的所有功能。

流程

1. **在终端中，前往集合内的 roles 目录。**

2. **在集合中创建一个名为 role_name 的角色：**

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

该集合现在在 roles 目录中包含名为 my_role 的角色：

```
~/.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. **可以使用 --role-skeleton 参数提供自定义角色框架目录。这使得组织能够为新角色创建标准化模板，以满足其需求。**

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

这将通过将 ~/role_skeleton 的内容复制到 my _role 来创建名为 my_role 的角色。role_skeleton 的内容可以是在角色目录中有效的任何文件或文件夹。

其他资源

- **有关创建角色的更多信息，请参阅 Ansible Galaxy 文档中的创建角色。**

6.4. 创建自动化执行环境

自动化执行环境定义文件将指定

- Ansible 版本
- Python 版本（默认为系统 Python）
- 组所需的 Python 库
- 零个或多个内容集合（可选）
- 这些特定集合的 Python 依赖项

为环境指定一组集合的概念是解析并安装其依赖项。不需要将集合本身安装到您要在其上生成自动化执行环境的机器上。

从此定义构建自动化执行环境，生成容器镜像。请阅读 Ansible Builder 文档，以了解创建这些镜像涉及的步骤。

第 7 章 迁移现有内容

以下小节了解如何在升级到 Red Hat Ansible Automation Platform 2.0 和自动化控制器 4.0 后，使用 awx-manage 命令协助迁移过程中的其他步骤。此外，了解更多关于在 Ansible 版本间迁移的信息。

7.1. 将虚拟环境迁移到自动化执行环境

升级到 Red Hat Ansible Automation Platform 2.0 和自动化控制器 4.0 后，请使用以下部分协助迁移过程中的其他步骤。

7.1.1. 列出自定义虚拟环境

您可以使用 awx-manage 命令列出自动化控制器实例上的虚拟环境。

流程

1. SSH 到自动化控制器实例并运行：

```
$ awx-manage list_custom_venvs
```

这时将显示已发现的虚拟环境列表。

```
# 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. 查看与自定义虚拟环境关联的对象

使用 awx-manage 命令，查看与自定义虚拟环境关联的组织、作业和清单源。

流程

1. SSH 到自动化控制器实例并运行：

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

这时将显示相关对象的列表。

```
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. 选择要导出的自定义虚拟环境

选择您要使用 **awx-manage export_custom_venv** 命令导出的自定义虚拟环境。

流程

1. **SSH 到自动化控制器实例并运行：**

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

此命令的输出将显示在指定虚拟环境中的 **pip freeze** 状态。此信息可复制到 **Ansible Builder** 的 **requirements.txt** 文件中，用于创建新的自动化执行环境镜像

```
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
```

注意

在运行 **awx-manage list_custom_venvs** 时传递 **-q** 标志来减少输出。

7.2. 在 ANSIBLE 内核版本间迁移

在 Ansible 核心版本之间迁移需要您更新 playbook、插件和 Ansible 基础架构的其他部分，以确保它们使用最新版本。此过程要求根据对每个连续版本的 Ansible Core 所做的更新来验证更改。如果您要从 Ansible 的早期版本迁移到 Ansible-core 2.15，您首先需要验证是否满足遵循您的版本的 Ansible 版本的要求，并从那里对 2.15 进行连续的更新。

7.2.1. Ansible 移植指南

Ansible *移植指南*是一系列文档，提供关于连续 Ansible 版本之间行为更改的信息。从 Ansible 版本迁移到较新版本时，请参考指南。

7.2.2. 其他资源

- 如需 Ansible 2.8 和 Ansible 2.9 之间的行为更改，请参阅 Ansible 2.9。
- 如需 Ansible 2.9 和 Ansible 2.10 之间的行为更改，请参阅 Ansible 2.10。

第 8 章 使用自动化内容导航器执行您的内容

现在，您已构建了自动化执行环境，您可以使用自动化内容导航器验证内容是否以与自动化控制器运行相同的方式运行。

8.1. 使用自动化内容导航器运行 ANSIBLE PLAYBOOK

作为内容创建者，您可以使用自动化内容导航器并以交互方式执行 Ansible playbook，以交互方式进入每个 play 的结果，以及验证或排除 playbook 的任务。您还可以在执行环境中执行 Ansible playbook，且无需执行环境，以比较和排除任何问题。

8.1.1. 从自动化内容导航器执行 playbook

您可以使用自动化内容导航器基于文本的用户界面运行 Ansible playbook，以遵循任务的执行，并转至每个任务的结果。

先决条件

- 一个 playbook。
- 有效的清单文件（如果没有使用 localhost）或清单插件。

流程

1. 启动自动化内容导航器

```
$ ansible-navigator
```

2. 运行 playbook。

```
$ :run
```

3. 可选：键入 ansible-navigator run simple-playbook.yml -i inventory.yml 以运行 playbook。

4. 验证或添加清单以及任何其他命令行参数。

```
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. 点 Submit 并按回车。您应该会看到任务正在执行。

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

6. 输入 play 旁边的数字以进入 play 结果，或者如果大于 9， 键入 :<number>。

```
   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
```

如果您为自动化内容导航器启用了颜色，则失败的任务以红色显示。

7. 键入要查看任务结果的任务旁边的数字，或者如果大于 9，键入 :<number>。

```
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
```

8. 可选： type:doc 调出任务中使用的模块或插件文档，以帮助进行故障排除。

```
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 ...>
```

其他资源

- **ansible-playbook**
- **Ansible playbook**

8.1.2. 查看 playbook 结果，其中包含自动化内容导航器工件文件

自动化内容导航器将 playbook 运行的结果保存到 JSON 构件文件中。您可以使用此文件与其他人共享 playbook 结果，出于安全或合规性的原因将其保存，或者稍后进行检查和故障排除。您只需要构件文件即可查看 playbook 运行。您不需要访问 playbook 本身或清单访问权限。

先决条件

- 来自 playbook 运行的自动化内容导航器工件 JSON 文件。

流程

- 使用工件文件启动自动化内容导航器。

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

1. 检查 playbook 运行时的 playbook 结果。

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

现在，您可以键入 play 和任务旁边的数字，以逐一检查结果，如执行 playbook 后一样。

其他资源

- ansible-playbook
- Ansible playbook

第 9 章 总结

现在，您应能够根据特定的自动化需求自定义自动化执行环境，并通过容器 registry 共享和使用它们。