



Red Hat Ansible Automation Platform 2.4

自动化内容导航器创建者指南

开发与 Ansible Automation Platform 兼容的内容

开发与 Ansible Automation Platform 兼容的内容

法律通告

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.

摘要

本指南演示了如何使用自动化内容导航器开发与 Ansible Automation Platform 兼容的 Ansible playbook、集合和角色。

目录

对红帽文档提供反馈	3
第 1 章 自动化内容导航器简介	4
1.1. 使用自动化内容导航器	4
1.2. 自动化内容导航器模式	4
1.3. 自动化内容导航器命令	5
1.4. ANSIBLE 和自动化内容导航器命令之间的关系	6
第 2 章 在 RHEL 上安装自动化内容导航器	7
2.1. 从 RPM 在 RHEL 上安装自动化内容导航器	7
第 3 章 使用自动化内容导航器查看自动化执行环境	9
3.1. 从自动化内容导航器查看自动化执行环境	9
第 4 章 查看带有自动化内容导航器的清单	10
4.1. 查看自动化内容导航器中的清单	10
第 5 章 使用自动化内容导航器浏览集合	12
5.1. 自动化内容导航程序集合显示	12
5.2. 从自动化内容导航器浏览集合	12
5.3. 查看自动化内容导航器中的文档	14
第 6 章 使用自动化内容导航器运行 ANSIBLE PLAYBOOK	17
6.1. 从自动化内容导航器执行 PLAYBOOK	17
6.2. 查看 PLAYBOOK 结果，其中包含自动化内容导航器工件文件	18
第 7 章 使用自动化内容导航器查看 ANSIBLE 配置	20
7.1. 从自动化内容导航器查看 ANSIBLE 配置	20
第 8 章 自动化内容导航器配置设置	22
8.1. 创建自动化内容导航器设置文件	22
8.2. 自动化内容导航器常规设置	23
8.3. 自动化内容导航器配置子命令设置	31
8.4. 自动化内容导航器 DOC 子命令设置	32
8.5. 自动化内容导航器 INVENTORY 子命令设置	33
8.6. 自动化内容导航器 REPLAY 子命令设置	34
8.7. 自动化内容导航器 RUN 子命令设置	35
第 9 章 使用自动化内容导航器对 ANSIBLE 内容进行故障排除	38
9.1. 查看 PLAYBOOK 结果，其中包含自动化内容导航器工件文件	38
9.2. 有关自动化内容导航器的常见问题	38

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 自动化内容导航器简介

作为内容创建者，您可以使用自动化内容导航器开发与 Red Hat Ansible Automation Platform 兼容的 Ansible playbook、集合和角色。您可以在以下环境中使用自动化内容导航器，并在所有这些环境中无缝且可预测的结果：

- 本地开发机器
- 自动执行环境

自动化内容导航器还会生成可用于帮助您开发 playbook 和对问题区域进行故障排除的工件文件。

1.1. 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。

1.2. 自动化内容导航器模式

自动化内容导航器以两种模式运行：

stdout 模式

在命令行中接受大多数现有的 Ansible 命令和扩展。

基于文本的用户界面模式

为 Ansible 命令提供基于文本的交互式界面。使用工件文件运行 playbook 后，使用此模式评估内容、运行 playbook 并对它们进行故障排除。

1.2.1. stdout 模式

使用带有自动化内容导航器的 **-m stdout** 子命令在自动化执行环境或本地开发环境中使用熟悉的 Ansible 命令，如 **ansible-playbook**。您可以使用您熟悉的命令来执行快速任务。

在此模式中，自动化内容导航器还提供大量帮助：

--help

可从 **ansible-navigator** 命令或任何子命令访问，如 **ansible-navigator config --help**。

子命令帮助

可从子命令访问，如 **ansible-navigator config --help-config**。此帮助显示相关 Ansible 命令支持的所有参数的详细信息。

1.2.2. 基于文本的用户界面模式

基于文本的用户界面模式增强了与自动化执行环境、集合、playbook 和清单的交互。此模式兼容集成开发环境 (IDE)，如 Visual Studio Code。

```

0 ## Welcome
1 -----
2
3 Some things you can try from here:
4 - `:collections`           Explore available collections
5 - `:config`               Explore the current ansible configuration
6 - `:doc <plugin>`        Review documentation for a module or plugin
7 - `:help`                 Show the main help page
8 - `:images`              Explore execution environment images
9 - `:inventory -i <inventory>` Explore an inventory
10 - `:log`                  Review the application log
11 - `:open`                 Open current page in the editor
12 - `:replay`              Explore a previous run using a playbook artifact
13 - `:run <playbook> -i <inventory>` Run a playbook in interactive mode
14 - `:quit`                 Quit the application
15
16 happy automating,
17
18 -winston

```

^f/PgUp page up ^b/PgDn page down ↵ scroll esc back :help help

这个模式包括很多有用的用户界面选项：

冒号命令

您可以使用冒号访问所有自动化内容导航器命令，如 `:run` 或 `:collections`

导航基于文本的界面

屏幕中显示了如何向上或向下滚动页面、滚动到之前的屏幕或 `access:help`。

按行号输出

您可以在输出中的任何行编号前面加上冒号，例如 `:12`。

颜色编码的输出

启用颜色后，自动化内容导航器以红色显示项目，如已弃用的模块。

分页和滚动

您可以使用每个自动化内容导航器屏幕底部显示的选项来向上或向下滚动、滚动或转义。

您无法在自动化内容导航器运行后在模式间切换。

本文档对大多数步骤都使用基于文本的用户界面模式。

1.3. 自动化内容导航器命令

自动化内容导航器命令以 `-m stdout` 模式运行熟悉的 Ansible CLI 命令。您可以在相关的 Ansible CLI 命令中使用所有子命令和选项。详情请参阅 `ansible-navigator --help`。

表 1.1. 自动化内容导航器命令

命令	描述	CLI 示例
collections	探索可用的集合	<code>ansible-navigator collections --help</code>
config	探索当前的 Ansible 配置	<code>ansible-navigator config --help</code>

命令	描述	CLI 示例
doc	查看模块或插件的文档	ansible-navigator doc --help
images	探索执行环境镜像	ansible-navigator images --help
清单 (inventory)	探索清单	ansible-navigator inventory -help
replay	使用 playbook 工件浏览之前的运行	ansible-navigator replay --help
run	运行 playbook	ansible-navigator run --help
welcome	从欢迎页面开始	ansible-navigator welcome --help

1.4. ANSIBLE 和自动化内容导航器命令之间的关系

自动化内容导航器命令以 **-m stdout** 模式运行熟悉的 Ansible CLI 命令。您可以使用相关 Ansible CLI 命令中提供的所有子命令和选项。详情请参阅 **ansible-navigator --help**。

表 1.2. 自动化内容导航器和 Ansible CLI 命令的比较

自动化内容导航器命令	Ansible CLI 命令
ansible-navigator collections	ansible-galaxy collection
ansible-navigator config	ansible-config
ansible-navigator doc	ansible-doc
ansible-navigator inventory	ansible-inventory
ansible-navigator run	ansible-playbook

第 2 章 在 RHEL 上安装自动化内容导航器

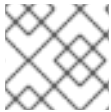
作为内容创建者，您可以在 Red Hat Enterprise Linux (RHEL) 8.6 或更高版本上安装自动化内容导航程序。

2.1. 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
```

第 3 章 使用自动化内容导航器查看自动化执行环境

作为内容开发人员，您可以使用自动化内容导航器查看自动化执行环境，并显示自动化执行环境中包含的软件包和集合。自动化内容导航器运行 `playbook` 来提取和显示结果。

3.1. 从自动化内容导航器查看自动化执行环境

您可以使用自动化内容导航器基于文本的用户界面查看自动化执行环境。

先决条件

- 自动执行环境

流程

1. 查看自动化内容导航器配置中包含的自动化执行环境。

```
$ ansible-navigator images
```

NAME	TAG	EXECUTION ENVIRONMENT	CREATED	SIZE
0 ansible-automation-platform-20-ee-minimal-rhel8	latest	True	4 weeks ago	411 MB
1 ansible-automation-platform-20-ee-supported-rhel8 (primary)	latest	True	45 hours ago	923 MB
2 ansible-runner	devel	True	4 weeks ago	652 MB
3 ccutil	amazing	False	16 months ago	1.67 GB

2. 如需更多详细信息，请键入您要提取的自动化执行环境的数量。

ANSIBLE-AUTOMATION-PLATFORM-20-EE-SUPPORTED-RHEL8:LATEST (PRIMARY)	DESCRIPTION
0 Image information	Information collected from image inspection
1 General information	OS and python version information
2 Ansible version and collections	Information about ansible and ansible collections
3 Python packages	Information about python and python packages
4 Operating system packages	Information about operating system packages
5 Everything	All image information

您可以检查每个安装的自动化执行环境的软件包和版本，以及任何包含的集合的 Ansible 版本。

3. 可选：传递您要使用的自动化执行环境。这将是自动化内容导航器使用的自动化执行环境。

```
$ ansible-navigator images --eei registry.example.com/example-enterprise-ee:latest
```

验证

- 查看自动化执行环境输出。

ANSIBLE-AUTOMATION-PLATFORM-20-EE-SUPPORTED-RHEL8:LATEST (PRIMARY)	DESCRIPTION
0 Image information	Information collected from image inspection
1 General information	OS and python version information
2 Ansible version and collections	Information about ansible and ansible collections
3 Python packages	Information about python and python packages
4 Operating system packages	Information about operating system packages
5 Everything	All image information

第 4 章 查看带有自动化内容导航器的清单

作为内容创建者，您可以使用自动化内容浏览器来查看 Ansible 清单，并以交互方式签入到组和主机。

4.1. 查看自动化内容导航器中的清单

您可以以交互模式通过自动化内容导航器用户界面查看 Ansible 清单，并将 delve 放入组和主机中以了解更多详细信息。

先决条件

- 有效的清单文件或清单插件。

流程

- 启动自动化内容导航器。

```
$ ansible-navigator
```

可选：从命令行键入 **ansible-navigator inventory -i simple_inventory.yml** 来查看清单。

- 检查清单。

```
:inventory -i simple_inventory.yml

TITLE      DESCRIPTION
0| Browse groups  Explore each inventory group and group members members
1| Browse hosts   Explore the inventory with a list of all hosts
```

- 键入 **0** 以浏览组。

```
NAME          TAXONOMY          TYPE
0| general     all               group
1| nodes      all               group
2| ungrouped   all               group
```

TAXONOMY 字段详细说明所选组或节点所属组的层次结构。

- 键入与您要转至的组对应的数字。

```
NAME          TAXONOMY          TYPE
0| node-0     all▶nodes        host
1| node-1     all▶nodes        host
2| node-2     all▶nodes        host
```

- 输入与您要提取的主机对应的数字，或者如果大于 9，键入 **:<number>**。

```
[node-1]
0| ---
1| ansible_host: node-1.example.com
2| inventory_hostname: node-1
```

验证

- 检查清单输出。

	TITLE	DESCRIPTION
0	Browse groups	Explore each inventory group and group members members
1	Browse hosts	Explore the inventory with a list of all hosts

其他资源

- [ansible-inventory](#).
- [如何构建您的清单](#)。

第 5 章 使用自动化内容导航器浏览集合

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浏览 Ansible 集合，并以交互方式浏览在本地或自动化执行环境中开发的每个集合。

5.1. 自动化内容导航程序集合显示

自动化内容导航器显示有关您的集合的信息，每个集合的详情如下：

SHADOWED

表示在搜索顺序中，集合的额外副本较高，playbook 首选使用该集合。

TYPE

显示集合是否包含在作为 **bind_mount** 的自动化执行环境或挂载到自动化执行环境中的卷中。

PATH

根据集合 TYPE 字段，反映自动化执行环境或本地文件系统中的集合位置。

NAME	VERSION	SHADOWED	TYPE	PATH
0 amazon.aws	1.5.0	False	contained	/usr/share/ansible/collections/ansible_collections/amazon/aws/
1 ansible.posix	1.2.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/posix/
2 ansible.utils	2.2.0	False	bind_mount	/home/samccann/aap/collections/ansible_collections/ansible/utils/
3 ansible.windows	1.6.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/windows/
4 awx.awx	19.2.2	False	bind_mount	/home/samccann/aap/collections/ansible_collections/awx/awx/
5 awx.awx	19.2.0	True	contained	/usr/share/ansible/collections/ansible_collections/awx/awx/
6 azure.azcollection	1.7.0	False	contained	/usr/share/ansible/collections/ansible_collections/azure/azcollection/

5.2. 从自动化内容导航器浏览集合

您可以使用交互模式的自动化内容导航器用户界面浏览 Ansible 集合，并转至每个集合。自动化内容导航器显示当前项目目录中的集合，以及自动化执行环境中可用的集合

先决条件

- 本地可访问的集合或安装的自动化执行环境。

流程

1. 启动自动化内容导航器

```
$ ansible-navigator
```

2. 浏览该集合。或者，您也可以键入 **ansible-navigator collections** 来直接浏览集合。

```
$ :collections
```

NAME	VERSION	SHADOWED	TYPE	PATH
0 amazon.aws	1.4.1	False	contained	/usr/share/ansible/collections/ansible_collections/amazon/aws/
1 ansible.netcommon	2.1.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/netcommon/
2 ansible.posix	1.2.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/posix/
3 ansible.tower	3.8.3	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/tower/
4 ansible.utils	2.2.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/utils/
5 ansible.windows	1.5.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/windows/
6 arista.eos	2.1.2	False	contained	/usr/share/ansible/collections/ansible_collections/arista/eos/

3. 键入您要浏览的集合数。

```
:4
```


	ANSIBLE_UTILS	TYPE	ADDED	DEPRECATED	DESCRIPTION
0	cli_parse	module	1.0.0	False	Parse cli output or text using a variety of parsers
1	fact_diff	module	1.0.0	False	Find the difference between currently set facts
2	from_xml	filter	2.0.2	False	Convert given XML string to native python dictionary.
3	get_path	lookup	1.0.0	False	Retrieve the value in a variable using a path
4	get_path	filter	1.0.0	False	Retrieve the value in a variable using a path
5	in_any_network	test	2.2.0	False	Test if an IP or network falls in any network
6	in_network	test	2.2.0	False	Test if IP address falls in the network

4. 键入与您要转至的模块对应的数字。

```

ANSIBLE_UTILS.IP_ADDRESS: Test if something in an IP address
0 | ---
1 | additional_information: {}
2 | collection_info:
3 |   authors:
4 |     - Ansible Community
5 |   dependencies: {}
6 |   description: Ansible Collection with utilities to ease the management, manipulation,
7 |     and validation of data within a playbook
8 |   documentation: null
9 |   homepage: null
10 |  issues: null
11 |  license: []
12 |  license_file: LICENSE
13 |  name: ansible.utils
14 |  namespace: ansible
15 |  path: /usr/share/ansible/collections/ansible_collections/ansible/utils/
16 |  readme: README.md
<... output truncated...>

```

5. 可选：跳至此模块的文档示例。

```

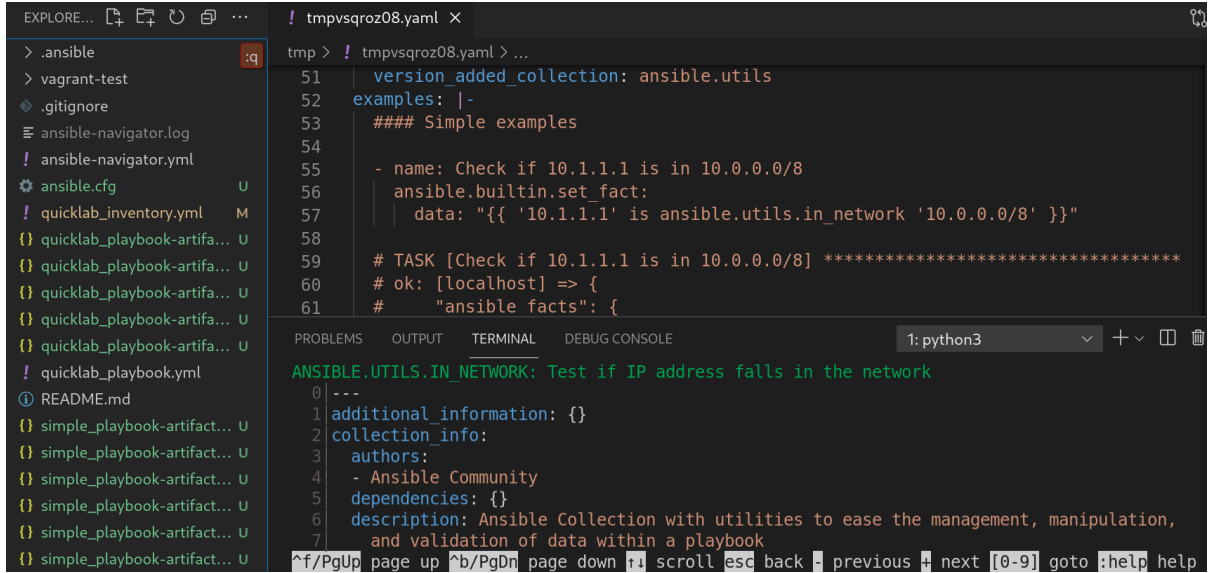
:{{ examples }}

0 |
1 |
2 | ##### Simple examples
3 |
4 | - name: Check if 10.1.1.1 is a valid IP address
5 |   ansible.builtin.set_fact:
6 |     data: "{{ '10.1.1.1' is ansible.utils.ip_address }}"
7 |
8 | # TASK [Check if 10.1.1.1 is a valid IP address] *****
9 | # ok: [localhost] => {
10 | #   "ansible_facts": {
11 | #     "data": true
12 | #   },
13 | #   "changed": false
14 | # }
15 |

```

6. 可选：打开编辑器中的示例，将其复制到 playbook 中。

```
:open
```



验证

- 浏览集合列表。

	NAME	VERSION	SHADOWED	TYPE	PATH
0	amazon.aws	1.4.1	False	contained	/usr/share/ansible/collections/ansible_collections/amazon/aws/
1	ansible.netcommon	2.1.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/netcommon/
2	ansible.posix	1.2.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/posix/
3	ansible.tower	3.8.3	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/tower/
4	ansible.utils	2.2.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/utils/
5	ansible.windows	1.5.0	False	contained	/usr/share/ansible/collections/ansible_collections/ansible/windows/
6	arista.eos	2.1.2	False	contained	/usr/share/ansible/collections/ansible_collections/arista/eos/

其他资源

- [ansible-collection](#)。
- [Ansible 清单简介](#)。

5.3. 查看自动化内容导航器中的文档

您可以在交互模式中查看带有自动化内容导航器文本用户界面的集合和插件的 Ansible 文档。自动化内容导航器显示当前项目目录中的集合，以及自动化执行环境中可用的集合

先决条件

- 本地可访问的集合或安装的自动化执行环境。

流程

1. 启动自动化内容导航器

```
$ ansible-navigator
```

2. 查看您感兴趣的模块。或者，您也可以键入 **ansible-navigator doc** 来访问文档。

```
:doc ansible.utils.ip_address
```

```
ANSIBLE UTILS.IP_ADDRESS: Test if something in an IP address
0|---
```

```

1 | additional_information: {}
2 | collection_info:
3 |   authors:
4 |     - Ansible Community
5 |   dependencies: {}
6 |   description: Ansible Collection with utilities to ease the management, manipulation,
7 |     and validation of data within a playbook
8 |   documentation: null
9 |   homepage: null
10 |  issues: null
11 |  license: []
12 |  license_file: LICENSE
13 |  name: ansible.utils
14 |  namespace: ansible
15 |  path:/usr/share/ansible/collections/ansible_collections/ansible/utils/
16 |  readme: README.md
<... output truncated...>

```

3. 跳转到此模块的文档示例。

```

:{{ examples }}
0 |
1 |
2 | ##### Simple examples
3 |
4 | - name: Check if 10.1.1.1 is a valid IP address
5 |   ansible.builtin.set_fact:
6 |     data: "{{ '10.1.1.1' is ansible.utils.ip_address }}"
7 |
8 | # TASK [Check if 10.1.1.1 is a valid IP address] *****
9 | # ok: [localhost] => {
10 | #   "ansible_facts": {
11 | #     "data": true
12 | #   },
13 | #   "changed": false
14 | # }
15 |

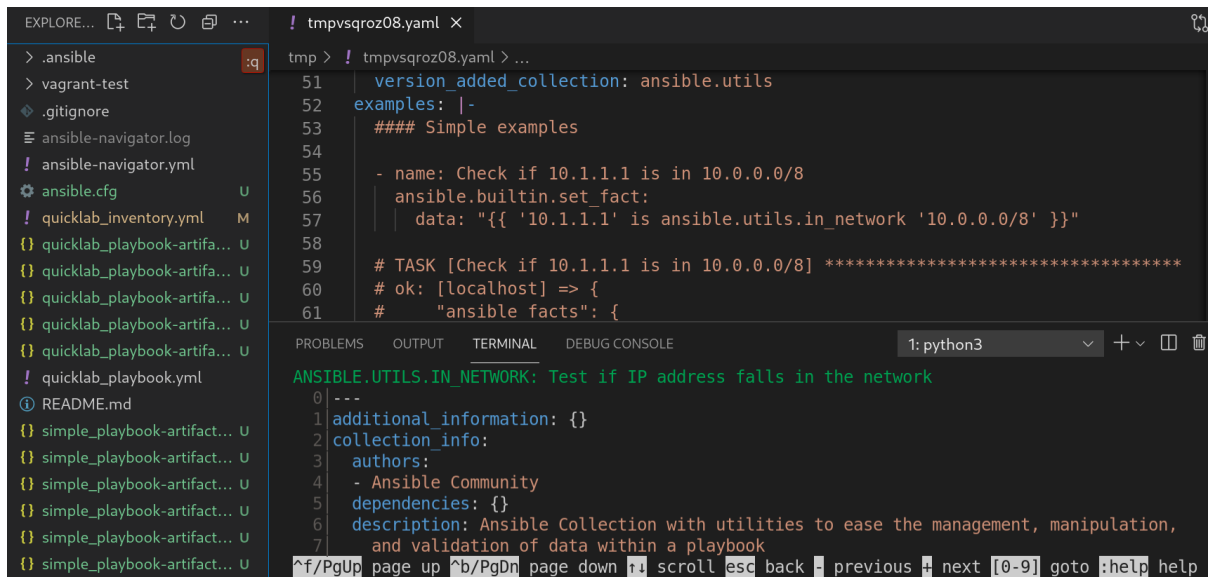
```

4. 可选：打开编辑器中的示例，将其复制到 playbook 中。

```

:open

```



The screenshot shows the Ansible Navigator interface. On the left is a file explorer with a tree view containing files like `.ansible`, `vagrant-test`, `.gitignore`, `ansible-navigator.log`, `ansible-navigator.yml`, `ansible.cfg`, `quicklab_inventory.yml`, and several `quicklab_playbook-artifa...` files. The main editor displays the content of `tmpvsqroz08.yml`, which includes a `version_added` field, an `examples` list, and a task named "Check if 10.1.1.1 is in 10.0.0.0/8". Below the editor is a terminal window showing the output of the task: `ANSIBLE_UTILS.IN_NETWORK: Test if IP address falls in the network` followed by a JSON object containing `additional_information`, `collection_info`, `authors`, `dependencies`, and `description`.

有关如何设置编辑器的详情，请参阅[自动化内容导航器常规设置](#)。

其他资源

- [集合索引](#)
- [使用 Ansible 集合](#)
- [构建 Ansible 清单](#)

第 6 章 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。

6.1. 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](#)

6.2. 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 7 章 使用自动化内容导航器查看 ANSIBLE 配置

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查看 Ansible 配置到设置中。

7.1. 从自动化内容导航器查看 ANSIBLE 配置

您可以使用交互模式的自动化内容导航器用户界面查看 Ansible 配置，并转至设置中。自动化内容导航器从可访问的 Ansible 配置文件中提取结果，或者返回默认值（如果没有配置文件）。

先决条件

- 如果您需要访问其他自动化执行环境，您需要向 Red Hat registry 进行身份验证。详情请参阅 [Red Hat Container Registry 身份验证](#)。

流程

1. 启动自动化内容导航器

```
$ ansible-navigator
```

可选：从命令行键入 **ansible-navigator config** 来访问 Ansible 配置设置。

2. 检查 Ansible 配置。

```
:config
```

OPTION	DEFAULT	SOURCE	VIA	CURRENT VALUE
0 ACTION_WARNINGS	True	default	default	True
1 AGNOSTIC_BECOME_PROMPT	True	default	default	True
2 ALLOW_WORLD_READABLE_TMPFI	True	default	default	False
3 ANSIBLE_CONNECTION_PATH	True	default	default	None
4 ANSIBLE_COW_ACCEPTLIST	False	/home/samccann/ansible-nav/home/samccann/ansible-nav	['bud-frogs', 'bunny', 'cheese']	None
5 ANSIBLE_COW_PATH	True	default	default	default
6 ANSIBLE_COW_SELECTION	True	default	default	default
7 ANSIBLE_FORCE_COLOR	True	default	default	False

某些值反映了自动化执行环境正常工作所需的自动化执行环境中进行的设置。这些设置显示为您在 Ansible 配置文件中无法设置的非默认设置。

3. 输入与您要提取的设置对应的数字，或者如果大于 9，键入 **:<number>**。

```
ANSIBLE COW ACCEPTLIST (current: ['bud-frogs', 'bunny', 'cheese']) (default:
0 | ---
1 | current:
2 | - bud-frogs
3 | - bunny
4 | - cheese
5 | default:
6 | - bud-frogs
7 | - bunny
8 | - cheese
9 | - daemon
```

输出显示了当前设置以及默认设置。请注意，本例中的 **source** 是 **env**，因为设置来自自动化执行环境。

验证

- 检查配置输出。

```
OPTION          DEFAULT SOURCE      VIA                CURRENT VALUE
0 ACTION_WARNINGS      True default       default           True
1 AGNOSTIC_BECOME_PROMPT True default       default           True
2 ALLOW_WORLD_READABLE_TMPFI True default       default           False
3 ANSIBLE_CONNECTION_PATH True default       default           None
4 ANSIBLE_COW_ACCEPTLIST False /home/samccann/ansible-nav/home/samccann/ansible-nav['bud-frogs', 'bunny',
5 ANSIBLE_COW_PATH     True default       default           None
6 ANSIBLE_COW_SELECTION True default       default           default
7 ANSIBLE_FORCE_COLOR  True default       default           False
```

其他资源

- [ansible-config](#)。
- [Ansible 配置简介](#)。

第 8 章 自动化内容导航器配置设置

作为内容创建者，您可以配置自动化内容导航器以适合您的开发环境。

8.1. 创建自动化内容导航器设置文件

您可以通过以下方法更改默认自动化内容导航器设置：

- 命令行
- 在一个设置文件中
- 作为一个环境变量

自动化内容导航器按以下顺序检查设置文件，并使用第一个匹配项：

- **ANSIBLE_NAVIGATOR_CONFIG** - 设置文件路径环境变量（如果已设置）。
- **./ansible-navigator.<ext>** - 当前项目目录中的设置文件，文件名中没有点。
- **~/ansible-navigator.<ext>** - 您的主目录，文件名中有一个点。

在创建自动化内容导航器设置文件时请考虑以下几点：

- 设置文件可以是 **JSON** 或 **YAML** 格式。
- 对于 **JSON** 格式的设置，扩展名必须为 **.json**。
- 对于 **YAML** 格式的设置，扩展名必须为 **.yml** 或 **.yaml**。
- 项目和主目录分别只能包含一个设置文件。
- 如果自动化内容导航器在任一目录中找到多个设置文件，则会导致错误。

您可以将以下示例设置文件复制到其中一种路径中，以启动 **ansible-navigator** 设置文件。

```
---
ansible-navigator:
# ansible:
# config: /tmp/ansible.cfg
# cmdline: "--forks 15"
# inventories:
# - /tmp/test_inventory.yml
# playbook: /tmp/test_playbook.yml
# ansible-runner:
# artifact-dir: /tmp/test1
# rotate-artifacts-count: 10
# timeout: 300
# app: run
# collection-doc-cache-path: /tmp/cache.db
# color:
# enable: False
# osc4: False
# editor:
# command: vim_from_setting
# console: False
```

```

# documentation:
# plugin:
#   name: shell
#   type: become
# execution-environment:
#   container-engine: podman
#   enabled: False
#   environment-variables:
#     pass:
#       - ONE
#       - TWO
#       - THREE
#     set:
#       KEY1: VALUE1
#       KEY2: VALUE2
#       KEY3: VALUE3
# image: test_image:latest
# pull-policy: never
# volume-mounts:
#   - src: "/test1"
#     dest: "/test1"
#     label: "Z"
# help-config: True
# help-doc: True
# help-inventory: True
# help-playbook: False
# inventory-columns:
#   - ansible_network_os
#   - ansible_network_cli_ssh_type
#   - ansible_connection
  logging:
#   append: False
  level: critical
#   file: /tmp/log.txt
#   mode: stdout
#   playbook-artifact:
#     enable: True
#     replay: /tmp/test_artifact.json
#     save-as: /tmp/test_artifact.json

```

8.2. 自动化内容导航器常规设置

下表描述了自动化内容导航器的每个常规参数和设置选项。

表 8.1. 自动化内容导航器常规参数设置

参数	描述	设置选项
----	----	------

参数	描述	设置选项
ansible-runner-artifact-dir	存储由 ansible-runner 生成工件的目录路径。	<p>Default:没有设置默认值</p> <p>CLI: <code>--rad</code> 或 <code>--ansible-runner-artifact-dir</code></p> <p>ENV: <code>ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_ARTIFACT_DIR</code></p> <p>Settings file:</p> <pre>ansible-navigator: ansible-runner: artifact-dir:</pre>
ansible-runner-rotate-artifacts-count	为最后 n 个运行保留 ansible-runner 构件目录。如果设置为 0, 则工件目录不会被删除。	<p>Default:没有设置默认值</p> <p>CLI: <code>--rac</code> 或 <code>--ansible-runner-rotate-artifacts-count</code></p> <p>ENV: <code>ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_ROTATE_ARTIFACTS_COUNT</code></p> <p>Settings file:</p> <pre>ansible-navigator: ansible-runner: rotate-artifacts-count:</pre>
ansible-runner-timeout	ansible-runner 强制停止执行超时值。	<p>Default:没有设置默认值</p> <p>CLI: <code>--rt</code> 或 <code>--ansible-runner-timeout</code></p> <p>ENV: <code>ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_TIMEOUT</code></p> <p>Settings file:</p> <pre>ansible-navigator: ansible-runner: timeout:</pre>

参数	描述	设置选项
app	自动化内容导航器的入口点。	<p>Choices: collections, config, doc, images, inventory, replay, run or welcome</p> <p>Default: welcome</p> <p>CLI example: ansible-navigator collections</p> <p>ENV: ANSIBLE_NAVIGATOR_APP</p> <p>Settings file:</p> <pre>ansible-navigator: app:</pre>
cmdline	传递给相应命令的额外参数。	<p>Default: 没有默认值</p> <p>CLI : 位置</p> <p>ENV: ANSIBLE_NAVIGATOR_CMDLINE</p> <p>Settings file:</p> <pre>ansible-navigator: ansible: cmdline:</pre>
collection-doc-cache-path	集合 doc 缓存的路径。	<p>Default: \$HOME/.cache/ansible-navigator/collection_doc_cache.db</p> <p>CLI: --cdcp 或 --collection-doc-cache-path</p> <p>ENV: ANSIBLE_NAVIGATOR_COLLECTION_DOC_CACHE_PATH</p> <p>Settings file:</p> <pre>ansible-navigator: collection-doc-cache-path:</pre>

参数	描述	设置选项
container-engine	指定容器引擎（ auto=podman then docker ）。	<p>Choices: auto, podman 或 docker</p> <p>Default: auto</p> <p>CLI: --ce 或 --container-engine</p> <p>ENV: ANSIBLE_NAVIGATOR_CONTAINER_ENGINE</p> <p>Settings file:</p> <pre>ansible-navigator: execution-environment: container-engine:</pre>
display-color	在显示中使用颜色。	<p>Choices: True 或 False</p> <p>Default: True</p> <p>CLI: --dc 或 --display-color</p> <p>ENV: NO_COLOR</p> <p>Settings file:</p> <pre>ansible-navigator: color: enable:</pre>
editor-command	指定自动化内容导航器使用的编辑器	<p>默认值 : * vi +{line_number} {filename}</p> <p>CLI: --ecmd 或 --editor-command</p> <p>ENV: ANSIBLE_NAVIGATOR_EDITOR_COMMAND</p> <p>Settings file:</p> <pre>ansible-navigator: editor: command:</pre>

参数	描述	设置选项
editor-console	指定编辑器是否基于控制台。	<p>Choices: True 或 False</p> <p>Default: True</p> <p>CLI: --econ 或 --editor-console</p> <p>ENV: ANSIBLE_NAVIGATOR_EDIT OR_CONSOLE</p> <p>Settings file:</p> <pre>ansible-navigator: editor: console:</pre>
execute-environment	启用或禁用自动化执行环境的使用。	<p>Choices: True 或 False</p> <p>Default: True</p> <p>CLI: --ee 或 --execution-environment</p> <p>ENV:*</p> <p>ANSIBLE_NAVIGATOR_EXE CUTION_ENVIRONMENT</p> <p>Settings file:</p> <pre>ansible-navigator: execution-environment: enabled:</pre>

参数	描述	设置选项
execution-environment-image	指定自动化执行环境镜像的名称。	<p>Default: quay.io/ansible/ansible-runner:devel</p> <p>CLI: <code>--eei</code> 或 <code>--execution-environment-image</code></p> <p>ENV: ANSIBLE_NAVIGATOR_EXECUTION_ENVIRONMENT_IMAGE</p> <p>Settings file:</p> <pre>ansible-navigator: execution-environment: image:</pre>
execution-environment-volume-mounts	指定要在自动化执行环境中绑定的卷 (<code>--eev /home/user/test:/home/user/test:Z</code>)	<p>Default: 没有设置默认值</p> <p>CLI: <code>--eev</code> 或 <code>--execution-environment-volume-mounts</code></p> <p>ENV: ANSIBLE_NAVIGATOR_EXECUTION_ENVIRONMENT_VOLUME_MOUNTS</p> <p>Settings file:</p> <pre>ansible-navigator: execution-environment: volume-mounts:</pre>
log-append	指定是否应将日志消息附加到现有日志文件中，否则每个会话都会创建新的日志文件。	<p>Choices: True 或 False</p> <p>Default: True</p> <p>CLI: <code>--la</code> 或 <code>--log-append</code></p> <p>ENV: ANSIBLE_NAVIGATOR_LOG_APPEND</p> <p>Settings file:</p> <pre>ansible-navigator: logging: append:</pre>

参数	描述	设置选项
log-file	指定自动化内容导航器日志文件的完整路径。	<p>Default: \$PWD/ansible-navigator.log</p> <p>CLI: --lf 或 --log-file</p> <p>ENV: ANSIBLE_NAVIGATOR_LOG_FILE</p> <p>Settings file:</p> <pre>ansible-navigator: logging: file:</pre>
log-level	指定自动化内容导航器日志级别。	<p>Choices: debug, info, warning, error 或 critical</p> <p>Default: warning</p> <p>CLI: --ll 或 --log-level</p> <p>ENV: ANSIBLE_NAVIGATOR_LOG_LEVEL</p> <p>Settings file:</p> <pre>ansible-navigator: logging: level:</pre>
模式	指定 user-interface 模式。	<p>Choices: stdout 或 interactive</p> <p>Default: interactive</p> <p>CLI: -m 或 --mode</p> <p>ENV: ANSIBLE_NAVIGATOR_MODE</p> <p>Settings file:</p> <pre>ansible-navigator: mode:</pre>

参数	描述	设置选项
osc4	启用或禁用 OSC 4 支持的终端颜色。	<p>Choices: True 或 False</p> <p>Default: True</p> <p>CLI: --osc4 或 --osc4</p> <p>ENV: ANSIBLE_NAVIGATOR_OSC4</p> <p>Settings file:</p> <pre>ansible-navigator: color: osc4:</pre>
pass-environment-variable	指定要传递给的退出环境变量，并在自动化执行环境 (--penv MY_VAR) 内设置。	<p>Default: 没有设置默认值</p> <p>CLI: --penv 或 --pass-environment-variable</p> <p>ENV: ANSIBLE_NAVIGATOR_PAS S_ENVIRONMENT_VARIABL ES</p> <p>Settings file:</p> <pre>ansible-navigator: execution-environment: environment-variables: pass:</pre>
pull-policy	<p>指定镜像拉取策略。</p> <p>always - 总拉取 (pull) 镜像</p> <p>missing - 在本地没有时拉取</p> <p>Never - 不拉取镜像</p> <p>tag - 如果镜像标签为 latest 则总拉取镜像，否则在本地没有时进行拉取</p>	<p>Choices: always, missing, never, 或 tag</p> <p>Default: tag</p> <p>CLI: --pp 或 --pull-policy</p> <p>ENV: ANSIBLE_NAVIGATOR_PUL L_POLICY</p> <p>Settings file:</p> <pre>ansible-navigator: execution-environment: pull-policy:</pre>

参数	描述	设置选项
set-environment-variable	指定要在自动化执行环境内设置的环境变量和值 (--senv MY_VAR=42)	<p>Default:没有设置默认值</p> <p>CLI: --senv 或 --set-environment-variable</p> <p>ENV: ANSIBLE_NAVIGATOR_SET_ENVIRONMENT_VARIABLES</p> <p>Settings file:</p> <pre> ansible-navigator: execution-environment: environment-variables: set: </pre>

8.3. 自动化内容导航器配置子命令设置

下表描述了自动化内容导航器 **config** 子命令的每个参数和设置选项。

表 8.2. 自动化内容导航器 **config** 子命令参数设置

参数	描述	设置选项
config	指定 Ansible 配置文件的路径。	<p>Default:没有设置默认值</p> <p>CLI: -c 或 --config</p> <p>ENV: ANSIBLE_CONFIG</p> <p>Settings file:</p> <pre> ansible-navigator: ansible: config: path: </pre>

参数	描述	设置选项
help-config	以 stdout 模式显示 ansible-config 命令的帮助选项。	<p>Choices: * True 或 False</p> <p>Default: False</p> <p>CLI: --hc 或 --help-config</p> <p>ENV: ANSIBLE_NAVIGATOR_HELP_CONFIG</p> <p>Settings file:</p> <pre>ansible-navigator: help-config:</pre>

8.4. 自动化内容导航器 doc 子命令设置

下表描述了自动化内容导航器 **doc** 子命令的每个参数和设置选项。

表 8.3. 自动化内容导航器 doc 子命令参数设置

参数	描述	设置选项
help-doc	在 stdout 模式中显示 ansible-doc 命令的帮助选项。	<p>Choices: True 或 False</p> <p>Default: False</p> <p>CLI: --hd 或 --help-doc</p> <p>ENV: ANSIBLE_NAVIGATOR_HELP_DOC</p> <p>Settings file:</p> <pre>ansible-navigator: help-doc:</pre>

参数	描述	设置选项
plugin-name	指定插件名称。	<p>Default:没有设置默认值</p> <p>CLI : 位置</p> <p>ENV: ANSIBLE_NAVIGATOR_PLUGIN_NAME</p> <p>Settings file:</p> <pre>ansible-navigator: documentation: plugin: name:</pre>
plugin-type	指定插件类型。	<p>Choices: become, cache, callback, cliconf, connection, httpapi, inventory, lookup, module, netconf, shell, strategy, or vars</p> <p>Default: module</p> <p>CLI: -t 或 ----type</p> <p>ENV: ANSIBLE_NAVIGATOR_PLUGIN_TYPE</p> <p>Settings file:</p> <pre>ansible-navigator: documentation: plugin: type:</pre>

8.5. 自动化内容导航器 INVENTORY 子命令设置

下表描述了自动化内容导航器 **inventory** 子命令的每个参数和设置选项。

表 8.4. 自动化内容导航器 **inventory** 子命令参数设置

参数	描述	设置选项
----	----	------

参数	描述	设置选项
help-inventory	以 stdout 模式显示 ansible-inventory 命令的帮助选项。	<p>Choices: True 或 False</p> <p>Default: False</p> <p>CLI: --hi 或 --help-inventory</p> <p>ENV: ANSIBLE_NAVIGATOR_INVENTORY_DOC</p> <p>Settings file:</p> <pre>ansible-navigator: help-inventory:</pre>
清单 (inventory)	指定清单文件路径或逗号分隔的主机列表。	<p>Default: 没有设置默认值</p> <p>CLI: --i 或 --inventory</p> <p>ENV: ANSIBLE_NAVIGATOR_INVENTORIES</p> <p>Settings file:</p> <pre>ansible-navigator: inventories:</pre>
inventory-column	指定要在清单视图中显示的主机属性。	<p>Default: 没有设置默认值</p> <p>CLI: --ic 或 --inventory-column</p> <p>ENV:*</p> <p>ANSIBLE_NAVIGATOR_INVENTORY_COLUMNS</p> <p>Settings file:</p> <pre>ansible-navigator: inventory-columns:</pre>

8.6. 自动化内容导航器 REPLAY 子命令设置

下表描述了自动化内容导航器 **replay** 子命令的每个参数和设置选项。

表 8.5. 自动化内容导航器 **replay** 播放子命令参数设置

参数	描述	设置选项
----	----	------

参数	描述	设置选项
playbook-artifact-replay	指定要重播的 playbook 构件的路径。	<p>Default:没有设置默认值</p> <p>CLI : 位置</p> <p>ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_REPLAY</p> <p>Settings file:</p> <pre>ansible-navigator: playbook-artifact: replay:</pre>

8.7. 自动化内容导航器 RUN 子命令设置

下表描述了自动化内容导航器 `run` 子命令的每个参数和设置选项。

表 8.6. 自动化内容导航器 `run` 子命令参数设置

参数	描述	设置选项
playbook-artifact-replay	指定要重播的 playbook 构件的路径。	<p>Default:没有设置默认值</p> <p>CLI : 位置</p> <p>ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_REPLAY</p> <p>Settings file:</p> <pre>ansible-navigator: playbook-artifact: replay:</pre>
help-playbook	以 <code>stdout</code> 模式显示 <code>ansible-playbook</code> 命令的帮助选项。	<p>Choices: True 或 False</p> <p>Default: False</p> <p>CLI: <code>--hp</code> 或 <code>--help-playbook</code></p> <p>ENV: ANSIBLE_NAVIGATOR_HELP_PLAYBOOK</p> <p>Settings file:</p> <pre>ansible-navigator: help-playbook:</pre>

参数	描述	设置选项
清单 (inventory)	指定清单文件路径或逗号分隔的主机列表。	<p>Default: 没有设置默认值</p> <p>CLI: <code>--i</code> 或 <code>--inventory</code></p> <p>ENV: ANSIBLE_NAVIGATOR_INVENTORIES</p> <p>Settings file:</p> <pre>ansible-navigator: inventories:</pre>
inventory-column	指定要在清单视图中显示的主机属性。	<p>Default: 没有设置默认值</p> <p>CLI: <code>--ic</code> 或 <code>--inventory-column</code></p> <p>ENV:* ANSIBLE_NAVIGATOR_INVENTORY_COLUMNS</p> <p>Settings file:</p> <pre>ansible-navigator: inventory-columns:</pre>
playbook	指定 playbook 名称。	<p>Default: 没有设置默认值</p> <p>CLI: 位置</p> <p>ENV: ANSIBLE_NAVIGATOR_PLAYBOOK</p> <p>Settings file:*</p> <pre>ansible-navigator: ansible: playbook:</pre>

参数	描述	设置选项
playbook-artifact-enable	启用或禁用为已完成的 playbook 创建工作件。注意：当 playbook 需要用户输入时，与 --mode stdout 不兼容。	<p>Choices: True 或 False</p> <p>Default: True</p> <p>CLI: --pae 或 --playbook-artifact-enable</p> <p>ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_ENABLE</p> <p>Settings file:</p> <pre>ansible-navigator: playbook-artifact: enable:</pre>
playbook-artifact-save-as	指定从完成的 playbook 创建的工件的名称。	<p>Default: {playbook_dir}/{playbook_name}-artifact-{ts_utc}.json</p> <p>CLI: --pas 或 --playbook-artifact-save-as</p> <p>ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_SAVE_AS</p> <p>Settings file:</p> <pre>ansible-navigator: playbook-artifact: save-as:</pre>

第 9 章 使用自动化内容导航器对 ANSIBLE 内容进行故障排除

作为内容创建者，您可以使用自动化内容导航器对 Ansible 内容（收集、自动化执行环境和 playbook）进行故障排除，并以交互方式对 playbook 进行故障排除。您还可以比较自动化执行环境内部或外部的结果，并对任何问题进行故障排除。

9.1. 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
```

- 检查 playbook 运行时的 playbook 结果。

PLAY NAME	OK	CHANGED	UNREACHABLE	FAILED	SKIPPED	IGNORED	IN PROGRESS	TASK COUNT	PROGRESS
alt	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](#)

9.2. 有关自动化内容导航器的常见问题

使用以下自动化内容导航器常见问题解答，以帮助您对环境中的问题进行故障排除。

在使用自动化执行环境时，ansible.cfg 文件应位于什么位置？

最简单的方法是将 **ansible.cfg** 文件放置在直接位于 playbook 旁边的项目目录中。playbook 目录自动挂载到自动化执行环境中，Automation content navigator 会在那里找到 **ansible.cfg** 文件。如果 **ansible.cfg** 文件位于其他目录中，设置 **ANSIBLE_CONFIG** 变量，并将该目录指定为一个自定义卷挂载。（请参阅 [execution-environment-volume-mounts](#) 的自动化内容导航器设置。）

在没有使用自动化执行环境时，ansible.cfg 文件应位于什么位置？

在没有使用自动化执行环境时，Ansible 会在典型位置查找 **ansible.cfg**。详情请参阅 [Ansible 配置设置](#)。

在使用自动化执行环境时，Ansible 集合应放置在哪里？

使用 Ansible 集合的最简单位置是位于项目目录中，位于 playbook 相邻集合目录中（例如 **ansible-galaxy collections install ansible.utils -p ./collections**）。playbook 目录自动挂载到自动化执行环境中，Automation content navigator 会在此处查找集合。另一种选择是使用 Ansible Builder 将集合构建到自动化执行环境中。这有助于内容创建者编写准备好生产的 playbook，因为自动化控制器支持

playbook 相邻集合目录。如果集合位于另一个目录中，请设置 **ANSIBLE_COLLECTIONS_PATHS** 变量，并为该目录配置自定义卷挂载。（请参阅 [execution-environment-volume-mounts](#) 的 [自动化内容导航器常规设置](#)。

如果不使用自动化执行环境，应放置 Ansible 集合的位置？

如果不使用自动化执行环境，Ansible 会查看集合的默认位置。请参阅 [使用 Ansible 集合指南](#)。

在使用 vars_prompt 或 pause/prompt 时，为什么 playbook 会出现挂起的情况？

默认情况下，自动化内容导航器以与自动化控制器运行 playbook 相同的方式运行 playbook。这有助于内容创建者编写生产就绪的 playbook。如果您无法使用 **vars_prompt** 或 **pause/prompt**，禁用 **playbook-artifact** 创建会导致自动化内容导航器以与 **ansible-playbook** 兼容并允许用户交互的方式运行 playbook。

为什么自动化内容导航器更改终端颜色或外观？

自动化内容导航器查询终端以获取 OSC4 兼容性。OSC4、10、11、104、110、111 表示终端支持颜色更改和恢复。这可能是终端错误地代表了其能力。您可以通过设置 **--osc4 false** 来禁用 OSC4 检测。

（请参阅 [自动化内容导航器常规设置](#)，以了解如何使用环境变量或设置文件中处理此设置）。

如何更改自动化内容导航器使用的颜色？

使用 **--osc4 false** 来强制自动化内容导航器使用定义的终端颜色。（请参阅 [自动化内容导航器常规设置](#)，以了解如何使用环境变量或设置文件中处理此设置）。

playbook 目录中所有这些 site-artifact-2021-06-02T16:02:33.911259+00:00.json 文件是什么？

自动化内容导航器为每个 playbook 运行创建一个 playbook 工件。这有助于在自动化完成后，审查自动化的结果，与同事共享并进行故障排除，或保持合规或满足更改控制的目的。playbook 工件文件包含有关每个 play 和任务的详细信息，以及 playbook 运行的 **stdout**。您可以在自动化内容导航器会话中查看带有 **ansible-navigator replay <filename>** 或 **:replay <filename >** 的 playbook 工件。您可以根据所需的视图，查看所有带有 **--mode stdout** 和 **--mode 互动** 的 playbook 工件。您可以禁用 playbook 工件编写和默认的文件命名约定。（请参阅 [自动化内容导航器常规设置](#)，以了解如何使用环境变量或设置文件中处理此设置）。

当使用 :open 时，为什么会打开 vi？

自动化内容浏览器会在默认编辑器的终端中打开任何显示内容。默认值为 **vi +{line_number} {filename}** 或 **EDITOR** 环境变量的当前值。与此相关的是 **editor-console** 设置，它指示编辑器基于控制台还是终端。以下是可能有用的备用设置示例：

```
# emacs
ansible-navigator:
  editor:
    command: emacs -nw +{line_number} {filename}
    console: true
```

```
# vscode
ansible-navigator:
  editor:
    command: code -g {filename}:{line_number}
    console: false
```

```
# pycharm
ansible-navigator:
  editor:
    command: charm --line {line_number} {filename}
    console: false
```

配置设置应用的顺序是什么？

自动化内容导航器配置系统从各种来源拉取设置，并按以下顺序应用它们（其中最后一个应用的更改是最预先评估的）：

1. 默认内部值
2. 来自设置文件中的值
3. 来自环境变量中的值
4. 命令行中指定的标志和参数
5. 在基于文本的用户界面中输入：命令

我如何对其进行故障排除？

自动化内容导航器具有相关的日志记录消息。您可以使用 `--log-level debug` 来启用 `debug` 级别的日志。如果您认为可能发现了一个程序漏洞，请报告相关问题，并包含日志文件中的详细信息。