



Red Hat Ansible Automation Platform 2.0-*ea*

Ansible Navigator Creator ガイド

Ansible Creator ワークフローで Ansible Navigator の使用

Red Hat Ansible Automation Platform 2.0-ea Ansible Navigator Creator ガイド

Ansible Creator ワークフローで Ansible Navigator の使用

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2021 | You need to change the HOLDER entity in the en-US/Ansible_Navigator_Creator_Guide.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

フィードバックの提供: 本書の改善のご意見がある場合や、誤植を発見した場合は、<http://issues.redhat.com> で問題について起票してください。Ansible Automation Platform プロジェクトを選択し、Documentation コンポーネントを使用します。

目次

第1章 ANSIBLE コンテンツナビゲーター	3
1.1. 自動化コンテンツナビゲーターについて	3
1.1.1. 自動化コンテンツナビゲーターを使用する理由	3
1.2. 自動化コンテンツナビゲーターの起動	3
1.2.1. stdout モード	3
1.2.2. テキストベースのユーザーインターフェースモード	4
1.3. 自動化コンテンツナビゲーターコマンド	5
1.4. 自動コンテンツナビゲーターコマンドの比較	5
第2章 RHEL への自動コンテンツナビゲーターのインストール	7
2.1. 前提条件	7
2.2. RPM からの RHEL への自動化コンテンツナビゲーターのインストール	7
第3章 自動化コンテンツナビゲーターを使用した自動化実行環境の確認	8
3.1. 前提条件	8
3.2. 自動化コンテンツナビゲーターからの自動化実行環境の確認	8
第4章 自動化コンテンツナビゲーターでインベントリの確認	10
4.1. 自動化コンテンツナビゲーターからインベントリの確認	10
第5章 自動化コンテンツナビゲーターでコレクションを参照	12
5.1. 自動化コンテンツナビゲーターコレクションの表示	12
5.2. 自動化コンテンツナビゲーターからのコレクションの参照	12
5.3. 自動化コンテンツナビゲーターのドキュメントの確認	14
第6章 自動化コンテンツナビゲーターを使用した ANSIBLE PLAYBOOK の実行	17
6.1. 自動化コンテンツナビゲーターからの PLAYBOOK の実行	17
6.2. 自動化コンテンツナビゲーターアーティファクトファイルを使用した PLAYBOOK の結果の確認	19
第7章 自動化コンテンツナビゲーターで ANSIBLE 設定の確認	20
7.1. 自動化コンテンツナビゲーターで ANSIBLE 設定の確認	20
第8章 自動化コンテンツナビゲーターの構成設定	22
8.1. 自動化コンテンツナビゲーター設定	22
8.2. 自動化コンテンツナビゲーターの一般設定	23
8.3. 自動化コンテンツナビゲーターの CONFIG サブコマンド設定	31
8.4. 自動化コンテンツナビゲーターの DOC サブコマンド設定	32
8.5. 自動化コンテンツナビゲーターの INVENTORY サブコマンド設定	33
8.6. 自動化コンテンツナビゲーターの REPLAY サブコマンド設定	34
8.7. 自動化コンテンツナビゲーターの RUN サブコマンド設定	35
第9章 自動化コンテンツナビゲーターを使用した ANSIBLE コンテンツのトラブルシューティング	38
9.1. 自動化コンテンツナビゲーターアーティファクトファイルを使用した PLAYBOOK の結果の確認	38
9.2. 自動化コンテンツナビゲーターでよくある質問 (FAQ)	38

第1章 ANSIBLE コンテンツナビゲーター

コンテンツ作成者は、自動化コンテンツナビゲーターを使用して、Red Hat Ansible Automation Platform と互換性のある Ansible Playbook、コレクション、およびロールを開発できます。ローカル開発環境、自動化実行環境、または Red Hat Ansible Automation Platform で、自動化コンテンツナビゲーターを使用できます。また、すべてでシームレスかつ予測可能な結果が得られます。また、自動化コンテンツナビゲーターは、Playbook の開発および問題領域のトラブルシューティングに役立つアーティファクトファイルも作成します。

1.1. 自動化コンテンツナビゲーターについて

自動化コンテンツナビゲーターは、次のことができるテキストベースのユーザーインターフェースを備えたコマンドラインのコンテンツクリエイターに焦点を当てたツールです。

- ジョブおよび Playbook を起動し、監視します。
- 保存され、完成した Playbook とジョブ実行アーティファクトを JSON 形式で共有します。
- 自動化実行環境を閲覧およびイントロスペクションします。
- ファイルベースのインベントリを参照します。
- Ansible モジュールのドキュメントをレンダリングし、Playbook で使用できるサンプルを展開します。

1.1.1. 自動化コンテンツナビゲーターを使用する理由

現在、**ansible-playbook** コマンドなど、Ansible コアの一部である Ansible コマンドは、コンテナで実行できません。このニーズを満たすために、自動化コンテンツナビゲーターは、自動化実行環境でこれらの Ansible CLI コマンドセットを有効にします。Automation コンテンツナビゲーターには、テキストベースのユーザーインターフェース内にさらに詳細な出力が含まれます。

1.2. 自動化コンテンツナビゲーターの起動

自動化コンテンツナビゲーターは 2 つのモードで動作します。

stdout モード

コマンドラインで既存の Ansible コマンドおよび拡張機能の大部分を受け入れます。

テキストベースのユーザーインターフェースモード

アーティファクトファイルを使用して実行した後に、コンテンツの評価、ならびに Playbook の実行およびトラブルシューティングを行う上で、Ansible コマンドに対する、インタラクティブなテキストベースのインターフェースを Ansible コマンドに提供します。

1.2.1. stdout モード

-m stdout サブコマンドを使用する場合は、自動化コンテンツナビゲーターは、自動化実行環境内の **ansible-playbook** やローカル開発環境など、一般的な Ansible コマンドを使用します。このモードでは、クイックタスクに精通しているコマンドを引き続き使用できます。

また、自動化コンテンツナビゲーターは、このモードで豊富なヘルプも提供します。

--help

ansible-navigator コマンドまたはサブコマンド (**ansible-navigator config --help** など) からアクセスできます。

サブコマンドのヘルプ

サブコマンド (例: **ansible-navigator config --help-config**) からアクセスできます。このサブコマンドは、関連する Ansible コマンド (この例では **ansible-config**) でサポートされるすべてのパラメーターの詳細を確認できます。

```
$ 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
```

1.2.2. テキストベースのユーザーインターフェースモード

テキストベースのユーザーインターフェースモードは、自動化実行環境、コレクション、Playbook、およびインベントリとの対話を強化します。このモードは、Visual Studio Code などの統合開発環境 (IDE) と互換性があります。

```
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  nl scroll           esc back           :help help
```

このモードには、便利なユーザーインターフェースオプションが多数含まれています。

コロンコマンド

:run、**:collections** など、コロンを使用して、すべての Automation コンテンツナビゲーターコマンドにアクセスできます。

テキストベースのインターフェースの移動

この画面には、ページアップまたはページダウン、スクロール、前の画面へのエスケープ、または **:help** へのアクセス方法が表示されます。

行番号による出力

表示された出力の行番号の前にコロンを付ける (例: **:12**) と、その行にアクセスできます。

色でコーディングされた出力

色を有効にすると、自動化コンテンツナビゲーターは非推奨のモジュールなどのアイテムを赤で表示します。

ページネーションおよびスクロール

各自動化コンテンツナビゲーター画面の下部に表示されるオプションを使用して、ページを上下に移動したり、スクロールしてエスケープしたりできます。

自動化コンテンツナビゲーターを実行したら、モード間の切り替えができません。

1.3. 自動化コンテンツナビゲーターコマンド

自動化コンテンツナビゲーターコマンドは、一般的な Ansible CLI コマンドを **-m stdout** モードで実行します。関連する Ansible CLI コマンドで許可されるすべてのサブコマンドおよびオプションを使用できます。詳細は、**ansible-navigator --help** を使用してください。

表1.1 自動コンテンツナビゲーターコマンドの表

コマンド	説明	CLI の例
collections	利用可能なコレクションを調べる	ansible-navigator collections --help
config	現在の Ansible 設定を調べる	ansible-navigator config --help
doc	モジュールまたはプラグインのドキュメントを参照する	ansible-navigator doc --help
images	実行環境イメージを調べる	ansible-navigator images --help
inventory	インベントリを調べる	ansible-navigator inventory -help
replay	Playbook アーティファクトを使用して以前の実行を調べる	ansible-navigator replay --help
run	Playbook を実行する	ansible-navigator run --help
welcome	Welcome ページから開始する	ansible-navigator welcome --help

1.4. 自動コンテンツナビゲーターコマンドの比較

自動化コンテンツナビゲーターコマンドは、一般的な Ansible CLI コマンドを **-m stdout** モードで実行します。関連する Ansible CLI コマンドで許可されるすべてのサブコマンドおよびオプションを使用できます。詳細は、**ansible-navigator --help** を使用してください。

表1.2 自動コンテンツナビゲーターと Ansible CLI コマンドの表の比較

Ansible ナビゲーターコマンド	Ansible CLI コマンド
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 RHEL への自動コンテンツナビゲーターのインストール

コンテンツ作成者は、Red Hat Enterprise Linux (RHEL) 8 に、自動化コンテンツナビゲーターをインストールできます。

2.1. 前提条件

- RHEL 8 以降をインストールしている。

2.2. RPM からの RHEL への自動化コンテンツナビゲーターのインストール

RPM から、Red Hat Enterprise Linux (RHEL) に自動化コンテンツナビゲーターをインストールできません。

前提条件

- Red Hat Subscription Manager でシステムを登録している。

手順

1. Red Hat Ansible Automation Platform SKU を割り当てます。

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

2. RHEL 8 のリポジトリを有効にします。

```
$ sudo subscription-manager repos --enable ansible-automation-platform-2.0-early-access-for-rhel-8-x86_64-rpms
```

3. 自動化コンテンツナビゲーターをインストールします。

```
$ dnf install ansible-navigator
```

検証

- 自動化コンテンツナビゲーターのインストールを確認します。

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

自動化コンテンツナビゲーターの主な機能を説明するヘルプ出力が表示されます。

これで、自動化コンテンツナビゲーターをインストールし、このツールを使用して Playbook を評価し、コレクションおよびインベントリを表示して、モジュールレベルのドキュメントを詳しく調べることができるようになりました。

第3章 自動化コンテンツナビゲーターを使用した自動化実行環境の確認

コンテンツ開発者は、自動化コンテンツナビゲーターで自動化実行環境を確認し、自動化実行環境に含まれるパッケージとコレクションを表示できます。自動化コンテンツナビゲーターは Playbook を実行して結果を抽出し、表示します。

3.1. 前提条件

- 追加の自動化実行環境にアクセスする必要がある場合は、Red Hat レジストリーに対して認証されている。詳細は、「[Red Hat コンテナレジストリーの認証](#)」を参照してください。

3.2. 自動化コンテンツナビゲーターからの自動化実行環境の確認

自動化コンテンツナビゲーターのテキストベースのユーザーインターフェースを使用して、自動化実行環境を確認できます。

前提条件

- 自動化実行環境

手順

- 自動化コンテンツナビゲーター設定に含まれる自動化実行環境を確認します。

```
$ 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

- 詳細については、詳しく調べたい自動化実行環境の数を入力してください。

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

インストールされている各自動化実行環境のパッケージおよびバージョン、および Ansible バージョンに含まれるコレクションを確認できます。

- 必要に応じて、使用する自動化実行環境を渡すことができます。これはプライマリーであり、自動化コンテンツナビゲーターが使用する自動化実行環境です。

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

検証

- 自動化実行環境の出力を確認します。

	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 自動化コンテンツナビゲーターでインベントリーの確認

コンテンツ作成者は、自動化コンテンツナビゲーターを使用して Ansible インベントリーを確認し、グループとホストをインタラクティブに調べることができます。

4.1. 自動化コンテンツナビゲーターからインベントリーの確認

インタラクティブモードの自動化コンテンツナビゲーターのテキストベースのユーザーインターフェースを使用して Ansible インベントリーを確認し、グループとホストを詳しく調べて詳細を確認できます。

前提条件

- 有効なインベントリーファイルまたはインベントリープラグイン

手順

1. 自動化コンテンツナビゲーターの起動

```
$ ansible-navigator
```

必要に応じて、コマンドラインから **ansible-navigator inventory -i simple_inventory.yml** を入力してインベントリーを直接表示できます。

2. インベントリーの確認

```
: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
```

3. **0** と入力してグループを表示します。

```

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

TAXONOMY フィールドには、選択したグループまたはノードが属するグループの階層の詳細が表示されます。

4. 調べるグループに対応する番号を入力します。

```

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

5. 調べるホストに対応する番号を入力します。番号が 9 以上になる場合は、**:<number>** と入力します。

```
[node-1]
```

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

検証

- インベントリーの出力を確認します。

```
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
```

関連情報

- [ansible-inventory](#)
- [Ansible インベントリーの概要](#)

第5章 自動化コンテンツナビゲーターでコレクションを参照

コンテンツ作成者は、自動化コンテンツナビゲーターで Ansible コレクションを参照し、ローカルまたは自動化実行環境で開発した各コレクションをインタラクティブに調べることができます。

5.1. 自動化コンテンツナビゲーターコレクションの表示

自動化コンテンツナビゲーターは、各コレクションに以下の詳細を含むコレクションに関する情報を表示します。

SHADOWED

コレクションの追加コピーが検索順序の上位にあり、Playbook がそのコレクションを優先することを示します。

TYPE

コレクションが自動化実行環境にマウントされている自動化実行環境またはボリュームに **bind_mount** として含まれるかどうかを示します。

PATH

コレクション TYPE フィールドに基づいて、自動化実行環境またはローカルファイルシステム内のコレクションの場所を反映します。

	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. 自動化コンテンツナビゲーターからのコレクションの参照

インタラクティブモードの自動化コンテンツナビゲーターのテキストベースのユーザーインターフェースを使用して Ansible コレクションを参照し、各コレクションを詳しく調べることができます。自動化コンテンツナビゲーターは、現在のプロジェクトディレクトリー内のコレクションと自動化実行環境で利用可能なコレクションを表示します。

前提条件

- ローカルでアクセス可能なコレクションまたはインストールされた自動化実行環境。

手順

1. 自動化コンテンツナビゲーターの起動

```
$ ansible-navigator
```

2. コレクションを参照します。または、**ansible-navigator collections** と入力してコレクションを直接参照することもできます。

```
$ :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. 確認するコレクションの数を入力します。

```
: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. 調べるモジュールに対応する番号を入力します。

```
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. 必要に応じて、このモジュールのドキュメントの例に移動します。

```
:{{ 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 |

```

- 必要に応じて、エディターでサンプルを開き、それを Playbook にコピーします。

```

:open

```

```

EXPLORE... ! tmpvsqroz08.yaml x
> .ansible
> vagrant-test
> .gitignore
ansible-navigator.log
! ansible-navigator.yml
ansible.cfg U
! quicklab_inventory.yml M
{} quicklab_playbook-artifa... U
{} quicklab_playbook-artifa... U
{} quicklab_playbook-artifa... U
{} quicklab_playbook-artifa... U
{} quicklab_playbook-artifa... U
! quicklab_playbook.yml
① README.md
{} simple_playbook-artifact... U
{} simple_playbook-artifact... U
{} simple_playbook-artifact... U
{} simple_playbook-artifact... U
{} simple_playbook-artifact... U
{} simple_playbook-artifact... U

tmp > ! tmpvsqroz08.yaml > ...
51 version_added_collection: ansible.utils
52 examples: |-
53 ##### Simple examples
54
55 - name: Check if 10.1.1.1 is in 10.0.0.0/8
56   ansible.builtin.set_fact:
57     data: "{{ '10.1.1.1' is ansible.utils.in_network '10.0.0.0/8' }}"
58
59 # TASK [Check if 10.1.1.1 is in 10.0.0.0/8] *****
60 # ok: [localhost] => {
61 #   "ansible_facts": {
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: python3
ANSIBLE_UTILS.IN_NETWORK: Test if IP address falls in the network
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
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

検証

- コレクション一覧を参照します。

	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/

関連情報

- [ansible-collection](#)
- [Ansible インベントリーの概要](#)

5.3. 自動化コンテンツナビゲーターのドキュメントの確認

対話モードで自動化コンテンツナビゲーターのテキストベースのユーザーインターフェースを使用し、コレクションおよびプラグインの Ansible ドキュメントを確認することができます。自動化コンテンツナビゲーターは、現在のプロジェクトディレクトリー内のコレクションと自動化実行環境で利用可能なコレクションを表示します。

前提条件

- ローカルでアクセス可能なコレクションまたはインストールされた自動化実行環境。

手順

- 自動化コンテンツナビゲーターの起動

```
$ ansible-navigator
```

- 対象のモジュールを確認します。または、**ansible-navigator doc** と入力してドキュメントに直接アクセスすることもできます。

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

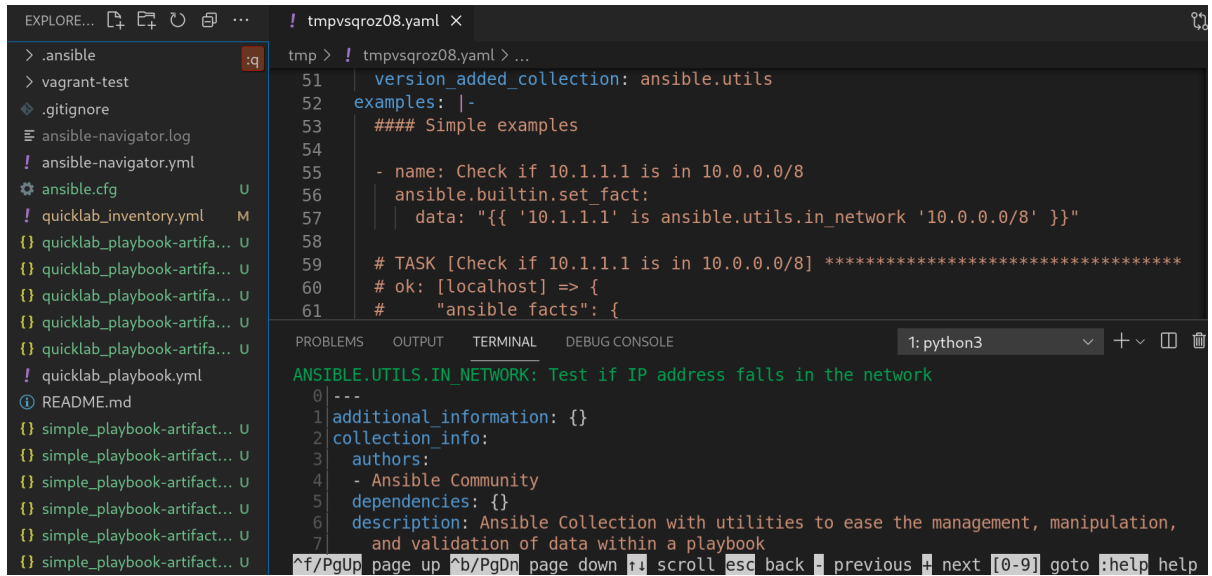
```
ANSIBLE.UTILS.IP_ADDRESS: Test if something in an IP address
0 | ---
1 | additional_information: {}
2 | collection_info:
3 |   authors:
4 |     - Ansible Community
5 |   dependencies: {}
6 |   description: Ansible Collection with utilities to ease the management, manipulation,
7 |     and validation 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...>
```

- このモジュールのドキュメント例に移動します。

```
:{{ 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 |
```

- 必要に応じて、エディターでサンプルを開き、それを Playbook にコピーします。

```
:open
```



The screenshot displays the Ansible Navigator Creator interface. On the left, a file explorer shows a project structure with 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 shows a YAML file named `tmpvsqroz08.yaml` with the following content:

```
51 version_added_collection: ansible.utils
52 examples: |-
53     ### Simple examples
54
55     - name: Check if 10.1.1.1 is in 10.0.0.0/8
56       ansible.builtin.set_fact:
57         data: "{{ '10.1.1.1' is ansible.utils.in_network '10.0.0.0/8' }}"
58
59     # TASK [Check if 10.1.1.1 is in 10.0.0.0/8] *****
60     # ok: [localhost] => {
61     #   "ansible_facts": {
```

Below the editor is a terminal window with the following output:

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: python3
ANSIBLE_UTILS_IN_NETWORK: Test if IP address falls in the network
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
```

エディターの設定方法の詳細は「[自動化コンテンツナビゲーターの一般設定](#)」を参照してください。

関連情報

- [ansible-collection](#)
- [Ansible インベントリーの概要](#)

第6章 自動化コンテンツナビゲーターを使用した ANSIBLE PLAYBOOK の実行

コンテンツ作成者は、自動化コンテンツナビゲーターを使用して Ansible Playbook を実行し、各プレイとタスクの結果をインタラクティブに調べて、Playbook を検証またはトラブルシューティングすることができます。また、実行環境内で Ansible Playbook を実行して、実行環境なしでも、問題を比較し、トラブルシューティングすることもできます。

6.1. 自動化コンテンツナビゲーターからの PLAYBOOK の実行

自動化ナビゲーターのテキストベースのユーザーインターフェースで Ansible Playbook を実行して、タスクの実行を追跡し、各タスクの結果を詳しく調べることができます。

前提条件

- Playbook
- **localhost** または inventory プラグインを使用しない場合の、有効なインベントリーファイルです。

手順

1. 自動化コンテンツナビゲーターの起動

```
$ ansible-navigator
```

2. Playbook を実行します。必要に応じて、**ansible-navigator run simple-playbook.yml -i inventory.yml** を実行して Playbook を直接実行することができます。

```
$ :run
```

3. インベントリーおよびその他のコマンドラインパラメーターを確認または追加します。

```
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

4. Tab キーを押して **Submit** に移動し、Enter キーを押します。実行中のタスクが表示されるはずですが。

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

5. プレイ結果にステップインするプレイの横に番号を入力します。9 を超える数字の場合は：**<number>** と入力します。

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

自動化コンテンツナビゲーターで色を有効にしている場合は、失敗したタスクが赤で表示されることに注意してください。

- タスクの結果を確認するタスクの横に番号を入力します。9を超える番号の場合は `:<number>` と入力します。

```
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
```

- 必要に応じて、`:doc` と入力し、トラブルシューティングを支援するタスクで使用されるモジュールまたはプラグインのドキュメントを起動します。

```
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 ...>
```

関連情報

- [ansible-playbook](#).
- [Ansible Playbook の概要](#)

6.2. 自動化コンテンツナビゲーターアーティファクトファイルを使用した PLAYBOOK の結果の確認

Automation コンテンツナビゲーターは、JSON アーティファクトファイルに Playbook 実行の結果を保存します。このファイルを使用して、Playbook の結果を他の人と共有したり、セキュリティやコンプライアンスの理由で保存したり、後で確認してトラブルシューティングしたりできます。Playbook の実行を確認するためにアーティファクトファイルのみが必要になります。Playbook 自体またはイベントリクエストにアクセスする必要はありません。

前提条件

- Playbook 実行からの自動化コンテンツナビゲーターアーティファクトの JSON ファイル。

手順

- アーティファクトファイルによる自動化コンテンツナビゲーターの開始

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

1. Playbook の実行時に一致する Playbook の結果を確認します。

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

Playbook を実行した後と同じように、プレイとタスクの横に番号を入力して、それぞれにステップインして結果を確認できるようになりました。

関連情報

- [ansible-playbook](#).
- [Ansible Playbook の概要](#)

第7章 自動化コンテンツナビゲーターで ANSIBLE 設定の確認

コンテンツ作成者は、自動化コンテンツナビゲーターで Ansible 設定を確認し、インタラクティブに設定を調べることができます。

7.1. 自動化コンテンツナビゲーターで ANSIBLE 設定の確認

Ansible 設定は、自動化コンテンツナビゲーターのテキストベースのユーザーインターフェースを対話モードで確認でき、設定を詳しく調べることができます。自動化コンテンツナビゲーターは、アクセス可能な Ansible 設定ファイルから結果をプルするか、設定ファイルが存在しない場合にはデフォルトを返します。

手順

1. 自動化コンテンツナビゲーターの起動

```
$ ansible-navigator
```

必要に応じて、コマンドラインから **ansible-navigator config** を入力し、Ansible 設定に直接アクセスできます。

2. Ansible 設定を確認します。

```
: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',		
5 ANSIBLE_COW_PATH	True	default	default	None
6 ANSIBLE_COW_SELECTION	True	default	default	default
7 ANSIBLE_FORCE_COLOR	True	default	default	False

一部の値は、自動化実行環境が機能するために必要な自動化実行環境の設定を反映しています。これらは、Ansible 設定ファイルでは設定できないデフォルト以外の設定として表示されません。

3. 調べる設定に対応する番号を入力します。番号が 9 以上になる場合は、**:<number>** と入力します。

```
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
```

出力には、現在の **設定** と **デフォルト** が表示されます。設定は自動化実行環境からのものであるため、この例の **source** は **env** であることに注意してください。

快捷

- 設定出力を確認します。

```
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
```

関連情報

- [ansible-config](#)
- [Ansible 設定の概要](#)

第8章 自動化コンテンツナビゲーターの構成設定

コンテンツ作成者は、開発環境に合わせるために、自動化コンテンツナビゲーターを設定できます。

8.1. 自動化コンテンツナビゲーター設定

以下を使用して、デフォルトの自動化コンテンツナビゲーター設定を変更できます。

- コマンドライン
- 設定ファイル内で
- 環境変数として

自動化コンテンツナビゲーターは以下の順序で設定ファイルを確認し、最初の一致を使用します。

- **ANSIBLE_NAVIGATOR_CONFIG**: 設定されている場合は、設定ファイルのパス環境変数。
- **./ansible-navigator.<ext>**: 現行プロジェクトディレクトリー内の設定ファイルです。ファイル名にドットはありません。
- **~/ansible-navigator.<ext>**: ファイル名にドットが含まれるホームディレクトリーです。

自動化コンテンツナビゲーター設定ファイルを作成する場合は、以下を考慮してください。

- 設定ファイルは、**JSON** 形式または **YAML** 形式のいずれかを使用できます。
- **JSON** 形式の設定の場合、拡張は **.json** にする必要があります。
- **YAML** 形式の設定には、拡張子は **.yml** または **.yaml** である必要があります。
- プロジェクトおよびホームディレクトリーにはそれぞれ1つの設定ファイルのみを含めることができます。
- 自動化コンテンツナビゲーターが、いずれかのディレクトリーで複数の設定ファイルを見つけると、エラーが発生します。

以下の設定ファイル例をそれらのパスのいずれかにコピーして、**ansible-navigator** 設定ファイルを開始できます。

```
---
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. 自動化コンテンツナビゲーターの一般設定

以下の表は、自動化コンテンツナビゲーターの一般的な各パラメーターと設定オプションについて説明しています。

表8.1 自動化コンテンツナビゲーターの一般的なパラメーター設定テーブル

パラメーター	説明	オプションの設定
--------	----	----------

パラメーター	説明	オプションの設定
ansible-runner-artifact-dir	ansible-runner によって生成されたアーティファクトを保存するディレクトリーパス。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --rad または --ansible-runner-artifact-dir</p> <p>ENV: ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_ARTIFACT_DIR</p> <p>設定ファイル:</p> <pre>ansible-navigator: ansible-runner: artifact-dir:</pre>
ansible-runner-rotate-artifacts-count	最後の n の実行には、ansible-runner アーティファクトディレクトリーを保持します。0 に設定すると、アーティファクトディレクトリーは削除されません。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --rac または --ansible-runner-rotate-artifacts-count</p> <p>ENV: ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_ROTATE_ARTIFACTS_COUNT</p> <p>設定ファイル:</p> <pre>ansible-navigator: ansible-runner: rotate-artifacts-count:</pre>
ansible-runner-timeout	ansible-runner が強制的に実行を停止する場合のタイムアウト値。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --rt または --ansible-runner-timeout</p> <p>ENV: ANSIBLE_NAVIGATOR_ANSIBLE_RUNNER_TIMEOUT</p> <p>設定ファイル:</p> <pre>ansible-navigator: ansible-runner: timeout:</pre>

パラメーター	説明	オプションの設定
app	自動化コンテンツナビゲーターのエントリーポイント。	<p>選択肢: collections、config、doc、images、inventory、replay、run、または welcome</p> <p>デフォルト: welcome</p> <p>CLI の例: ansible-navigator collections</p> <p>ENV: ANSIBLE_NAVIGATOR_APP</p> <p>設定ファイル:</p> <pre>ansible-navigator: app:</pre>
cmdline	対応するコマンドに渡す追加のパラメーター。	<p>デフォルト: デフォルト値なし</p> <p>CLI: 位置</p> <p>ENV: ANSIBLE_NAVIGATOR_CMDLINE</p> <p>設定ファイル:</p> <pre>ansible-navigator: ansible: cmdline:</pre>
collection-doc-cache-path	コレクションのドキュメントキャッシュへのパス。	<p>Default: \$HOME/.cache/ansible-navigator/collection_doc_cache.db</p> <p>CLI: --cdcp または --collection-doc-cache-path</p> <p>ENV: ANSIBLE_NAVIGATOR_COLLECTION_DOC_CACHE_PATH</p> <p>設定ファイル:</p> <pre>ansible-navigator: collection-doc-cache-path:</pre>

パラメーター	説明	オプションの設定
container-engine	コンテナエンジンを指定します (auto=podman then docker).	<p>選択肢: auto、podman、または docker</p> <p>デフォルト: auto</p> <p>CLI: --ce または --container-engine</p> <p>ENV: ANSIBLE_NAVIGATOR_CONTAINER_ENGINE</p> <p>設定ファイル:</p> <pre>ansible-navigator: execution-environment: container-engine:</pre>
display-color	ディスプレイでの色の使用を有効にします。	<p>選択肢: True または False</p> <p>デフォルト: True</p> <p>CLI: --dc または --display-color</p> <p>ENV: NO_COLOR</p> <p>設定ファイル:</p> <pre>ansible-navigator: color: enable:</pre>
editor-command	自動化コンテンツナビゲーターが使用するエディターの指定	<p>デフォルト: <code>* vi +{line_number} {filename}</code></p> <p>CLI: --ecmd または --editor-command</p> <p>ENV: ANSIBLE_NAVIGATOR_EDITOR_COMMAND</p> <p>設定ファイル:</p> <pre>ansible-navigator: editor: command:</pre>

パラメーター	説明	オプションの設定
editor-console	エディターがコンソールベースであるかどうかを指定します。	<p>選択肢: True または False</p> <p>デフォルト: True</p> <p>CLI: --econ または --editor-console</p> <p>ENV: ANSIBLE_NAVIGATOR_EDITOR_CONSOLE</p> <p>設定ファイル:</p> <pre>ansible-navigator: editor: console:</pre>
execution-environment	自動化実行環境の使用を有効または無効にします。	<p>選択肢: True または False</p> <p>デフォルト: True</p> <p>CLI: --ee または --execution-environment</p> <p>ENV:* ANSIBLE_NAVIGATOR_EXECUTION_ENVIRONMENT</p> <p>設定ファイル:</p> <pre>ansible-navigator: execution-environment: enabled:</pre>
execution-environment-image	自動化実行環境イメージの名前を指定します。	<p>デフォルト: quay.io/ansible/ansible-runner:devel</p> <p>CLI: --eei または --execution-environment-image</p> <p>ENV: ANSIBLE_NAVIGATOR_EXECUTION_ENVIRONMENT_IMAGE</p> <p>設定ファイル:</p> <pre>ansible-navigator: execution-environment: image:</pre>

パラメーター	説明	オプションの設定
execution-environment-volume-mounts	自動化実行環境内にバインドマウントするボリュームを指定します (--eev /home/user/test:/home/user/test:Z)	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --eev または --execution-environment-volume-mounts</p> <p>ENV: ANSIBLE_NAVIGATOR_EXECUTION_ENVIRONMENT_VOLUME_MOUNTS</p> <p>設定ファイル:</p> <pre>ansible-navigator: execution-environment: volume-mounts:</pre>
log-append	ログメッセージを既存のログファイルに追加するかどうかを指定します。追加しないと、セッションごとに新しいログファイルが作成されます。	<p>選択肢: True または False</p> <p>デフォルト: True</p> <p>CLI: --la または --log-append</p> <p>ENV: ANSIBLE_NAVIGATOR_LOG_APPEND</p> <p>設定ファイル:</p> <pre>ansible-navigator: logging: append:</pre>
log-file	自動化コンテンツナビゲーターのログファイルの完全パスを指定します。	<p>デフォルト: \$PWD/ansible-navigator.log</p> <p>CLI: --lf または --log-file</p> <p>ENV: ANSIBLE_NAVIGATOR_LOG_FILE</p> <p>設定ファイル:</p> <pre>ansible-navigator: logging: file:</pre>

パラメーター	説明	オプションの設定
ログレベル	自動化コンテンツナビゲーターのログレベルを指定します。	<p>選択肢: debug、info、warning、error、または critical</p> <p>デフォルト: warning</p> <p>CLI: --ll または --log-level</p> <p>ENV: ANSIBLE_NAVIGATOR_LOG_LEVEL</p> <p>設定ファイル:</p> <pre> ansible-navigator: logging: level: </pre>
mode	ユーザーインターフェースモードを指定します。	<p>選択肢: stdout または interactive</p> <p>デフォルト: interactive</p> <p>CLI: -m または --mode</p> <p>ENV: ANSIBLE_NAVIGATOR_MODE</p> <p>設定ファイル:</p> <pre> ansible-navigator: mode: </pre>
osc4	OSC 4 で端末の色変更のサポートを有効または無効にします。	<p>選択肢: True または False</p> <p>デフォルト: True</p> <p>CLI: --osc4 または --osc4</p> <p>ENV: ANSIBLE_NAVIGATOR_OSC4</p> <p>設定ファイル:</p> <pre> ansible-navigator: color: osc4: </pre>

パラメーター	説明	オプションの設定
pass-environment-variable	<p>自動化実行環境内で渡される終了環境変数を指定します (--penv MY_VAR)。</p>	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --penv または --pass-environment-variable</p> <p>ENV: ANSIBLE_NAVIGATOR_PAS S_ENVIRONMENT_VARIABL ES</p> <p>設定ファイル:</p> <pre>ansible-navigator: execution-environment: environment-variables: pass:</pre>
pull-policy	<p>イメージプルポリシーを指定します。</p> <p>always: 常にイメージをプルします。</p> <p>missing: ローカルで利用できない場合はプル</p> <p>never: イメージをプルしません。</p> <p>tag: イメージタグが latest の場合は、常にイメージをプルしてください。そうでないと、ローカルで利用可能な場合にプルされません。</p>	<p>選択肢: always、missing、never、または tag</p> <p>デフォルト: tag</p> <p>CLI: --pp または --pull-policy</p> <p>ENV: ANSIBLE_NAVIGATOR_PUL L_POLICY</p> <p>設定ファイル:</p> <pre>ansible-navigator: execution-environment: pull-policy:</pre>

パラメーター	説明	オプションの設定
set-environment-variable	自動化実行環境内で設定する環境変数および値を指定します (--senv MY_VAR=42)。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --senv または --set-environment-variable</p> <p>ENV: ANSIBLE_NAVIGATOR_SET_ENVIRONMENT_VARIABLES</p> <p>設定ファイル:</p> <pre>ansible-navigator: execution-environment: environment-variables: set:</pre>

8.3. 自動化コンテンツナビゲーターの CONFIG サブコマンド設定

以下の表は、自動化コンテンツナビゲーター **config** サブコマンドの各パラメーターと設定オプションを示しています。

表8.2 自動化コンテンツナビゲーターconfig サブコマンドパラメーター設定テーブル

パラメーター	説明	オプションの設定
config	Ansible 設定ファイルのパスを指定します。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: -c または --config</p> <p>ENV: ANSIBLE_CONFIG</p> <p>設定ファイル:</p> <pre>ansible-navigator: ansible: config:</pre>

パラメーター	説明	オプションの設定
help-config	stdout モードの ansible-config コマンドのヘルプオプション。	選択肢:* True または False デフォルト: False CLI: --hc または --help-config ENV: ANSIBLE_NAVIGATOR_HELP_CONFIG 設定ファイル: ansible-navigator: help-config:

8.4. 自動化コンテンツナビゲーターの doc サブコマンド設定

以下の表は、自動化コンテンツナビゲーターの **doc** サブコマンド用の各パラメーターおよび設定オプションを示しています。

表8.3 自動化コンテンツナビゲーターの doc サブコマンドパラメーター設定の表

パラメーター	説明	オプションの設定
help-doc	stdout モードの ansible-doc コマンドのヘルプオプション。	選択肢: True または False デフォルト: False CLI: --hd または --help-doc ENV: ANSIBLE_NAVIGATOR_HELP_DOC 設定ファイル: ansible-navigator: help-doc:

パラメーター	説明	オプションの設定
plugin-name	プラグイン名を指定します。	デフォルト: デフォルト値設定なし CLI: 位置 ENV: ANSIBLE_NAVIGATOR_PLUGIN_NAME 設定ファイル: <pre>ansible-navigator: documentation: plugin: name:</pre>
plugin-type	プラグインタイプを指定します。	選択肢: become、cache、callback、cli conf、connection、httpapi 、inventory、lookup、module、netconf、shell、strategy 、または vars デフォルト: module CLI: -t または ----type ENV: ANSIBLE_NAVIGATOR_PLUGIN_TYPE 設定ファイル: <pre>ansible-navigator: documentation: plugin: type:</pre>

8.5. 自動化コンテンツナビゲーターの INVENTORY サブコマンド設定

以下の表は、自動化コンテンツナビゲーターの **inventory** サブコマンドの各パラメーターおよび設定オプションを示しています。

表8.4 自動化コンテンツナビゲーターの **inventory** サブコマンド設定の表

パラメーター	説明	オプションの設定
--------	----	----------

パラメーター	説明	オプションの設定
help-inventory	stdout モードの ansible-inventory コマンドのヘルプオプション。	<p>選択肢: True または False</p> <p>デフォルト: False</p> <p>CLI: --hi または --help-inventory</p> <p>ENV: ANSIBLE_NAVIGATOR_INVENTORY_DOC</p> <p>設定ファイル:</p> <pre>ansible-navigator: help-inventory:</pre>
inventory	インベントリーファイルパスまたはコンマ区切りホスト一覧を指定します。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --i または --inventory</p> <p>ENV: ANSIBLE_NAVIGATOR_INVENTORIES</p> <p>設定ファイル:</p> <pre>ansible-navigator: inventories:</pre>
inventory-column	インベントリービューに表示されるホスト属性を指定します。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --ic または --inventory-column</p> <p>ENV: * ANSIBLE_NAVIGATOR_INVENTORY_COLUMNS設定ファイル:</p> <pre>ansible-navigator: inventory-columns:</pre>

8.6. 自動化コンテンツナビゲーターの REPLAY サブコマンド設定

以下の表は、自動化コンテンツナビゲーターの **replay** サブコマンド用の各パラメーターおよび設定オプションを示しています。

表8.5 自動化コンテンツナビゲーターの **replay** サブコマンドパラメーター設定の表

パラメーター	説明	オプションの設定
playbook-artifact-replay	再生する Playbook アーティファクトのパスを指定します。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: 位置</p> <p>ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_REPLAY</p> <p>設定ファイル:</p> <pre> ansible-navigator: playbook-artifact: replay: </pre>

8.7. 自動化コンテンツナビゲーターの RUN サブコマンド設定

以下の表は、自動化コンテンツナビゲーターの **run** サブコマンド用の各パラメーターおよび設定オプションを示しています。

表8.6 自動化コンテンツナビゲーターの run サブコマンドパラメーター設定の表

パラメーター	説明	オプションの設定
playbook-artifact-replay	再生する Playbook アーティファクトのパスを指定します。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: 位置</p> <p>ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_REPLAY</p> <p>設定ファイル:</p> <pre> ansible-navigator: playbook-artifact: replay: </pre>

パラメーター	説明	オプションの設定
help-playbook	stdout モードの ansible-playbook コマンドのヘルプオプション。	<p>選択肢: True または False</p> <p>デフォルト: False</p> <p>CLI: --hp または --help-playbook</p> <p>ENV: ANSIBLE_NAVIGATOR_HELP_PLAYBOOK</p> <p>設定ファイル:</p> <pre>ansible-navigator: help-playbook:</pre>
inventory	インベントリーファイルパスまたはコンマ区切りホスト一覧を指定します。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --i または --inventory</p> <p>ENV: ANSIBLE_NAVIGATOR_INVENTORIES</p> <p>設定ファイル:</p> <pre>ansible-navigator: inventories:</pre>
inventory-column	インベントリービューに表示されるホスト属性を指定します。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: --ic または --inventory-column</p> <p>ENV: * ANSIBLE_NAVIGATOR_INVENTORY_COLUMNS 設定ファイル:</p> <pre>ansible-navigator: inventory-columns:</pre>

パラメーター	説明	オプションの設定
Playbook	Playbook 名を指定します。	<p>デフォルト: デフォルト値設定なし</p> <p>CLI: 位置</p> <p>ENV: ANSIBLE_NAVIGATOR_PLAYBOOK</p> <p>設定ファイル:</p> <pre>ansible-navigator: ansible: playbook:</pre>
playbook-artifact-enable	完了した Playbook のアーティファクトの作成を有効または無効にします。注記: Playbook でユーザー入力が必要な場合には、 --mode stdout と互換性がありません。	<p>選択肢: True または False</p> <p>デフォルト: True</p> <p>CLI: --pae または --playbook-artifact-enable ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_ENABLE 設定ファイル:</p> <pre>ansible-navigator: playbook-artifact: enable:</pre>
playbook-artifact-save-as	完了した Playbook から作成したアーティファクトの名前を指定します。	<p>Default: {playbook_dir}/{playbook_name}-artifact-{ts_utc}.json</p> <p>CLI: --pas または --playbook-artifact-save-as</p> <p>ENV: ANSIBLE_NAVIGATOR_PLAYBOOK_ARTIFACT_SAVE_AS</p> <p>設定ファイル:</p> <pre>ansible-navigator: playbook-artifact: save-as:</pre>

第9章 自動化コンテンツナビゲーターを使用した ANSIBLE コンテンツのトラブルシューティング

コンテンツ作成者は、自動化コンテンツナビゲーターで Ansible コンテンツ (コレクション、自動化実行環境、および Playbook) をトラブルシューティングしたり、Playbook を対話形式でトラブルシューティングしたりできます。自動化実行環境の内外で結果を比較し、問題をトラブルシューティングすることもできます。

9.1. 自動化コンテンツナビゲーターアーティファクトファイルを使用した PLAYBOOK の結果の確認

Automation コンテンツナビゲーターは、JSON アーティファクトファイルに Playbook 実行の結果を保存します。このファイルを使用して、Playbook の結果を他の人と共有したり、セキュリティやコンプライアンスの理由で保存したり、後で確認してトラブルシューティングしたりできます。Playbook の実行を確認するためにアーティファクトファイルのみが必要になります。Playbook 自体またはイベントリーアクセスにアクセスする必要はありません。

前提条件

- Playbook 実行からの自動化コンテンツナビゲーターアーティファクトの JSON ファイル。

手順

- アーティファクトファイルによる自動化コンテンツナビゲーターの開始

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

1. Playbook の実行時に一致する Playbook の結果を確認します。

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

Playbook を実行した後と同じように、プレイとタスクの横に番号を入力して、それぞれにステップインして結果を確認できるようになりました。

関連情報

- [ansible-playbook](#).
- [Ansible Playbook の概要](#)

9.2. 自動化コンテンツナビゲーターでよくある質問 (FAQ)

次の自動化コンテンツナビゲーターの FAQ を、お使いの環境で発生している問題のトラブルシューティングに役立ててください。

自動化実行環境を使用する場合、ansible.cfg ファイルはどこに置く必要がありますか。

ansible.cfg を使用する最も簡単な方法は、Playbook に隣接するプロジェクトディレクトリーに置くことです。Playbook ディレクトリーは実行環境に自動的にマウントされ、**ansible.cfg** ファイルが表示されます。**ansible.cfg** ファイルが別のディレクトリーにある場合は、**ANSIBLE_CONFIG** 変数とカスタムボリュームマウントとして指定されたディレクトリーを設定する必要があります。**execution-environment-volume-mounts** については、自動化コンテンツナビゲーターの設定を参照してください。

自動化実行環境を使用しない場合の **ansible.cfg** ファイルの移動先は、Ansible が、自動化実行環境を使用しない場合の一般的な場所で **ansible.cfg** を検索します。詳細は、「[Ansible 構成設定](#)」を参照してください。

自動化実行環境を使用する場合、Ansible コレクションはどこに置く必要がありますか。

Ansible コレクションを作成する最も簡単な場所は、プロジェクトディレクトリー、Playbook に隣接するコレクションディレクトリーです (たとえば、**ansible-galaxy collections install ansible.utils -p ./collections**)。Playbook ディレクトリーは自動化実行環境に自動的にマウントされ、自動化コンテンツナビゲーターはそこでコレクションを見つけます。または、Ansible Builder を使用して、コレクションを自動化実行環境に構築する方法もあります。自動化コントローラーは Playbook に隣接するコレクションディレクトリーをサポートしているため、これはコンテンツ作成者が実稼働用の Playbook を作成するのに役立ちます。コレクションが別のディレクトリーにある場合は、**ANSIBLE_COLLECTIONS_PATHS** 変数を設定して、そのディレクトリーのカスタムボリュームマウントを設定します。**execution-environment-volume-mounts** の場合は「[自動化コンテンツナビゲーターの一般設定](#)」を参照してください。

自動化実行環境を使用していない場合、Ansible コレクションはどこに置く必要がありますか。

自動化実行環境を使用しない場合は、Ansible がコレクションのデフォルト場所を検索します。『[Ansible Collections ユーザーガイド](#)』を参照してください。

vars_prompt または pause/prompt が使用されたときに Playbook がハングするのはなぜですか。

デフォルトでは、自動化コンテンツナビゲーターは、自動化コントローラーが Playbook を実行するのと同じ方法で Playbook を実行します。これは、実稼働環境の準備が整う可能性のあるコンテンツ作成者 Playbook を支援するために実行されました。**vars_prompt** または **pause/prompt** の使用を回避できない場合は、**playbook-artifact** の作成を無効にすると、自動化コンテンツナビゲーターは **ansible-playbook** と互換性のある方法で Playbook を実行し、ユーザーの対話を可能にします。

端末コンテンツナビゲーターが端末の色を変更したり、見栄えが悪くなるのはなぜですか。

自動化コンテンツナビゲーターは、OSC4 互換性の端末にクエリーを実行します。OSC4、10、11、104、110、111 は、端末が色変更をサポートすることを示し、元に戻します。端末が機能を誤って表示している可能性があります。OSC4 検出は、**--osc4 false** を設定して無効にできます。環境変数または設定ファイルでこれを処理する方法は、「[自動化コンテンツナビゲーターの一般設定](#)」を参照してください。

自動化コンテンツナビゲーターで使用される色を変更するにはどうすればよいですか。

--osc4 false を使用して、自動化コンテンツナビゲーターを強制的に端末定義した色を使用するようにします。環境変数または設定ファイルでこれを処理する方法は、「[自動化コンテンツナビゲーターの一般設定](#)」を参照してください。

playbook ディレクトリーにあるすべての site-artifact-2021-06-02T16:02:33.911259+00:00.json ファイルは何に使用しますか。

自動化コンテンツナビゲーターは、すべての Playbook の実行に対して Playbook アーティファクトを作成します。これらは、自動化が完了した後の結果の確認、同僚との共有とトラブルシューティング、またはコンプライアンスや変更管理の目的での維持に役立ちます。Playbook アーティファクトファイルには、すべてのプレイおよびタスク、Playbook 実行の標準出力 **stdout** に関する詳細情報が含まれます。自動化コンテンツナビゲーターセッション中に **ansible-navigator replay <filename>** または **:replay <filename>** を使用して Playbook アーティファクトを確認できます。すべての Playbook アーティファクトは、希望のビューに応じて、**--mode stdout** および **--mode interactive** の両方で確認することができます。Playbook のアーティファクトの記述およびデフォルトのファイルの命名規則を無効にすることができます。環境変数または設定ファイルでこれを処理する方法は、「[自動化コンテンツナビゲーターの一般設定](#)」を参照してください。

:open を使用すると vi が開くのはなぜですか。

自動化コンテンツナビゲーターは、デフォルトのエディターで端末に表示されるものを開きます。デフォルトは **vi +{line_number} {filename}**、または **EDITOR** 環境変数の現在の値のいずれかに設定されます。これは、エディターがコンソールベースまたは端末ベースであるかを示す **editor-console** 設定です。役立つ可能性のある代替設定の例を以下に示します。

```
# 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
```

どの順番で設定が適用されますか。

自動化コンテンツナビゲーター設定システムは、さまざまなソースから設定をプルするため、以下の順序で階層的に適用します (最後に適用された変更が最も一般的なものになります)。

1. デフォルトの内部値
2. 設定ファイルの値
3. 環境変数からの値
4. コマンドラインで指定されたフラグおよび引数
5. テキストベースのユーザーインターフェース内で : コマンドを実行している間

何かが機能しませんでした。どうすればトラブルシューティングできますか。

自動化コンテンツナビゲータには、適切なログメッセージがあります。 **--log-level debug** を使用して、 **debug** ログを有効にできます。バグを見つけたと思われる場合は、問題をログに記録し、ログファイルから詳細情報を追加してください。