Red Hat Ansible Automation Platform 2.4

Automation Hub でのコンテンツの管理

Automation Hub でのコレクション、コンテンツ、リポジトリーの作成および管理

Last Updated: 2024-08-16

Red Hat Ansible Automation Platform 2.4 Automation Hub でのコンテンツの管理

Automation Hub でのコレクション、コンテンツ、リポジトリーの作成および管理

法律上の通知

概要

このガイドでは、Automation Hub でコンテンツを作成、編集、削除、移動する方法を説明します。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)

このドキュメントの改善に関するご意見がある場合や、エラーを発見した場合は、https://access.redhat.com から Technical Support チームに連絡してください。

第1章 AUTOMATION HUB の RED HAT 認定済み、検証済み、および ANSIBLE GALAXY コンテンツ

Ansible Certified Content Collections は、Red Hat Ansible Automation Platform のサブスクリプションに含まれています。Red Hat Ansible コンテンツには、Ansible Certified Content Collections と Ansible 検証済みコンテンツの 2 種類のコンテンツが含まれます。Ansible Automation Hub を使用すると、あらゆる形式の Ansible コンテンツにアクセスし、独自のコレクションセットをキュレートできます。

Red Hat Ansible コンテンツには、次の 2 種類のコンテンツが含まれます。

- Ansible Certified Content Collections
- Ansible 検証済みコンテンツコレクション

Ansible の検証済みコレクションは、プラットフォームインストーラーを通じて Private Automation Hub で利用できます。バンドルされたインストーラーを使用して Red Hat Ansible Automation Platform をダウンロードすると、インベントリーの一部として Private Automation Hub を有効にした場合に限り、検証済みのコンテンツがデフォルトで Private Automation Hub に事前入力されます。

バンドルインストーラーを使用していない場合は、Red Hat が提供する Ansible Playbook を使用して検証済みのコンテンツをインストールできます。詳細は、Ansible 検証済みコンテンツ を参照してください。

このコレクションは、パッケージをダウンロードして手動で更新できます。

Ansible コレクションを認定する理由

Ansible 認定プログラムにより、Red Hat とエコシステムパートナーの間で Red Hat Ansible Certified Content のサポートについて共同で対応できるようになりました。Ansible および認定パートナーのコンテンツで問題が発生しているエンドカスタマーは、情報のリクエストや Red Hat の問題などのサポートチケットを作成し、Red Hat およびエコシステムパートナーによってチケットが解決されることを期待できます。

Red Hat は、認定パートナー企業が市場の認知度を高め、需要を創出し、共同販売を行えるような市場参入の利益を提供します。

Red Hat Ansible Certified Content Collections は、Ansible Automation Hub (サブスクリプションが必要) を通じて配布されます。これは、共同でサポートされる Ansible コンテンツの集中型リポジトリーです。認定パートナーとして、コレクションを Ansible Automation Hub にパブリッシュすることで、エンドユーザーは、信頼できるオートメーションコンテンツが、周知されているサポートライフサイクルで、実稼働環境で使用される方法を管理できます。

ソリューションの認定を開始する方法の詳細は、Red Hat Partner Connect を参照してください。

コレクションの認定を受ける方法

コレクションを認定する手順については、Red Hat Partner Connect の Ansible 認定ポリシーガイドを参照してください。

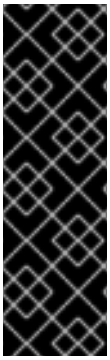
Certified Collections に関する共同サポート契約の仕組み

お客様が認定コレクションに関して Red Hat サポートチームに問題を提起した場合は、Red Hat サポートが問題を評価し、問題が Ansible または Ansible の使用に存在するかどうかを確認します。また、問題が認定コレクションにあるかどうかも確認します。認定コレクションに問題がある場合は、サポートチームが、TSANet などの合意されたツールを通じて、認定コレクションのベンダー企業側所有者に問題を転送します。

Ansible ロールのみを含むコレクションを作成して認定できるか

ロールのみを含むコレクションを作成および認定できます。現在のテスト要件は、モジュールを含むコレクションに集中しており、ロールのみを含むコレクションをテストするための追加のリソースが現在進行中です。詳細は、ansiblepartners@redhat.com までお問い合わせください。

1.1. AUTOMATION HUB での ANSIBLE CONTENT COLLECTIONS の同期



重要

2.4 リリースでは、引き続きコンテンツを同期できますが、同期リストは非推奨となり、今後のバージョンで削除される予定です。

コンテンツを同期するために、rh-certified リモートから手動で作成した要件ファイルをアップロードできるようになりました。

リモートは、外部コレクションソースからカスタムリポジトリーにコンテンツを同期できる設定です。

Ansible Automation Hub を使用して、同期リストまたは要件ファイルを作成することで、関連する Red Hat Ansible Certified Content Collections をユーザーに配布できます。要件ファイルの使用の詳細は、**Using Ansible collections** ガイドの Install multiple collections with a requirements file を参照してください。

1.1.1. Red Hat Ansible Certified Content Collections の同期リストの説明

同期リストは、組織管理者が集めてキュレートした Red Hat Certified Collections のグループです。同期リストは、ローカルの Ansible Automation Hub と同期します。同期リストを使用して、必要なコンテンツのみを管理し、不要なコレクションを除外します。console.redhat.com の Red Hat コンテンツの一部として利用可能なコンテンツから同期リストを設計および管理します。

各同期リストには独自の一意のリポジトリー URL があります。これを使用して Automation Hub 内のコンテンツのリモートソースとして指定できます。API トークンを使用して各同期リストにセキュアにアクセスします。

1.1.2. Red Hat Ansible Certified Content Collections の同期リストの作成

console.redhat.com の Ansible Automation Hub で、精選された Red Hat Ansible Certified Content の同期リストを作成できます。synclist リポジトリーは、Automation Hub ナビゲーションパネルの **Collection → Repositories** の下にあり、Ansible Certified Content Collections 内でコンテンツを管理するたびに更新されます。

すべての Ansible Certified Content Collections は、初期の組織の同期リストにデフォルトで含まれています。

前提条件

- 有効な Ansible Automation Platform サブスクリプションがある。
- consoleredhat.com の組織管理者権限がある。
- 次のドメイン名は、ファイアウォールまたはプロキシーの許可リストの一部です。これらは、Automation Hub または Galaxy サーバーに正常に接続し、コレクションをダウンロードするために必要です。
    - **galaxy.ansible.com**

- **cloud.redhat.com**
- **console.redhat.com**
- **sso.redhat.com**

- Ansible Automation Hub のリソースは、Amazon Simple Storage に保存されます。次のドメイン名が許可リストに含まれている必要があります。
  - **automation-hub-prd.s3.us-east-2.amazonaws.com**
  - **ansible-galaxy.s3.amazonaws.com**
- 自己署名証明書または Red Hat ドメインを使用する場合に SSL インスペクションが無効になっている。

手順

1. **console.redhat.com** にログインします。
2. **Automation Hub → Collections** に移動します。
3. 各コレクションのトグルスイッチを設定して、同期リストから除外または追加します。
4. リモートリポジトリーの同期を開始するには、Automation Hub に移動し、**Collection → Repositories** を選択します。
5. **More Actions** アイコン ⋮ をクリックし、**Sync** を選択して、Private Automation Hub へのリモートリポジトリーの同期を開始します。
6. オプション: リモートリポジトリーがすでに設定されている場合は、Red Hat Ansible Certified Content Collections を Private Automation Hub に手動で同期することにより、ローカルユーザーに提供したコレクションコンテンツを更新します。

1.2. コンテンツを同期するための ANSIBLE AUTOMATION HUB リモートリポジトリーの設定

リモート設定を使用して、**console.redhat.com** でホストされている Ansible Certified Content Collections または Ansible Galaxy のコレクションと同期するように Private Automation Hub を設定します。

重要

2.4 リリースでは、引き続きコンテンツを同期できますが、同期リストは非推奨となり、今後のバージョンで削除される予定です。

コンテンツを同期するために、rh-certified リモートから手動で作成した要件ファイルをアップロードできるようになりました。

リモートは、外部コレクションソースからカスタムリポジトリーにコンテンツを同期できる設定です。

Ansible Galaxy と Ansible Automation Hub の相違点
Ansible Galaxy に公開されたコレクションは、Ansible コミュニティーによって公開された最新のコンテンツです。これらのコレクションに対する共同でのサポート対応は行われていません。Ansible

Galaxy は、コンテンツにアクセスする Ansible コミュニティーを対象とした、推奨されるフロントエンドのディレクトリーです。

Ansible Automation Hub に公開されたコレクションは、Red Hat と特定パートナーの共同のお客様を対象としたものです。Ansible Automation Hub のコレクションにアクセスしてダウンロードするには、Ansible サブスクリプションが必要です。認定コレクションとは、Red Hat とパートナーが適所に戦略的関係を築いており、共同してお客様をサポートする準備ができており、そのために追加のテストと検証が行われた可能性があることを意味します。

Ansible Galaxy で名前空間をリクエストする方法
Ansible Galaxy GitHub のイシューを通じて名前空間をリクエストするには、次の手順に従います。

- ansiblepartners@redhat.com にメールを送信します。
- Ansible Galaxy へのサインアップに使用した GitHub ユーザー名を記載します。

システムが検証できるように、少なくとも 1 回はログインする必要があります。

ユーザーが名前空間の管理者として追加されると、セルフサービスのプロセスを使用してさらに管理者を追加できます。

Ansible Galaxy 名前空間の命名制限
コレクションの名前空間は、Python モジュール名の規則に従う必要があります。つまり、コレクションにはすべて小文字の短い名前を付ける必要があります。読みやすさが向上する場合は、コレクション名にアンダースコアを使用できます。

1.2.1. リモート設定を作成する理由

Collections → Remotes にある各リモート設定は、**community** および **rh-certified** リポジトリーの両方に、リポジトリーが **最後に更新された** 日時に関する情報を提供します。**Collection → Repositories** ページに含まれる **Edit** と **Sync** 機能を使用して、いつでも Ansible Automation Hub に新しいコンテンツを追加できます。

1.2.2. Red Hat Certified Collection の同期 URL と API トークンの取得

組織によってキュレートされた Ansible Certified Content Collections を **console.redhat.com** から Private Automation Hub に同期できます。API トークンは、コンテンツを保護するために使用されるシークレットトークンです。

前提条件

- console.redhat.com で同期リストを作成するための組織管理者権限がある。

手順

1. 組織管理者として **console.redhat.com** にログインします。
2. **Automation Hub → Connect to Hub** に移動します。
3. **Offline token** で、**Load token** をクリックします。
4. **Copy to clipboard** をクリックし、API トークンをコピーします。
5. API トークンをファイルに貼り付け、安全な場所に保存します。

1.2.3. rh-certified リモートリポジトリーの設定および Red Hat Ansible Certified Content Collection の同期

rh-certified リモートリポジトリーを編集して、console.redhat.com でホストされている Automation Hub から Private Automation Hub にコレクションを同期できます。デフォルトでは、Private Automation Hub **rh-certified** リポジトリーには、Ansible Certified Content Collections のグループ全体の URL が含まれています。

組織で指定されたコレクションのみを使用するには、Private Automation Hub 管理者は、手動で作成した要件ファイルを **rh-certified** リモートからアップロードできます。

要件ファイルの使用の詳細は、**Using Ansible collections** ガイドの Install multiple collections with a requirements file を参照してください。

要件ファイルに、**A**、**B**、および **C** コレクションがあり、使用する console.redhat.com に新しいコレクション **X** を追加する場合は、Private Automation Hub の要件ファイルに **X** を追加して同期する必要があります。

前提条件

- 有効な **Ansible リポジトリーコンテンツの変更** 権限を持っている。権限に関する詳細は、Private Automation Hub のユーザーアクセスの設定 を参照してください。
- console.redhat.com で Automation Hub がホストするサービスから同期 URL および API トークンを取得している。
- ポート 443 へのアクセスを設定している。これは、認定されたコレクションを同期するために必要です。詳細は、「Red Hat Ansible Automation Platform 計画ガイド」の ネットワークポートとプロトコル の章にある Automation Hub の表を参照してください。

手順

1. Private Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Remotes** を選択します。
3. **rh-certified** リモートリポジトリーで、**More Actions** アイコン ⋮ をクリックし、**Edit** をクリックします。
4. **URL** フィールドに、**Sync URL** を貼り付けます。
5. **Token** フィールドに、console.redhat.com から取得したトークンを貼り付けます。
6. **Save** をクリックします。
   console.redhat.com の組織の同期リストと Private Automation Hub の間でコレクションを同期できるようになりました。
7. **More Actions** アイコン ⋮ をクリックし、**Sync** を選択します。

検証

Sync status 通知が更新され、Red Hat Certified Content Collections の同期が完了したことが通知されます。

- コレクションコンテンツのドロップダウンリストから **Red Hat Certified** を選択し、コレクションコンテンツが正常に同期されたことを確認します。

1.2.4. コミュニティーリモートリポジトリーの設定および Ansible Galaxy コレクションの同期

コミュニティー のリモートリポジトリーを編集して、選択したコレクションを Ansible Galaxy から Private Automation Hub に同期できます。デフォルトでは、Private Automation Hub コミュニティーリポジトリーは **galaxy.ansible.com/api/** に送信されます。

前提条件

- **Modify Ansible repo content**権限がある。権限に関する詳細は、Private Automation Hub のユーザーアクセスの設定 を参照してください。
- 次の例のように、Ansible Galaxy から同期するコレクションを識別する **requirements.yml** ファイルがある。

  requirements.yml の例

```
collections:
  # Install a collection from Ansible Galaxy.
  - name: community.aws
    version: 5.2.0
    source: https://galaxy.ansible.com
```

手順

1. Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Remotes** を選択します。
3. **Community** リモートで、**More Actions** アイコン ⋮ をクリックし、**Edit** を選択します。
4. **YAML requirements** フィールドで、**Browse** をクリックし、ローカルマシン上の **requirements.yml** ファイルを見つけます。
5. **Save** をクリックします。
   requirements.yml ファイルで識別されたコレクションを、Ansible Galaxy から Private Automation Hub に同期できるようになりました。
6. **More Actions** アイコン ⋮ をクリックし、**Sync** を選択して、Ansible Galaxy および Ansible Automation Hub からコレクションを同期します。

検証

同期ステータス 通知が更新され、Ansible Automation Hub への Ansible Galaxy コレクション同期の完了または失敗が通知されます。

- コレクションのコンテンツドロップダウンリストから **Community** を選択して、同期が成功したことを確認します。

1.2.5. プロキシー設定

Private Automation Hub がネットワークプロキシーの背後にある場合は、リモートでプロキシーを設定して、ローカルネットワーク外にあるコンテンツを同期できます。

前提条件

前提条件

- 有効な **Ansible リポジトリーコンテンツの変更** 権限を持っている。権限に関する詳細は、Private Automation Hub のユーザーアクセスの設定 を参照してください。
- ローカルネットワーク管理者からのプロキシー URL および認証情報を持っている。

手順

1. Private Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Remotes** を選択します。
3. **rh-certified** または **Community** リモートのいずれかで、**More Actions** アイコン ⋮ をクリックし、**Edit** を選択します。
4. **Show advanced options** ドロップダウンメニューを展開します。
5. プロキシー URL、プロキシーユーザー名、およびプロキシーパスワードを適切なフィールドに入力します。
6. **Save** をクリックします。

1.3. PRIVATE AUTOMATION HUB のコレクションおよびコンテンツ署名

組織の自動化管理者は、組織内の異なるグループから Ansible コンテンツコレクションの署名および公開用に Private Automation Hub を設定できます。

セキュリティーを強化するために、自動化作成者は Ansible-Galaxy CLI を設定してこのコレクションを検証し、Automation Hub へのアップロード後に変更されていないことを確認できます。

1.3.1. Private Automation Hub でのコンテンツ署名の設定

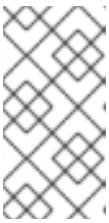
Ansible Certified Content Collections に正常に署名して公開するには、署名する Private Automation Hub を設定する必要があります。

前提条件

- GnuPG キーペアがセキュアに設定され、組織で管理されている。
- 公開鍵と秘密鍵のペアに、Private Automation Hub でコンテンツ署名を設定するのに適切なアクセス権がある。

手順

1. ファイル名のみを受け入れる署名スクリプトを作成します。



注記

このスクリプトは署名サービスとして機能し、**PULP_SIGNING_KEY_FINGERPRINT** 環境変数で指定された鍵を使用して、そのファイルの ASCII アーマー形式の **gpg** デタッチ署名を生成する必要があります。

スクリプトは、次の形式で JSON 構造を出力します。

```
{"file": "filename", "signature": "filename.asc"}
```

すべてのファイル名は、現在の作業ディレクトリー内の相対パスです。ファイル名は、デタッチ署名でも同じにする必要があります。

以下に例を示します。

次のスクリプトはコンテンツの署名を生成します。

```
#!/usr/bin/env bash

FILE_PATH=$1
SIGNATURE_PATH="$1.asc"

ADMIN_ID="$PULP_SIGNING_KEY_FINGERPRINT"
PASSWORD="password"

# Create a detached signature
gpg --quiet --batch --pinentry-mode loopback --yes --passphrase \
  $PASSWORD --homedir ~/.gnupg/ --detach-sign --default-key $ADMIN_ID \
  --armor --output $SIGNATURE_PATH $FILE_PATH

# Check the exit status
STATUS=$?
if [ $STATUS -eq 0 ]; then
  echo {\"file\": \"$FILE_PATH\", \"signature\": \"$SIGNATURE_PATH\"}
else
  exit $STATUS
fi
```

署名を有効にして Private Automation Hub を Ansible Automation Platform クラスターにデプロイすると、新しい UI が追加されたことがコレクションに表示されます。

2. **automationhub_*** で始まるオプションについては、Ansible Automation Platform インストーラーのインベントリーファイルを確認してください。

```
[all:vars]
.
.
.
automationhub_create_default_collection_signing_service = True
automationhub_auto_sign_collections = True
automationhub_require_content_approval = True
automationhub_collection_signing_service_key = /abs/path/to/galaxy_signing_service.gpg
automationhub_collection_signing_service_script = /abs/path/to/collection_signing.sh
```

2 つの新しいキー (**automationhub_auto_sign_collections** および **automationhub_require_content_approval**) は、コレクションが Private Automation Hub にアップロードされた後に署名および承認される必要があることを示します。

1.3.2. Private Automation Hub でのコンテンツ署名サービスの使用

Private Automation Hub でコンテンツ署名を設定した後、新しいコレクションに手動で署名したり、既存の署名を新しい署名に置き換えたりできます。この署名は、ユーザーが特定のコレクションをダウン

ロードする際に、そのコレクションがユーザー向けであり、認定後に変更されていないことを示すものとなります。

次のシナリオでは、Private Automation Hub でコンテンツ署名を使用できます。

- システムに自動署名が設定されていないため、手動署名プロセスを使用してコレクションに署名する必要がある場合。
- 自動的に設定されたコレクションの現在の署名が壊れているため、新しい署名が必要な場合。
- 以前に署名されたコンテンツに追加の署名が必要な場合。
- コレクションで署名をローテーションする必要がある場合。

手順

1. Ansible Automation Platform にログインします。
2. ナビゲーションパネルから、**Collections → Approval** を選択します。Approval ダッシュボードが開き、コレクションのリストが表示されます。
3. 署名するコレクションごとに、**Sign and approve** をクリックします。

検証

- 署名および手動で承認されたコレクションが **Collections** タブに表示されていることを確認します。

1.3.3. 署名公開鍵のダウンロード

コレクションに署名して承認したら、Automation Hub UI から署名公開キーをダウンロードします。公開鍵は、ローカルシステムのキーリングに追加する前にダウンロードする必要があります。

手順

1. Automation Hub にログインします。
2. ナビゲーションパネルから、**Signature Keys** を選択します。署名キーダッシュボードには、複数のキー (コレクションとコンテナーイメージ) のリストが表示されます。
   - コレクションを確認するには、**collections-** で始まるキーをダウンロードします。
   - コンテナーイメージを確認するには、**container-** で始まるキーをダウンロードします。
3. 次のいずれかの方法を選択して、公開鍵をダウンロードします。
   - メニューアイコンを選択し、**Download Key** をクリックして公開キーをダウンロードします。
   - リストから公開鍵を選択し、**Copy to clipboard** アイコンをクリックします。
   - **Public Key** タブの下のドロップダウンメニューをクリックし、公開鍵ブロック全体をコピーします。

コピーした公開鍵を使用して、インストールするコンテンツコレクションを確認します。

1.3.4. コレクションを検証するための Ansible-Galaxy CLI の設定

Ansible-Galaxy CLI を設定して、コレクションを検証することができます。これにより、ダウンロードしたコレクションが組織によって承認されたものであり、Automation Hub へのアップロード後に変更されていないことを確認できます。

コレクションが Automation Hub によって署名されている場合、サーバーは、コレクションのコンテンツの検証に **MANIFEST.json** を使用する前に、その信頼性を検証するために、ASCII アーマー形式の GPG デタッチ署名を提供します。**ansible-galaxy** の キーリングを設定する か、**--keyring** オプションでパスを指定して、署名検証をオプトインする必要があります。

前提条件

- 署名付きコレクションが Automation Hub で署名を検証するために利用できる。
- 認定コレクションが組織内の承認済みのロールによって署名できる。
- 検証用の公開鍵がローカルシステムキーリングに追加されている。

手順

1. **ansible-galaxy** で使用するデフォルト以外のキーリングに公開鍵をインポートするには、以下のコマンドを実行します。

```
gpg --import --no-default-keyring --keyring ~/.ansible/pubring.kbx my-public-key.asc
```

注記

Automation Hub が提供する署名のほかに、署名ソースは要件ファイルとコマンドラインで指定することもできます。署名ソースは URI である必要があります。

2. 追加の署名を使用して CLI で指定されたコレクション名を確認するには、次のコマンドを実行します。

```
ansible-galaxy collection install namespace.collection
--signature https://examplehost.com/detached_signature.asc
--signature file:///path/to/local/detached_signature.asc --keyring ~/.ansible/pubring.kbx
```

このオプションを複数回使用して、複数の署名を指定できます。

3. 以下の例のように、要件ファイルのコレクションに、コレクションの署名キーの後に追加の署名ソースが表示されていることを確認します。

```
# requirements.yml
collections:
  - name: ns.coll
    version: 1.0.0
    signatures:
      - https://examplehost.com/detached_signature.asc
      - file:///path/to/local/detached_signature.asc

ansible-galaxy collection verify -r requirements.yml --keyring ~/.ansible/pubring.kbx
```

Automation Hub からコレクションをインストールすると、サーバーが提供する署名はインストールされたコレクションと共に保存され、コレクションの信頼性を検証します。

4. (オプション) Ansible Galaxy サーバーにクエリーを実行せずにコレクションの内部整合性を再度確認する必要がある場合は、**--offline** オプションを使用して、以前に使用したのと同じコマンドを実行します。

コレクションの命名に関する推奨事項
company_name.product 形式でコレクションを作成します。この形式は、複数の製品が会社の名前空間の下に異なるコレクションを持つことができることを示しています。

Ansible Automation Hub で名前空間を取得する方法
デフォルトでは、Ansible Galaxy で使用される名前空間は、Ansible パートナーチームによって Ansible Automation Hub でも使用されます。質問や説明については、ansiblepartners@redhat.com までお問い合わせください。

1.4. ANSIBLE 検証済みコンテンツ

Red Hat Ansible Automation Platform には、既存の Red Hat Ansible Certified Content を補完する Ansible 検証済みコンテンツが含まれています。

Ansible 検証済みコンテンツを使用すると、エキスパートが推奨する方法に従って、Red Hat と信頼できるパートナーが提供する各種プラットフォーム上で運用タスクを実行できます。

1.4.1. インストーラーを使用した検証済みコレクションの設定

バンドルインストーラーをダウンロードして実行すると、認定および検証済みのコレクションが自動的にアップロードされます。認定コレクションは、**rh-certified** リポジトリーにアップロードされます。検証済みのコレクションは、**validated** のリポジトリーにアップロードされます。

次の 2 つの変数を使用して、デフォルトの設定に変更できます。

- **automationhub_seed_collections** は、プリロードが有効かどうかを定義するブール値です。
- **automationhub_collection_seed_repository**。この変数を **true** に設定すると、アップロードするコンテンツのタイプを指定できます。指定できる値は **certified** または **validated** です。両方のコンテンツセットが欠落している場合は、アップロードされます。

1.4.2. tarball を使用した検証済みコンテンツのインストール

バンドルインストーラーを使用しない場合は、スタンドアロンの tarball、**ansible-validated-content-bundle-1.tar.gz** を使用できます。また、後でこのスタンドアロンの tarball を使用して、新しい tarball が利用可能になったときに、バンドルインストーラーを再実行することなく、任意の環境で検証済みのコンテンツを更新することもできます。

前提条件

Playbook を実行するには次の変数が必要です。

名前	説明
automationhub_admin_password	管理者のパスワード。

名前	説明
automationhub_api_token	Automation Hub 用に生成された API トークン。
automationhub_main_url	例: **https://automationhub.example.com**
automationhub_require_content_approval	ブール値 (**true** または **false**) これは、Automation Hub のデプロイメント中に使用される値と一致する必要があります。 この変数はインストーラーによって **true** に設定されます。

手順

1. tarball を取得するには、Red Hat Ansible Automation Platform のダウンロード ページに移動し、**Ansible Validated Content** を選択します。
2. コンテンツをアップロードし、変数を定義します (この例では **automationhub_api_token** を使用します)。

```
ansible-playbook collection_seed.yml
-e automationhub_api_token=<api_token>
-e automationhub_main_url=https://automationhub.example.com
-e automationhub_require_content_approval=true
```

注記

automationhub_admin_password または **automationhub_api_token** の両方ではなく、いずれかを使用します。

完了すると、コレクションは Private Automation Hub の検証済みコレクションセクションに表示されます。ユーザーは、Private Automation Hub からコレクションを表示およびダウンロードできるようになりました。

関連情報

Ansible Playbook の実行の詳細は、ansible-playbook を参照してください。

第2章 AUTOMATION HUB でのコレクションの管理

コンテンツ作成者は、Automation Hub の名前空間を使用して、次の目的でコレクションをキュレートおよび管理できます。

- 名前空間をキュレートし、コレクションを Private Automation Hub にアップロードする権限を持つグループを作成する。
- コレクションのエンドユーザーの自動化タスクで役立つように、名前空間に情報とリソースを追加する。
- コレクションを名前空間にアップロードする。
- 名前空間のインポートログを確認して、コレクションのアップロードの成功または失敗と現在の承認ステータスを確認する。

コンテンツの作成方法については、Red Hat Ansible Automation Platform Creator ガイド を参照してください。

2.1. 名前空間を使用した AUTOMATION HUB でのコレクションの管理

名前空間とは、コンテンツコレクションをアップロードおよび公開できる Automation Hub 内の一意の場所です。Automation Hub の名前空間へのアクセスは、そこに表示されるコンテンツと関連情報を管理する権限を持つグループによって管理されます。

Automation Hub の名前空間を使用して、内部での配布と使用のために組織内で開発されたコレクションを整理できます。

名前空間を操作する場合は、コレクションを作成、編集し、名前空間にアップロードする権限を持つグループが必要です。名前空間にアップロードしたコレクションを公開して使用できるようにするには、管理者の承認が必要です。

2.1.1. コンテンツキュレーターのための新しいグループの作成

組織内のコンテンツのキュレーションを支援するために、Private Automation Hub に新しいグループを作成できます。このグループは、Private Automation Hub で公開するために内部で開発したコレクションに役立ちます。

コンテンツ開発者が名前空間を作成し、内部で開発したコレクションを Private Automation Hub にアップロードできるようにするには、先にグループを作成および編集し、必要な権限を割り当てる必要があります。

前提条件

- Private Automation Hub の管理者権限があり、グループを作成できる。

手順

1. Private Automation Hub にログインします。
2. ナビゲーションパネルから **User Access** → **Groups** を選択し、**Create** をクリックします。
3. モーダルのグループの **Name** として **Content Engineering** を入力し、**Create** をクリックします。新しいグループが作成され、**Groups** ページが開きます。

4. **Permissions** タブで、**Edit** をクリックします。
5. **Namespaces** で、**Add Namespace**、**Upload to Namespace**、および **Change Namespace** の権限を追加します。
6. **Save** をクリックします。
   割り当てた権限を使用して新しいグループが作成されます。その後、グループにユーザーを追加できます。
7. **Groups** ページの **Users** タブをクリックします。
8. **Add** をクリックします。
9. ユーザーを選択し、**Add** をクリックします。

2.1.2. 名前空間の作成

名前空間を作成して、コンテンツ開発者が Automation Hub にアップロードするコレクションを整理できます。名前空間の作成時に、その名前空間の所有者として Automation Hub 内のグループを割り当てることができます。

前提条件

- **Add Namespaces** および **Upload to Namespaces** の権限がある。

手順

1. Private Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Namespaces** を選択します。
3. **Create** をクリックし、**namespace name** を入力します。
4. **Namespace owners** のグループを割り当てます。
5. **Create** をクリックします。

これで、コンテンツ開発者が新しい名前空間にコレクションをアップロードして、所有者として割り当てられたグループ内のユーザーにコレクションのアップロードを許可できるようになりました。

2.1.3. 名前空間への情報およびリソースの追加

名前空間に含まれるコレクションに付随する情報を追加し、ユーザーにリソースを提供できます。ロゴおよび説明を追加し、ユーザーを GitHub リポジトリー、案件管理、またはその他のオンラインアセットにリンクします。**Edit resources** タブにマークダウンテキストを入力して、詳細情報を追加することもできます。これは、自動化タスクでコレクションを使用するユーザーに有用です。

前提条件

- **Change Namespaces** の権限がある。

手順

1. Private Automation Hub にログインします。

2. ナビゲーションパネルから、**Collections → Namespaces** を選択します。
3. **More Actions** アイコン ⋮ をクリックし、**Edit namespace** を選択します。
4. **Edit details** タブで、フィールドに情報を入力します。
5. **Edit resources** タブをクリックして、テキストフィールドにマークダウンを入力します。
6. **Save** をクリックします。

これで、コンテンツ開発者が新しい名前空間にコレクションをアップロードしたり、所有者として割り当てられたグループ内のユーザーにコレクションのアップロードを許可したりできるようになりました。

名前空間を作成すると、名前空間にアップロードする権限を持つグループは、承認を受けるためにコレクションの追加を開始できます。名前空間内のコレクションは、承認されると **Published** リポジトリーに表示されます。

2.1.4. コレクションの名前空間へのアップロード

内部で開発されたコレクションを **tar.gz** ファイル形式で Private Automation Hub 名前空間にアップロードし、Automation Hub 管理者によるレビューと承認を受けることができます。承認されると、コレクションは、Automation Hub ユーザーが表示およびダウンロードできる **Published** コンテンツリポジトリーに移動します。

注記

コレクションファイル名は、<my_namespace-my_collection-1.0.0.tar.gz> という形式にしてください。

前提条件

- コレクションをアップロードできる名前空間がある。

手順

1. Private Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Namespaces** を選択し、名前空間を選択します。
3. **Upload collection** をクリックします。
4. **New collection** ダイアログから **Select file** をクリックします。
5. アップロードするコレクションを選択します。
6. **Upload** をクリックします。

My Imports 画面にはテストの概要が表示され、コレクションのアップロードが成功したか失敗したかが通知されます。

2.1.5. 名前空間インポートログの確認

名前空間にアップロードしたコレクションのステータスを確認して、プロセスの成功または失敗を確認できます。

インポートされたコレクション情報には以下が含まれます。

Status
完了または失敗

承認ステータス
承認待ちまたは承認済み

バージョン
アップロードされたコレクションのバージョン

インポートログ
コレクションのインポート中に実行されたアクティビティー

前提条件

- コレクションをアップロードできる名前空間にアクセスできる。

手順

1. Private Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Namespaces** を選択します。
3. 名前空間を選択します。
4. **More Actions** アイコン ⋮ をクリックし、**My imports** を選択します。
5. 検索フィールドを使用するか、リストからインポートされたコレクションを見つけます。
6. インポートされたコレクションをクリックします。
7. コレクションのインポートの詳細を確認し、名前空間内のコレクションのステータスを確認します。

2.1.6. 名前空間の削除

不要な名前空間を削除して、Automation Hub サーバー上のストレージを管理できます。まず、依存関係のあるコレクションが名前空間に含まれていないことを確認する必要があります。

前提条件

- 削除する名前空間に、依存関係のあるコレクションがない。
- **名前空間の削除** 権限がある。

手順

1. Private Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Namespaces** を選択します。
3. 削除する名前空間をクリックします。
4. **More Actions** アイコン ⋮ をクリックしてから、**Delete namespace** をクリックします。

注記

Delete namespace ボタンが無効になっている場合、依存関係のあるコレクションが名前空間に含まれています。この名前空間内のコレクションを確認し、依存関係があれば削除します。詳細は、Automation Hub でのコレクションの削除 を参照してください。

削除した名前空間とその関連コレクションが削除され、名前空間のリストビューから削除されます。

2.2. AUTOMATION HUB での内部コレクションの公開プロセスの管理

Automation Hub を使用して、組織内で開発されたコンテンツコレクションを管理および公開します。コレクションを名前空間にアップロードしてグループ化できます。**Published** コンテンツリポジトリーに表示するには、管理者の承認が必要です。コレクションを公開すると、ユーザーはコレクションにアクセスしてダウンロードして使用できるようになります。

組織の認定基準を満たさない提出済みコレクションは、拒否することができます。

2.2.1. 承認について

ナビゲーションパネルにある **Approval** 機能を使用して、Automation Hub でアップロードされたコレクションを管理できます。

Approval ダッシュボード

デフォルトでは、**Approval** ダッシュボードには、**Needs Review** ステータスのすべてのコレクションが一覧表示されます。ここで **公開済み** リポジトリーに含まれているかどうかを確認できます。

コレクションの詳細表示

バージョン 番号をクリックすると、コレクションの詳細情報を表示できます。

コレクションのフィルタリング

Namespace、**Collection Name**、または **Repository** 別にコレクションをフィルタリングし、コンテンツを見つけ、ステータスを更新します。

2.2.2. 内部公開用のコレクションの承認

内部での公開および使用のために、個々の名前空間にアップロードされたコレクションを承認できます。レビュー待ちのすべてのコレクションは、**Staging** リポジトリーの **Approval** タブの配下にあります。

前提条件

- **Modify Ansible repo content**権限がある。

手順

1. ナビゲーションパネルから、**Collections** → **Approval** を選択します。
   承認を必要とするコレクションのステータスは **Needs review** となっています。
2. 確認するコレクションを選択します。
3. **Version** をクリックし、コレクションの内容を表示します。
4. **Certify** をクリックし、コレクションを承認します。

承認されたコレクションは **Published** リポジトリーに移動し、ユーザーはここでそのコレクションを表示およびダウンロードして使用することができます。

2.2.3. レビュー用にアップロードされたコレクションの拒否

個別の名前空間にアップロードされたコレクションを拒否できます。レビュー待ちのすべてのコレクションは、**Staging** リポジトリーの **Approval** タブの配下にあります。

承認を必要とするコレクションのステータスは **Needs review** となっています。**Version** をクリックし、コレクションの内容を表示します。

前提条件

- **Modify Ansible repo content**権限がある。

手順

1. ナビゲーションパネルから、**Collections → Approval** を選択します。
2. 確認するコレクションを見つけます。
3. **Reject** をクリックしてコレクションを拒否します。

公開を拒否するコレクションは **Rejected** リポジトリーに移動します。

2.3. AUTOMATION HUB によるリポジトリー管理

Automation Hub 管理者は、自動化コンテンツコレクションを作成、編集、削除し、リポジトリー間で移動できます。

2.3.1. Automation Hub のリポジトリーの種類

Automation Hub では、コレクションを検証するかどうかに応じて、次の 2 種類のリポジトリーにコレクションを公開できます。

ステージングリポジトリー

名前空間にアップロードする権限を持つユーザーは、これらのリポジトリーにコレクションを公開できます。これらのリポジトリー内のコレクションは、検索ページでは使用できません。代わりに、管理者が確認できるように承認ダッシュボードに表示されます。ステージングリポジトリーは、**pipeline=staging** ラベルでマークされます。

カスタムリポジトリー

リポジトリーに対する書き込み権限を持つユーザーは、これらのリポジトリーにコレクションを公開できます。カスタムリポジトリーは、すべてのユーザーが表示できるパブリックリポジトリーにすることも、表示権限を持つユーザーのみが表示できるプライベートリポジトリーにすることもできます。これらのリポジトリーは承認ダッシュボードには表示されません。リポジトリー所有者が検索を有効にしている場合、コレクションは検索結果に表示されます。

デフォルトでは、Automation Hub には 1 つのステージングリポジトリーが付属しており、コレクションのアップロードにリポジトリーが指定されていない場合に自動的に使用されます。ユーザーは、リポジトリーの作成 時に新しいステージングリポジトリーを作成できます。

2.3.2. Automation Hub のカスタムリポジトリーの承認パイプライン

Automation Hub では、コレクションを承認して、**pipeline=approved** ラベルが付いている任意のリポジトリーにプロモートできます。デフォルトでは、Automation Hub には承認済みコンテンツ用のリポジトリーが 1 つありますが、リポジトリー作成画面からさらにリポジトリーを追加するオプションもあります。**pipeline=approved** ラベルが付いているリポジトリーに直接公開することはできません。コレクションは、'pipleline=approved' リポジトリーに公開される前に、まずステージングリポジトリーを通過して承認される必要があります。

自動承認

自動承認が有効になっている場合は、ステージングリポジトリーにアップロードしたコレクションが **pipeline=approved** としてマークされたすべてのリポジトリーに自動的にプロモートされます。

承認が必要

自動承認が無効になっている場合、管理者は承認ダッシュボードを表示して、ステージングリポジトリーのいずれかにアップロードされたコレクションを確認できます。**Approve** をクリックすると、承認されたリポジトリーのリストが表示されます。管理者は、このリストから、コンテンツをプロモートする先の 1 つ以上のリポジトリーを選択できます。
承認されたリポジトリーが 1 つしかない場合は、コレクションが自動的にそのリポジトリーにプロモートされ、管理者はリポジトリーを選択するように求められません。

拒否

拒否されたコレクションは、事前にインストールされている拒否されたリポジトリーに自動的に配置されます。

2.3.3. カスタムリポジトリーへのアクセスを制限するロールベースのアクセス制御

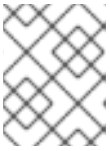
ロールベースのアクセス制御 (RBAC) を使用して、ユーザーのロールに基づいてアクセス権を定義し、カスタムリポジトリーへのユーザーアクセスを制限します。デフォルトでは、ユーザーは Automation Hub 内のすべてのパブリックリポジトリーを表示できますが、ロールで変更アクセス権が許可されていない限り、リポジトリーを変更することはできません。同じロジックがリポジトリー上の他の操作にも適用されます。たとえば、ユーザーのロールの権限を変更することで、カスタムリポジトリーからコンテンツをダウンロードするユーザーの機能を削除できます。Automation Hub でのユーザーアクセスの管理に関する詳細は、Private Automation Hub のユーザーアクセスの設定 を参照してください。

2.3.4. Automation Hub でのカスタムリポジトリーの作成

Red Hat Ansible Automation Platform を使用してリポジトリーを作成するときに、リポジトリーをプライベートに設定したり、検索結果から非表示にしたりできます。

手順

1. Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Repositories** を選択します。
3. **Add repository** をクリックします。
4. **Repository name** を入力します。
5. **Description** フィールドに、リポジトリーの目的を記述します。
6. 変更を加えるたびにリポジトリーの以前のバージョンを保持するには、**Retained number of versions** を選択します。保持されるバージョンの数は、0 から無制限までの範囲で指定できます。すべてのバージョンを保存するには、これを null に設定したままにします。



注記

カスタムリポジトリーへの変更で問題が発生した場合は、保持している 別のリポジトリーバージョンに戻す ことができます。

7. **Pipeline** フィールドで、リポジトリーのパイプラインを選択します。このオプションでは、コレクションをリポジトリーに公開できるユーザーを定義します。

   Staging

   誰でも自動化コンテンツをリポジトリーに公開できます。

   Approved

   このリポジトリーに追加されたコレクションは、ステージングリポジトリーを介して承認プロセスを通過する必要があります。自動承認が有効になっていると、ステージングリポジトリーにアップロードされたコレクションは、承認されたすべてのリポジトリーに自動的にプロモートされます。

   None

   リポジトリーに対する権限を持つすべてのユーザーがリポジトリーに直接公開できます。このリポジトリーは承認パイプラインには含まれません。

8. オプション: 検索結果からリポジトリーを非表示にするには、**Hide from search** を選択します。このオプションはデフォルトで選択されます。

9. オプション: リポジトリーをプライベートにするには、**Make private** を選択します。これにより、リポジトリーを表示する権限を持たない人に対してリポジトリーが非表示になります。

10. リモートリポジトリーのコンテンツをこのリポジトリーに同期するには、**Remote** を選択し、カスタムリポジトリーに含めるコレクションを含むリモートを選択します。詳細は、リポジトリーの同期 を参照してください。

11. **Save** をクリックします。

次のステップ

- リポジトリーが作成されると、詳細ページが表示されます。
  ここから、リポジトリーへのアクセスを提供したり、コレクションを確認または追加したり、カスタムリポジトリーの保存されたバージョンを操作したりできます。

2.3.5. カスタム Automation Hub リポジトリーへのアクセスの提供

デフォルトでは、プライベートリポジトリーと自動化コンテンツコレクションは、システム内のすべてのユーザーに対して非表示になります。パブリックリポジトリーはすべてのユーザーが表示できますが、変更することはできません。この手順を使用して、カスタムリポジトリーへのアクセスを提供します。

手順

1. Private Automation Hub にログインします。

2. ナビゲーションパネルから、**Collections → Repositories** を選択します。

3. リスト内でリポジトリーを見つけて、**More Actions** アイコン ⋮ をクリックし、**Edit** を選択します。

4. **Access** タブを選択します。

5. **Repository owners** のグループを選択します。
   ユーザーアクセスの実装に関する詳細は、Private Automation Hub のユーザーアクセスの設定を参照してください。

6. 選択したグループに割り当てるロールを選択します。

7. **Save** をクリックします。

2.3.6. Automation Hub リポジトリーへのコレクションの追加

リポジトリーを作成したら、自動化コンテンツコレクションの追加を開始できます。

手順

1. ナビゲーションパネルから、**Collections → Repositories** を選択します。

2. リスト内でリポジトリーを見つけて、**More Actions** アイコン ⋮ をクリックし、**Edit** を選択します。

3. **Collections version** タブを選択します。

4. **Add Collection** をクリックし、リポジトリーに追加するコレクションを選択します。

5. **Select** をクリックします。

2.3.7. 別の Automation Hub リポジトリーバージョンに戻す

自動化コンテンツコレクションをリポジトリーに追加またはリポジトリーから削除すると、新しいバージョンが作成されます。リポジトリーへの変更によって問題が発生した場合は、以前のバージョンに戻すことができます。元に戻すことは安全な操作です。元に戻しても、コレクションはシステムから削除されず、リポジトリーに関連付けられたコンテンツが変更されます。保存するバージョンの数は、リポジトリーの作成 時に、**Retained number of versions** 設定で定義します。

手順

1. Private Automation Hub にログインします。

2. ナビゲーションパネルから、**Collections → Repositories** を選択します。

3. リスト内でリポジトリーを見つけて、**More Actions** アイコン ⋮ をクリックし、**Edit** を選択します。

4. 元に戻すバージョンを見つけて、**More Actions** アイコン ⋮ をクリックし、**Revert to this version** を選択します。

5. **Revert** をクリックします。

2.3.8. Automation Hub でのリモート設定の管理

Automation Hub を実行している任意のサーバーにリモート設定をセットアップできます。リモート設定を使用すると、外部コレクションソースからカスタムリポジトリーにコンテンツを同期できます。

2.3.8.1. Automation Hub でのリモート設定の作成

Red Hat Ansible Automation Platform を使用して、外部のコレクションソースへのリモート設定を作成できます。その後、そのコレクションのコンテンツをカスタムリポジトリーに同期できます。

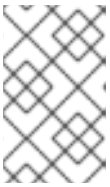
手順

1. Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Remotes** を選択します。
3. **Add Remote** をクリックします。
4. リモート設定の **Name** を入力します。
5. 特定のリポジトリーのパスを含む、リモートサーバーの **URL** を入力します。

注記

リモートサーバー URL とリポジトリーパスを見つけるには、**Collection → Repositories** に移動し、リポジトリーを選択して、**Copy CLI configuration** をクリックします。

6. 外部コレクションへのアクセスに必要な **Token** または **Username** と **Password** を入力して、リモートサーバーへの認証情報を設定します。



注記

ナビゲーションパネルからトークンを生成するには、**Collections → API token** を選択し、**Load token** をクリックして、ロードされたトークンをコピーします。

7. console.redhat.com からコレクションにアクセスするには、**SSO URL** を入力してアイデンティティプロバイダー (IdP) にサインインします。
8. **YAML 要件** ファイルを選択または作成して、カスタムリポジトリーと同期するコレクションとバージョン範囲を特定します。たとえば、kubernetes と AWS コレクションのバージョン 5.0.0 以降のみをダウンロードする場合、要件ファイルは次のようになります。

```
Collections:
   - name: community.kubernetes
  - name: community.aws
   version:”>=5.0.0”
```

注記

すべてのコレクションの依存関係は、同期プロセス中にダウンロードされます。

9. オプション: リモートをさらに設定するには、**Advanced configuration** で利用可能なオプションを使用します。
   a. 組織に企業プロキシーが設定されている場合は、**Proxy URL**、**Proxy Username**、および **Proxy Password** を入力します。
   b. **TLS validation** チェックボックスを使用して、トランスポート層セキュリティーを有効または無効にします。

c. 認証にデジタル証明書が必要な場合は、**Client key** と **Client certificate** を入力します。

d. サーバーに自己署名 SSL 証明書を使用している場合は、認証に使用される PEM エンコードされたクライアント証明書を **CA certificate** フィールドに入力します。

e. このリモートのコレクションをダウンロードできる速度を高速化するには、**Download concurrency** フィールドで同時にダウンロードできるコレクションの数を指定します。

f. このリモートで 1 秒あたりのクエリー数を制限するには、**Rate Limit** を指定します。

注記

一部のサーバーには特定の流量制御が設定されている場合があり、制限を超えると同期が失敗します。

2.3.8.2. リモート設定へのアクセスの提供

リモート設定を作成した後、それを使用できるようにするには、その設定へのアクセスを提供する必要があります。

手順

1. Private Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Remotes** を選択します。
3. リスト内でリポジトリーを見つけ、**More Actions** アイコン ⋮ をクリックして、**Edit** を選択します。
4. **Access** タブを選択します。
5. **Repository owners** のグループを選択します。ユーザーアクセスの実装に関する詳細は、Private Automation Hub のユーザーアクセスの設定 を参照してください。
6. 選択したグループに適切なロールを選択します。
7. **Save** をクリックします。

2.3.9. Automation Hub でのリポジトリーの同期

ある Automation Hub から別のオートメーションハブにリポジトリーを同期することで、関連する自動化コレクションコンテンツをユーザーに配布できます。最新のコレクション更新を確実に入手するには、カスタムリポジトリーをリモートと定期的に同期してください。

手順

1. Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections → Repositories** を選択します。
3. リスト内でリポジトリーを見つけて、**Sync** をクリックします。
   設定されたリモート内のすべてのコレクションがカスタムリポジトリーにダウンロードされます。コレクションの同期のステータスを確認するには、Navigation パネルから **Task Management** を選択します。

注記

リポジトリーの同期をリモート内の特定のコレクションに制限するには、requirements.yml ファイルを使用してプルする特定のコレクションを指定します。詳細は、Create a remote を参照してください。

関連情報

要件ファイルの使用の詳細は、**Using Ansible collections** ガイドの Install multiple collections with a requirements file を参照してください。

2.3.10. Automation Hub でのコレクションのエクスポートとインポート

Ansible Automation Hub は、自動化コンテンツコレクションをリポジトリー内に保存します。これらのコレクションは、自動化コンテンツ作成者によってバージョン管理されます。同じコレクションの多くのバージョンが、同じまたは異なるリポジトリーに同時に存在することがあります。

コレクションは、インポートおよびエクスポートできる .tar ファイルとして保存されます。この保存形式により、以前に作成してエクスポートしたコレクションと同じものを、新しいリポジトリーに確実にインポートできます。

2.3.10.1. Automation Hub での自動化コンテンツコレクションのエクスポート

コレクションが完成したら、組織全体の他のユーザーに配布できる場所にコレクションをインポートできます。

手順

1. Private Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections** → **Collections** を選択します。**Collections** ページには、すべてのリポジトリーにわたるすべてのコレクションが表示されます。特定のコレクションを検索できます。
3. エクスポートするコレクションを選択します。コレクションの詳細ページが開きます。
4. **Install** タブから、**Download tarball** を選択します。.tar ファイルがデフォルトのブラウザーのダウンロードフォルダーにダウンロードされます。これで、選択した場所にインポートできるようになります。

2.3.10.2. Automation Hub での自動化コンテンツコレクションのインポート

自動化コンテンツ作成者は、カスタムリポジトリーで使用するためにコレクションをインポートできます。カスタムリポジトリーでコレクションを使用するには、まずコレクションを名前空間にインポートして、Automation Hub 管理者が承認できるようにする必要があります。

手順

1. Automation Hub にログインします。
2. ナビゲーションパネルから、**Collections** → **Namespaces** を選択します。**Namespaces** ページには、使用可能なすべての名前空間が表示されます。
3. **View Collections** をクリックします。

4. **Upload Collection** をクリックします。
5. コレクションの tarball ファイルに移動し、ファイルを選択して、**Open** をクリックします。
6. **Upload** をクリックします。
   My Imports 画面にはテストの概要が表示され、コレクションのアップロードが成功したか失敗したかが通知されます。

注記

コレクションが承認されていない場合は、コレクションが公開リポジトリーに表示されません。

関連情報

- コレクションとリポジトリーの承認の詳細は、承認パイプライン を参照してください。

第3章 PRIVATE AUTOMATION HUB でのコンテナーの管理

Private Automation Hub のコンテナーレジストリーおよびリポジトリーを設定するための管理者のワークフローとプロセスを説明します。

3.1. PRIVATE AUTOMATION HUB コンテナーレジストリーの管理

Automation Hub コンテナーレジストリーを使用して、Ansible Automation Platform インフラストラクチャーでコンテナーイメージリポジトリーを管理します。Automation Hub を使用して次のタスクを実行できます。

- 個々のコンテナーリポジトリーにアクセスできるユーザーを制御する
- イメージのタグを変更する
- アクティビティーとイメージレイヤーを表示する
- 各コンテナーリポジトリーに関連する追加情報を提供する

3.1.1. コンテナーレジストリー

Automation Hub コンテナーレジストリーは、コンテナーイメージの保存と管理に使用されます。コンテナーイメージをビルドまたは取得したら、そのコンテナーイメージを Private Automation Hub のレジストリーの部分にプッシュしてコンテナーリポジトリーを作成できます。

次のステップ

- コンテナーイメージを Automation Hub のコンテナーレジストリーにプッシュします。
- レジストリー内のコンテナーリポジトリーにアクセスできるグループを作成します。
- 新規グループをコンテナーリポジトリーに追加します。
- コンテナーリポジトリーに README を追加して、ユーザーに情報や関連リンクを提供します。

3.2. PRIVATE AUTOMATION HUB でコンテナーリポジトリーのユーザーアクセスを設定する

Ansible Automation Platform 内のイメージにアクセスして管理できるユーザーを決定するには、Private Automation Hub 内のコンテナーリポジトリーに対するユーザーアクセスを設定する必要があります。

3.2.1. コンテナーレジストリーのグループ権限

ユーザーが Private Automation Hub で管理されているコンテナーと対話する方法を制御できます。次の権限のリストを使用して、コンテナーレジストリーに対する適切な権限を持つグループを作成します。

表3.1 Private Automation Hub でのコンテナー管理に使用するグループ権限のリスト

権限名	説明
新規コンテナーの作成	ユーザーは新規コンテナーを作成できます。

権限名	説明
コンテナーの名前空間権限の変更	ユーザーは、コンテナーリポジトリーの権限を変更できます。
コンテナーの変更	ユーザーはコンテナーの情報を変更できます。
イメージタグの変更	ユーザーはイメージタグを変更できます。
プライベートコンテナーのプル	ユーザーはプライベートコンテナーからイメージをプルできます。
既存コンテナーへのプッシュ	既存のコンテナーにイメージをプッシュすることができます。
プライベートコンテナーの表示	ユーザーは、プライベートとしてマークされているコンテナーを表示できます。

3.2.2. Private Automation Hub での新しいグループの作成

ユーザーがシステム内の指定された機能にアクセスできる権限を作成し、Private Automation Hub のグループに割り当てることができます。デフォルトでは、Automation Hub の **Admin** グループにはすべての権限が割り当てられています。このグループは初回ログイン時に使用可能になります。Private Automation Hub のインストール時に作成された認証情報を使用してください。

詳細は、「Automation Hub のスタートガイド」の Private Automation Hub での新しいグループの作成 を参照してください。

3.2.3. グループへの権限の割り当て

デフォルトでは、新しいグループには権限が割り当てられていません。ユーザーがシステム内の特定の機能にアクセスできる権限を Private Automation Hub のグループに割り当てることができます。

最初にグループを作成するときに権限を追加することも、既存のグループを編集して権限を追加または削除することもできます。

詳細は、「Automation Hub のスタートガイド」の グループへの権限の割り当て を参照してください。

関連情報

- 特定の権限についての詳細は、コンテナーレジストリーのグループ権限 を参照してください。

3.2.4. 既存のグループへのユーザーの追加

グループを作成するときに、グループにユーザーを追加できます。ただし、既存のグループにユーザーを手動で追加することもできます。

詳細は、「Automation Hub のスタートガイド」の 既存のグループへのユーザーの追加 を参照してください。

3.3. PRIVATE AUTOMATION HUB コンテナーレジストリーへの入力

デフォルトでは、Private Automation Hub にはコンテナーイメージが含まれていません。コンテナーレジストリーを設定するには、コンテナーイメージレジストリーにコンテナーイメージをプッシュする必要があります。

Private Automation Hub コンテナーレジストリーを設定するには、特定のワークフローに従う必要があります。

- Red Hat Ecosystem Catalog (registry.redhat.io) からイメージをプルします。
- イメージにタグを付けます。
- Private Automation Hub コンテナーレジストリーにプッシュします。

重要

イメージマニフェストとファイルシステム Blob はどちらも、元々は **registry.redhat.io** および **registry.access.redhat.com** から直接提供されていました。2023 年 5 月 1 日以降、ファイルシステム Blob は代わりに **quay.io** から提供されます。

- ネットワークポートとプロトコル (**表 5.10. 実行環境 (EE)** が、コンテナーイメージをプルする際の問題を回避するのに使用できることを確認してください。

registry.redhat.io または **registry.access.redhat.com** への送信接続を特に有効にするすべてのファイアウォール設定にこの変更を加えます。

ファイアウォールルールを設定するときは、IP アドレスの代わりにホスト名を使用します。

この変更を行った後、引き続き **registry.redhat.io** および **registry.access.redhat.com** からイメージをプルできます。Red Hat コンテナーイメージのプルを続行するために、**quay.io** にログインする必要も、**quay.io** レジストリーと直接やりとりする必要もありません。

ただし、Web ベースの Red Hat Subscription Management 上のマニフェスト (「サブスクリプション割り当て」とも呼ばれる) は、1 つの例外を除いて、2024 年初頭にサポートされなくなりました。システムが、Red Hat のサーバーから直接更新を受信しないクローズドネットワークまたは「エアギャップ」システムの一部である場合、マニフェストは Red Hat Satellite 6.16 のリリースまでサポートされます。Red Hat Satellite 6.16 のリリース日の発表については、Red Hat Satellite のリリース日 を確認してください。

3.3.1. Automation Hub で使用するイメージのプル

コンテナーイメージを Private Automation Hub にプッシュする前に、まず既存のレジストリーからプルし、使用できるようにタグを付ける必要があります。次の例では、Red Hat Ecosystem Catalog (registry.redhat.io) からイメージをプルする方法を説明します。

重要

2024 年初頭の時点で、Red Hat は Red Hat Subscription Management Web プラットフォーム上のマニフェストまたはマニフェストリスト (「サブスクリプション割り当て」と同義で使用) をサポートしなくなりました。Red Hat は、1 つの例外を除いて、Red Hat Satellite のマニフェスト機能のほとんどをサポートしなくなりました: * Red Hat サーバーから直接更新を受信しないクローズドネットワークまたは「エアギャップ」ネットワーク内の Red Hat Satellite ユーザーは、Red Hat Satellite 6.16 がリリースされるまで **access.redhat.com** を引き続き使用できます。

新しい Red Hat アカウントでは、サブスクリプションツールに Simple Content Access が自動的に使用されます。新しい Red Hat アカウントと、Red Hat のサーバーに接続できる既存の Satellite のお客様は、**console.redhat.com** でマニフェストを見つけることができます。

前提条件

- registry.redhat.io からイメージをプルする権限がある。
- Simple Content Access が有効になっている Red Hat アカウント。

手順

1. コンテナーイメージのマニフェストにアクセスする必要がある場合は、Red Hat コンソール にログインします。
2. コンテナーイメージに必要なマニフェストの 3 つのドットメニューをクリックし、**Export manifest** をクリックします。
3. registry.redhat.io の認証情報を使用して Podman にログインします。

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

4. ユーザー名およびパスワードを入力します。
5. コンテナーイメージをプルします。

```
$ podman pull registry.redhat.io/<container_image_name>:<tag>
```

検証

最近プルしたイメージがリストに含まれていることを確認するには、次の手順を実行します。

1. ローカルストレージ内のイメージをリスト表示します。

```
$ podman images
```

2. イメージ名を確認し、タグが正しいことを確認します。

関連情報

- イメージの登録および取得に関する詳細は、Red Hat Ecosystem Catalog Help を参照してください。

- Red Hat サブスクリプションツールの変更点に関する詳細は、接続された Satellite Server のマニフェストの作成と管理 を参照してください。

3.3.2. Automation Hub で使用するイメージのタグ付け

レジストリーからイメージをプルしたら、Private Automation Hub コンテナーレジストリーで使用するようにタグを付けます。

前提条件

- 外部レジストリーからコンテナーイメージをプルしている。
- Automation Hub インスタンスの FQDN または IP アドレスがある。

手順

- Automation Hub コンテナーリポジトリーを使用してローカルイメージにタグを付けます。

```
$ podman tag registry.redhat.io/<container_image_name>:<tag>
<automation_hub_hostname>/<container_image_name>
```

検証

1. ローカルストレージ内のイメージをリスト表示します。

```
$ podman images
```

2. Automation Hub の情報で最近タグ付けされたイメージが一覧に含まれていることを確認します。

3.3.3. Private Automation Hub へのコンテナーイメージのプッシュ

タグ付けされたコンテナーイメージを Private Automation Hub にプッシュして、新しいコンテナーを作成し、コンテナーレジストリーに追加できます。

前提条件

- 新規コンテナーを作成する権限がある。
- Automation Hub インスタンスの FQDN または IP アドレスがある。

手順

1. Automation Hub の場所および認証情報を使用して Podman にログインします。

```
$ podman login -u=<username> -p=<password> <automation_hub_url>
```

2. コンテナーイメージを Automation Hub のコンテナーレジストリーにプッシュします。

```
$ podman push <automation_hub_url>/<container_image_name>
```

トラブルシューティング

push 操作は、アップロード中にイメージレイヤーを再圧縮します。この操作は、再現性が保証されておらず、クライアントの実装に依存します。これにより、イメージレイヤーダイジェストが変更され、プッシュ操作が失敗し、**Error: Copying this image requires changing layer representation, which is not possible (image is signed or the destination specifies a digest)** エラーが発生します。

検証

1. Automation Hub にログインします。
2. **Container Registry** に移動します。
3. コンテナーリポジトリーリストでコンテナーを見つけます。

3.4. コンテナーリポジトリーの設定

コンテナーリポジトリーを設定する際には、説明の追加、README の追加、リポジトリーにアクセスできるグループの追加、およびイメージのタグ付けを行う必要があります。

3.4.1. コンテナーレジストリーを設定するための前提条件

- Private Automation Hub にログインしている。
- リポジトリーを変更する権限が必要です。

3.4.2. コンテナーリポジトリーへの README の追加

コンテナーリポジトリーに README を追加して、コンテナーを操作する方法をユーザーに提供します。Automation Hub コンテナーリポジトリーは、README を作成するためのマークダウンをサポートします。デフォルトでは、README は空です。

前提条件

- コンテナーを変更する権限がある。

手順

1. Automation Hub にログインします。
2. ナビゲーションパネルから、**Execution Environments → Execution Environments** を選択します。
3. コンテナーリポジトリーを選択します。
4. **Detail** タブで、**Add** をクリックします。
5. **Raw Markdown** テキストフィールドに、Markdown で README テキストを入力します。
6. 完了したら、**Save** をクリックします。

README を追加したら、**Edit** をクリックし、ステップ 4 および 5 を繰り返すことで、いつでも編集できます。

3.4.3. コンテナーリポジトリーへのアクセスの提供

イメージを操作する必要があるユーザーにコンテナーリポジトリーへのアクセスを提供します。グルー

プを追加すると、グループがコンテナーリポジトリーに対して持つことができる権限を変更できます。このオプションを使用して、グループが割り当てられている内容に応じて権限を拡張または制限できます。

前提条件

- **コンテナーの名前空間の権限を変更している。**

手順

1. Automation Hub にログインします。
2. ナビゲーションパネルから、**Execution Environments → Execution Environments** を選択します。
3. コンテナーリポジトリーを選択します。
4. **Access** タブで、**Select a group** をクリックします。
5. アクセスを許可するグループを選択し、**Next** をクリックします。
6. この実行環境に追加するロールを選択し、**Next** をクリックします。
7. **Add** をクリックします。

3.4.4. コンテナーイメージのタグ付け

Automation Hub コンテナーリポジトリーに保存されているイメージにタグを付けて、名前を追加します。イメージにタグが追加されない場合、Automation Hub の名前はデフォルトで **latest** に設定されます。

前提条件

- **change image tags** の権限がある。

手順

1. ナビゲーションパネルから、**Execution Environments → Execution Environments** を選択します。
2. コンテナーリポジトリーを選択します。
3. **Images** タブをクリックします。
4. **More Actions** アイコン ⋮ をクリックし、**Manage tags** をクリックします。
5. テキストフィールドに新しいタグを追加し、**Add** をクリックします。
6. (必要に応じて) そのイメージのいずれのタグの **x** をクリックして、**current tags** を削除します。
7. **Save** をクリックします。

検証

- **Activity** タブをクリックし、最新の変更を確認します。

3.4.5. Automation Controller での認証情報の作成

パスワードまたはトークンで保護されたレジストリーからコンテナーイメージをプルするには、Automation Controller で認証情報を作成する必要があります。

Ansible Automation Platform の以前のバージョンでは、実行環境イメージを格納するためにレジストリーをデプロイする必要がありました。Ansible Automation Platform 2.0 以降のシステムは、コンテナーレジストリーがすでに稼働していると想定して動作します。実行環境イメージを格納するには、選択したコンテナーレジストリーについてのみ認証情報を追加します。

手順

1. Automation Controller に移動します。
2. ナビゲーションパネルから **Resources** → **Credentials** を選択します。
3. **Add** をクリックして、新規の認証情報を作成します。
4. 承認用の **名前**、**説明**、および **組織** を入力します。
5. **認証情報のタイプ** を選択します。
6. **認証用 URL** を入力します。これはコンテナーレジストリーのアドレスです。
7. コンテナーレジストリーへのログインに必要な **ユーザー名** と **パスワードまたはトークン** を入力します。
8. オプション: SSL 検証を有効にするには、**Verify SSL** を選択します。
9. **Save** をクリックします。

3.5. コンテナーリポジトリーからのイメージのプル

Automation Hub コンテナーレジストリーからイメージを取得し、ローカルマシンにコピーを作成します。Automation Hub は、コンテナーリポジトリーの **latest** イメージごとに、**podman pull** コマンドを提供します。このコマンドを端末にコピーアンドペーストするか、**podman pull** を使用してイメージタグに基づいてイメージをコピーすることができます。

3.5.1. イメージのプル

Automation Hub コンテナーレジストリーからイメージをプルして、ローカルマシンにコピーできます。

前提条件

- プライベートコンテナーリポジトリーを表示およびプルする権限がある。

手順

1. パスワードまたはトークンで保護されたレジストリーからコンテナーイメージをプルする必要がある場合は、イメージをプルする前に Automation Controller で認証情報を作成 する必要があります。
2. ナビゲーションパネルから、**Execution Environments** → **Execution Environments** を選択します。

3. コンテナーリポジトリーを選択します。
4. **Pull this image** エントリーで、**Copy to clipboard** をクリックします。
5. 端末でコマンドを貼り付けます。

検証

- **podman images** を実行して、ローカルマシンにイメージを表示します。

3.5.2. コンテナーリポジトリーからのイメージの同期

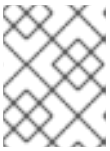
Automation Hub コンテナーレジストリーからイメージをプルして、イメージをローカルマシンに同期できます。リモートコンテナーレジストリーからイメージを同期するには、まずリモートレジストリーを設定する必要があります。

前提条件

プライベートコンテナーリポジトリーを表示およびプルする権限がある。

手順

1. ナビゲーションパネルから、**Execution Environments → Execution Environments** を選択します。
2. https://registry.redhat.io をレジストリーに追加します。
3. 認証に必要な認証情報を追加します。



注記

コンテナーレジストリーの中には、流量制御を積極的に行っているものもあります。**Advanced Options** で流量制御を設定します。

4. ナビゲーションパネルから、**Execution Environments → Execution Environments** を選択します。
5. ページヘッダーの **Add execution environment** をクリックします。
6. 取得元のレジストリーを選択します。**Name** フィールドには、ローカルレジストリーに表示されるイメージの名前が表示されます。

注記

Upstream name フィールドは、リモートサーバー上のイメージの名前です。たとえば、アップストリーム名が "alpine" に設定され、**Name** フィールドが "local/alpine" の場合、alpine イメージがリモートからダウンロードされ、"local/alpine" に名前が変更されます。

7. 追加または除外するタグのリストを設定します。イメージに多数のタグがあると、イメージの同期に時間がかかり、大量のディスク容量が使用されます。

関連情報

- レジストリーのリストについては、Red Hat コンテナーレジストリーの認証 を参照してください。
- イメージをプルする際に使用するオプションは、Podman とは ドキュメントを参照してください。

3.6. 署名済みコンテナーの操作

自動化実行環境は、Ansible Automation Controller がジョブを実行するために使用するコンテナーイメージです。このコンテンツを Private Automation Hub にダウンロードし、組織内で公開できます。

3.6.1. コンテナー署名用のシステムのデプロイ

Automation Hub は、実行環境コンテナーイメージのセキュリティーを強化するために、イメージ署名を実装します。

コンテナー署名の準備ができるようにシステムをデプロイするには、署名スクリプトを作成します。

注記

インストーラーは、インストーラーが配置されているサーバーと同じサーバー上でスクリプトとキーを探します。

手順

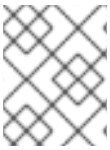
1. ターミナルから署名スクリプトを作成し、スクリプトパスをインストーラーパラメーターとして渡します。
   例:

```
#!/usr/bin/env bash

# pulp_container SigningService will pass the next 4 variables to the script.
MANIFEST_PATH=$1
FINGERPRINT="$PULP_SIGNING_KEY_FINGERPRINT"
IMAGE_REFERENCE="$REFERENCE"
SIGNATURE_PATH="$SIG_PATH"

# Create container signature using skopeo
skopeo standalone-sign \
  $MANIFEST_PATH \
  $IMAGE_REFERENCE \
  $FINGERPRINT \
  --output $SIGNATURE_PATH

# Optionally pass the passphrase to the key if password protected.
# --passphrase-file /path/to/key_password.txt

# Check the exit status
STATUS=$?
if [ $STATUS -eq 0 ]; then
  echo {\"signature_path\": \"$SIGNATURE_PATH\"}
else
  exit $STATUS
fi
```

2. Ansible Automation Platform インストーラーインベントリーファイルで、**automationhub_*** で始まるコンテナー署名のオプションを確認してください。

```
[all:vars]
.
.
.

automationhub_create_default_container_signing_service = True
automationhub_container_signing_service_key = /absolute/path/to/key/to/sign
automationhub_container_signing_service_script = /absolute/path/to/script/that/signs
```

3. インストールが完了したら、Automation Hub に移動します。
4. ナビゲーションパネルから、**Signature Keys** を選択します。
5. **container-default** または **container-anyname** というタイトルのキーがあることを確認します。



注記

Container-default サービスは、Ansible Automation Platform インストーラーによって作成されます。

3.6.2. Automation Hub に対するリモートでのコンテナーの追加

次の 2 つの方法のいずれかで、コンテナーを Automation Hub にリモートで追加できます。

- リモートの作成
- Execution Environment

手順

1. Automation Hub にログインします。
2. ナビゲーションパネルから、**Execution Environments → Remote Registries** を選択します。
3. **Add remote registry** をクリックします。
   - **Name** フィールドに、コンテナーが存在するレジストリーの名前を入力します。
   - **URL** フィールドに、コンテナーが存在するレジストリーの URL を入力します。
   - **Username** フィールドに、必要に応じてユーザー名を入力します。
   - **Password** フィールドに、必要に応じてパスワードを入力します。
   - **Save** をクリックします。

3.6.3. 実行環境の追加

自動化実行環境は、システムレベルの依存関係とコレクションベースのコンテンツを組み込むことを可能にするコンテナーイメージです。各実行環境では、ジョブを実行するためのカスタマイズされたイメージを使用できます。各イメージには、ジョブの実行時に必要なものだけを含めます。

手順

1. ナビゲーションパネルから、**Execution Environments → Execution Environments** を選択します。
2. **Add execution environment** をクリックします。
3. 実行環境の名前を入力します。
4. オプション: アップストリーム名を入力します。
5. **Registry** で、ドロップダウンメニューからレジストリーの名前を選択します。
6. **Add tag(s) to include** フィールドにタグを入力します。フィールドが空白の場合、すべてのタグが渡されます。どのリポジトリー固有のタグを渡すかを指定する必要があります。
7. 残りのフィールドはオプションです。
   - **Currently included tags**
   - **Add tag(s) to exclude**
   - **Currently excluded tag(s)**
   - **Description**
8. **Save** をクリックします。
9. イメージを同期します。

3.6.4. ローカル環境からのコンテナーイメージのプッシュ

次の手順を使用して、ローカルシステム上のイメージに署名し、それらの署名されたイメージを Automation Hub レジストリーにプッシュします。

手順

1. ターミナルから、Podman または現在使用しているコンテナークライアントにログインします。

```
> podman pull <container-name>
```

2. イメージをプルした後、タグを追加します (例: latest、rc、beta、または 1.0、2.3 などのバージョン番号)。

```
> podman tag <container-name> <server-address>/<container-name>:<tag name>
```

3. 変更を加えた後にイメージに署名し、Automation Hub レジストリーにプッシュし直します。

```
> podman push <server-address>/<container-name>:<tag name> --tls-verify=false --sign-by
<reference to the gpg key on your local>
```

イメージが署名されていない場合は、現在の署名が埋め込まれている場合にのみプッシュできます。あるいは、次のスクリプトを使用して、署名せずにイメージをプッシュすることもできます。

```
> podman push <server-address>/<container-name>:<tag name> --tls-verify=false
```

4. イメージがプッシュされたら、Automation Hub に移動します。
5. ナビゲーションパネルから、**Execution Environments → Execution Environments** を選択します。
6. 新しい実行環境を表示するには、**Refresh** アイコンをクリックします。
7. イメージの名前をクリックすると、プッシュされたイメージが表示されます。

トラブルシューティング

Automation Hub の詳細ページには、イメージが署名されているかが示されます。詳細ページに、イメージが **Unsigned** であることが示されている場合は、次の手順を使用して Automation Hub からイメージに署名できます。

1. イメージ名をクリックすると詳細ページに移動します。
2. **More Actions** アイコン ⋮ をクリックします。次の 3 つのオプションが利用可能です。
   - **Use in Controller**
   - **Delete**
   - **Sign**
3. ドロップダウンメニューから **Sign** をクリックします。

署名サービスがイメージに署名します。イメージが署名されると、ステータスが "signed" に変わります。

3.6.5. 署名済みイメージでのポリシーの使用

Podman またはその他のイメージクライアントがポリシーを使用して、特定のポリシーをその署名に割り当てることにより、イメージの有効性を保証できます。

3.6.6. Podman を使用して、イメージが特定の署名によって署名されていることを確認する

署名が特定の署名によって署名されていることを確認する場合は、その署名がローカル環境に存在する必要があります。

手順

1. ナビゲーションパネルから、**Signature Keys** を選択します。
2. 使用している署名の横にある **More Actions** アイコン ⋮ をクリックします。
3. ドロップダウンメニューから **Download key** を選択します。新しいウィンドウが開きます。
4. **Name** フィールドに、キーの名前を入力します。
5. **Save** をクリックします。

3.6.7. 署名を検証するためのクライアントの設定

リモートレジストリーからプルしたコンテナーイメージが適切に署名されていることを確認するには、まずポリシーファイルで適切な公開キーを使用してイメージを設定する必要があります。

前提条件

- 署名を検証するには、クライアントに sudo 権限が設定されている必要があります。

手順

1. ターミナルを開き、次のコマンドを使用します。

```
>  sudo <name of editor> /etc/containers/policy.json
```

表示されるファイルは次のようなものです。

```
{
  "default": [{"type": "reject"}],
  "transports": {
   "docker": {
     "quay.io": [{"type": "insecureAcceptAnything"}],
     "docker.io": [{"type": "insecureAcceptAnything"}],
     "<server-address>": [
       {
              "type": "signedBy",
              "keyType": "GPGKeys",
              "keyPath": "/tmp/containersig.txt"
       }]
   }
  }
}
```

このファイルは、**quay.io** も **docker.io** も検証を実行しないことを示しています。タイプが **insecureAcceptAnything** であり、これによってデフォルトのタイプの **reject** がオーバーライドされるためです。ただし、パラメーターの **type** が **"signedBy"** に設定されているため、**<server-address>** によって検証が実行されます。

注記

現在サポートされている唯一の **keyType** は GPG キーです。

2. **<server-address>** エントリーの下で、**keyPath** <1> を変更してキーファイルの名前を含めます。

```
{
    "default": [{"type": "reject"}],
    "transports": {
        "docker": {
            "quay.io": [{"type": "insecureAcceptAnything"}],
            "docker.io": [{"type": "insecureAcceptAnything"}],
            "<server-address>": [{
                "type": "signedBy",
                "keyType": "GPGKeys",
```

```
                "keyPath": "/tmp/<key file name>",
                "signedIdentity": {
                    "type": "matchExact"
                }
            }]
        }
    }
}
```

3. ファイルを保存してから閉じます。

検証

- Podman または任意のクライアントを使用して、ファイルをプルします。

```
> podman pull <server-address>/<container-name>:<tag name> --tls-verify=false
```

この応答は、イメージがエラーなしで署名されたことを確認します。イメージが署名されていない場合、コマンドは失敗します。

関連情報

- policy.json の詳細は、containers-policy.json のドキュメント を参照してください。

3.7. コンテナーリポジトリーの削除

Private Automation Hub からコンテナーリポジトリーを削除して、ディスク容量を管理します。**Container Repository** リストビューで、Red Hat Ansible Automation Platform インターフェイスからリポジトリーを削除できます。

前提条件

- リポジトリーを管理する権限があります。

手順

1. Automation Hub に移動します。
2. ナビゲーションパネルから、**Execution Environments → Execution Environments** を選択します。
3. 削除するコンテナーリポジトリーで、**More Actions** アイコン ⋮ をクリックし、**Delete** をクリックします。
4. 確認メッセージが表示されたら、チェックボックスをクリックし、**Delete** をクリックします。

検証

- **Execution Environments** リストビューに戻ります。コンテナーリポジトリーが正常に削除された場合、コンテナーリポジトリーはリストに表示されなくなります。