



# Red Hat Ansible Automation Platform 2.2

## Red Hat Ansible Automation Platform Operator の OpenShift Container Platform への デプロイ

このガイドでは、OpenShift Container Platform の Red Hat Ansible Automation Platform Operator でサポートされるインストールシナリオの手順および参考情報を提供します。



# Red Hat Ansible Automation Platform 2.2 Red Hat Ansible Automation Platform Operator の OpenShift Container Platform へのデプロイ

---

このガイドでは、OpenShift Container Platform の Red Hat Ansible Automation Platform Operator でサポートされるインストールシナリオの手順および参考情報を提供します。

## 法律上の通知

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

## 概要

フィードバックの提供: このドキュメントを改善するための提案がある場合、またはエラーを見つけた場合は、テクニカルサポート () に連絡し、Docs コンポーネントを使用して Ansible Automation Platform Jira プロジェクトで issue を作成してください。

## 目次

はじめに .....	4
多様性を受け入れるオープンソースの強化 .....	5
<b>第1章 RED HAT OPENSIFT CONTAINER PLATFORM への RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR のインストール計画 .....</b>	<b>6</b>
1.1. ANSIBLE AUTOMATION PLATFORM OPERATOR について	6
1.2. OPENSIFT CONTAINER PLATFORM バージョンの互換性	6
1.3. RED HAT OPENSIFT CONTAINER PLATFORM のサポート対象のインストールシナリオ	6
1.4. カスタムリソース	7
1.5. 関連情報	7
<b>第2章 RED HAT OPENSIFT CONTAINER PLATFORM への ANSIBLE AUTOMATION PLATFORM オペレーター のインストール .....</b>	<b>8</b>
<b>第3章 RED HAT OPENSIFT CONTAINER PLATFORM WEB コンソールへの AUTOMATION CONTROLLER のインストールおよび設定 .....</b>	<b>9</b>
3.1. 前提条件	9
3.2. AUTOMATION CONTROLLER OPERATOR のインストール	9
3.3. RED HAT ANSIBLE AUTOMATION PLATFORM オペレーターでの AUTOMATION CONTROLLER 用の外部データベースの設定	12
3.4. PVC の検索および削除	14
3.5. 関連情報	14
<b>第4章 RED HAT OPENSIFT CONTAINER PLATFORM WEB コンソールへの AUTOMATION HUB のインストールおよび設定 .....</b>	<b>15</b>
4.1. 前提条件	15
4.2. AUTOMATION HUB OPERATOR のインストール	15
4.3. AUTOMATION HUB ユーザーインターフェイスへのアクセス	19
4.4. RED HAT ANSIBLE AUTOMATION PLATFORM オペレーターでの AUTOMATION HUB 用の外部データベースの設定	19
4.5. PVC の検索および削除	20
4.6. 関連情報	21
<b>第5章 OPENSIFT CONTAINER PLATFORM CLI からの ANSIBLE AUTOMATION PLATFORM OPERATOR のインストール .....</b>	<b>22</b>
5.1. 前提条件	22
5.2. OPENSIFT CONTAINER PLATFORM CLI を使用した NAMESPACE の OPERATOR へのサブスクリプション	22
5.3. OPENSIFT CONTAINER PLATFORM CLI からの AUTOMATION CONTROLLER ログインの詳細の取得	23
5.4. 関連情報	25
<b>第6章 AUTOMATION HUB での RED HAT SINGLE SIGN-ON OPERATOR の使用 .....</b>	<b>26</b>
6.1. KEYCLOAK インスタンスの作成	26
6.2. ANSIBLE AUTOMATION PLATFORM の KEYCLOAK レルムの作成	27
6.3. KEYCLOAK クライアントの作成	28
6.4. KEYCLOAK ユーザーの作成	31
6.5. ANSIBLE AUTOMATION PLATFORM OPERATOR のインストール	32
6.6. RED HAT SINGLE SIGN-ON 接続シークレットの作成	32
6.7. OPERATOR を使用した AUTOMATION HUB のインストール	33
6.8. AUTOMATION HUB ルートの決定	34
6.9. RED HAT SINGLE SIGN-ON クライアントの更新	34
6.10. 関連情報	35

<b>第7章 RED HAT ANSIBLE AUTOMATION PLATFORM の ANSIBLE AUTOMATION PLATFORM OPERATOR への移行</b> .....	<b>36</b>
7.1. 移行に関する考慮事項	36
7.2. 移行の準備	36
7.3. ANSIBLE AUTOMATION PLATFORM OPERATOR へのデータの移行	39
7.4. 移行後のクリーンアップ	40
<b>第8章 OPENSIFT CONTAINER PLATFORM での ANSIBLE AUTOMATION PLATFORM OPERATOR のアップグレード</b> .....	<b>41</b>
8.1. アップグレードに関する考慮事項	41
8.2. 前提条件	41
8.3. ANSIBLE AUTOMATION PLATFORM OPERATOR のアップグレード	41



## はじめに

Red Hat Ansible Automation Platform に興味をお持ちいただきありがとうございます。Ansible Automation Platform は、Ansible を装備した環境に、制御、ナレッジ、委譲の機能を追加して、チームが複雑かつ複数層のデプロイメントを管理できるように支援する商用サービスです。

このドキュメントは、OpenShift Container Platform に Ansible Automation Platform Operator をデプロイするためのインストール、移行、およびアップグレードの要件を理解するのに役立ちます。



## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

# 第1章 RED HAT OPENSIFT CONTAINER PLATFORM への RED HAT ANSIBLE AUTOMATION PLATFORM OPERATOR のインストール計画

Red Hat Ansible Automation Platform は、Red Hat Enterprise Linux と Red Hat OpenShift の両方でサポートされます。

OpenShift Operator は、Red Hat OpenShift Container Platform に複雑な分散ソフトウェアの day 2 操作のインストールおよび自動化に役立ちます。Ansible Automation Platform Operator を使用すると、Red Hat OpenShift Container Platform に Red Hat OpenShift Container Platform をデプロイし、管理できます。

このセクションは、Red Hat OpenShift Container Platform 環境への Red Hat Ansible Automation Platform のインストールを計画するのに役立ちます。インストールを行う前に、サポートされているインストールシナリオを確認し、要件を満たしていることを確認してください。

## 1.1. ANSIBLE AUTOMATION PLATFORM OPERATOR について

Ansible Automation Platform Operator は、OpenShift 環境で新規の Ansible Automation Platform インスタンスのクラウドネイティブのプッシュボタンデプロイメントを提供します。Ansible Automation Platform Operator には、Automation Controller および Private Automation Hub のインスタンスをデプロイして管理するリソースタイプが含まれます。また、Automation Controller のデプロイメント内でジョブを定義して起動するための、オートメーションコントローラのジョブリソースも含まれています。

Kubernetes ネイティブ Operator を使用して Ansible Automation Platform インスタンスをデプロイすると、Red Hat OpenShift Container Platform にデプロイされた Playbook からインスタンスを起動するという利点があります。これには、Red Hat Ansible Automation Platform デプロイメントへのアップグレードおよび完全なライフサイクルサポートが含まれます。

OperatorHub の Red Hat Operator カタログから Ansible Automation Platform Operator をインストールできます。

## 1.2. OPENSIFT CONTAINER PLATFORM バージョンの互換性

Ansible Automation Platform 2.2 をインストールする Ansible Automation Platform Operator は、OpenShift Container Platform 4.8 以降のバージョンで利用できます。

## 1.3. RED HAT OPENSIFT CONTAINER PLATFORM のサポート対象のインストールシナリオ

Red Hat OpenShift Container Platform Web コンソールで OperatorHub を使用して、Ansible Automation Platform Operator をインストールできます。

または、OpenShift Container Platform コマンドラインインターフェイス (CLI) `oc` から Ansible Automation Platform Operator をインストールできます。

以下のワークフローのいずれかに従って Ansible Automation Platform Operator をインストールし、これを使用して必要な Ansible Automation Platform のコンポーネントをインストールします。

- 最初に Automation Controller カスタムリソース、次に Automation Hub カスタムリソース。
- 最初に Automation Hub カスタムリソース、次に Automation Controller カスタムリソース。

- Automation Controller カスタムリソース。
- Automation Hub カスタムリソース

## 1.4. カスタムリソース

プライマリーインストールワークフローごとにカスタムリソースを定義できます。

## 1.5. 関連情報

- OpenShift Container Platform OperatorHub の詳細は、[OperatorHub について](#) を参照してください。

## 第2章 RED HAT OPENSIFT CONTAINER PLATFORM への ANSIBLE AUTOMATION PLATFORM オペレーターのインストール

### 前提条件

- Operator Hub に Red Hat Ansible Automation Platform カタログがインストールされている。
- プラットフォーム用の **StorageClass** オブジェクトと、**ReadWriteMany** アクセスモードの永続ボリューム要求 (PVC) を作成している。詳細は、[Dynamic Provisioning](#) を参照してください。
- **ReadWriteMany** アクセスモードを使用して Amazon Web Services で Red Hat OpenShift Container Platform クラスターを実行するには、NFS またはその他のストレージを追加する必要がある。
  - AWS Elastic Block Store (EBS) の詳細、または **aws-efs** ストレージクラスの使用は、[AWS Elastic Block Store を使用した永続ストレージ](#) を参照してください。
  - AWS EBS のマルチアタッチ **ReadWriteMany** アクセスモードを使用するには、[Amazon EBS マルチアタッチを使用した複数のインスタンスへのボリュームのアタッチ](#) を参照してください。

### 手順

1. Red Hat OpenShift Container Platform にログインします。
2. **Operators** → **OperatorHub** に移動します。
3. Red Hat Ansible Automation Platform Operator を検索し、**Install** をクリックします。
4. **Update Channel** を選択します。
  - **stable-2.x**: namespace スコープの Operator をインストールします。これにより、Automation Hub および Automation Controller インスタンスのデプロイが、Operator がインストールされている namespace に制限されます。これはほとんどの場合に適しています。stable-2.x チャンネルは管理者特権を必要とせず、単一の namespace のみを監視するため、使用するリソースが少なくなります。
  - **stable-2.x-cluster-scoped**: クラスター内の複数の namespace にわたって Automation Hub と Automation Controller をデプロイし、クラスター内のすべての namespace に対して管理者特権を必要とします。
5. **Installation Mode**、**Installed Namespace**、および **Approval Strategy** を選択します。
6. **Install** をクリックします。

インストールプロセスが開始します。インストールが完了すると、Red Hat Ansible Automation Platform オペレーターが指定された名前空間にインストールされていることを通知するモーダルが表示されます。

- **View Operator** をクリックして、新しくインストールされた Red Hat Ansible Automation Platform Operator を表示します。

## 第3章 RED HAT OPENSIFT CONTAINER PLATFORM WEB コンソールへの AUTOMATION CONTROLLER のインストールおよび設定

これらの手順を使用して、自動化コントローラーオペレーターを Red Hat OpenShift Container Platform にインストールし、カスタムリソースを指定し、外部データベースを使用して Ansible Automation Platform をデプロイできます。



### 注記

自動化コントローラーのインスタンスが削除されても、関連する PVC は自動的に削除されません。これにより、新しいデプロイメントの名前が前のデプロイメントと同じ場合は、移行中に問題が発生する可能性があります。したがって、同じ namespace に新規 Automation Controller インスタンスをデプロイする前に、古い PVC を手動で削除することが推奨されます。詳細は、[PVC の検索および削除](#) を参照してください。

### 3.1. 前提条件

- Operator Hub に Red Hat Ansible Automation Platform カタログがインストールされている。

### 3.2. AUTOMATION CONTROLLER OPERATOR のインストール

1. **Operators** → **Installed Operators** に移動し、**Ansible Automation Platform** オペレーターをクリックします。
2. **Automation controller** タブを見つけて、**Create instance** をクリックします。

Form View または YAML ビューのいずれかを使用してインスタンスの設定を続行できます。

#### 3.2.1. 自動化コントローラーのフォームビューの作成

1. **フォームビュー** が選択されていることを確認します。デフォルトで選択されているはずですが。
2. 新しいコントローラーの名前を入力します。
3. オプション: 必要なラベルを追加します。
4. **Advanced configuration** をクリックします。
5. インスタンスの **ホスト名** を入力します。ホスト名はオプションです。デフォルトのホスト名は、選択したデプロイメント名に基づいて生成されます。
6. **管理者アカウントのユーザー名** を入力します。
7. **管理者の電子メールアドレス** を入力します。
8. **Admin password secret** ドロップダウンメニューで、**シークレット** を選択します。
9. **Database configuration secret** ドロップダウンメニューで、**シークレット** を選択します。
10. **Old Database configuration secret** ドロップダウンメニューで、**シークレット** を選択します。
11. **Secret key secret** ドロップダウンメニューで、**シークレット** を選択します。

12. **Broadcast Websocket Secret** ドロップダウンメニューで、シークレットを選択します。
13. 必要な **サービスアカウントアノテーション** を入力します。

### 3.2.2. コントローラーイメージのプルポリシーの設定

1. **Image Pull Policy** の下で、ラジオボタンをクリックして選択します。
  - Always
  - Never
  - IfNotPresent
2. **Image Pull Secrets** の下にあるオプションを表示するには、矢印をクリックします。
  - a. **Add Image Pull Secret** の横にある **+** をクリックして、値を入力します。
3. **Web container resource requirements** ドロップダウンリストの下にフィールドを表示するには、矢印をクリックします。
  - a. **Limits** および **Requests** で、**CPU cores**、**Memory**、および **Storage** の値を入力します。
4. **Task container resource requirements** ドロップダウンリストの下にフィールドを表示するには、矢印をクリックします。
  - a. **Limits** および **Requests** で、**CPU cores**、**Memory**、および **Storage** の値を入力します。
5. **EE Control Plane container resource requirements** ドロップダウンリストの下にフィールドを表示するには、矢印をクリックします。
  - a. **Limits** および **Requests** で、**CPU cores**、**Memory**、および **Storage** の値を入力します。
6. **PostgreSQL init container resource requirements (when using a managed service)** ドロップダウンリストの下にフィールドを表示するには、矢印をクリックします。
  - a. **Limits** および **Requests** で、**CPU cores**、**Memory**、および **Storage** の値を入力します。
7. **Redis container resource requirements** ドロップダウンリストの下にフィールドを表示するには、矢印をクリックします。
  - a. **Limits** および **Requests** で、**CPU cores**、**Memory**、および **Storage** の値を入力します。
8. **PostgreSQL container resource requirements (when using a managed instance)\*** ドロップダウンリストの下にフィールドを表示するには、矢印をクリックします。
  - a. **Limits** および **Requests** で、**CPU cores**、**Memory**、および **Storage** の値を入力します。
9. **PostgreSQL container storage requirements (when using a managed instance)** ドロップダウンリストを表示するには、矢印をクリックします。
  - a. **Limits** および **Requests** で、**CPU cores**、**Memory**、および **Storage** の値を入力します。
10. レプリカで、インスタンスレプリカの数を入力します。
11. **Remove used secrets on instance removal** で、**true** または **false** を選択します。デフォルトは **false** です。

12. **Preload instance with data upon creation**で、**true** または **false** を選択します。デフォルトは **true** です。

### 3.2.3. コントローラー LDAP セキュリティーの設定

1. **LDAP Certificate Authority Trust Bundle**の下でドロップダウンメニューをクリックし、**シークレット**を選択します。
2. **LDAP Password Secret**で、ドロップダウンメニューをクリックし、**シークレット**を選択します。
3. **EE Images Pull Credentials Secret**で、ドロップダウンメニューをクリックし、**シークレット**を選択します。
4. **Bundle Cacert Secret**で、ドロップダウンメニューをクリックし、**シークレット**を選択します。
5. **Service Type**の下で、ドロップダウンメニューをクリックして選択します。
  - **ClusterIP**
  - **LoadBalancer**
  - **NodePort**

### 3.2.4. Automation Controller Operator ルートオプションの設定

Red Hat Ansible Automation Platform Operator のインストールフォームでは、**Advanced configuration** で自動化コントローラー Operator ルートオプションをさらに設定できます。

1. **Advanced configuration** をクリックします。
2. **Ingress type**でドロップダウンメニューをクリックし、**Route**を選択します。
3. **Route DNS host**で、ルートの応答先となる共通のホスト名を入力します。
4. **Route TLS termination mechanism**ドロップダウンメニューをクリックし、**Edge** または **Passthrough** を選択します。ほとんどの場合は、**Edge** を選択する必要があります。
5. **Route TLS credential secret**で、ドロップダウンメニューをクリックし、一覧から**シークレット**を選択します。
6. **/var/lib/projects**ディレクトリーで**永続性を有効にする**で、スライダーを動かして **true** または **false** を選択します。

### 3.2.5. Automation Controller Operator の Ingress タイプの設定

Red Hat Ansible Automation Platform Operator のインストールフォームでは、**Advanced configuration** で Automation controller Operator Ingress をさらに設定できます。

#### 手順

1. **Advanced Configuration** をクリックします。
2. **Ingress type**でドロップダウンメニューをクリックし、**Ingress**を選択します。

3. **Ingress annotations** で、Ingress に追加するアノテーションを入力します。
4. **Ingress TLS secret** でドロップダウンメニューをクリックし、一覧からシークレットを選択します。

Automation Controller Operator を設定したら、フォームビューの下部にある **Create** をクリックします。Red Hat OpenShift Container Platform は Pod を作成します。これには数分の時間がかかる場合があります。

**Workloads** → **Pod** に移動し、新たに作成されたインスタンスを見つけることで、進捗を表示できます。

## 検証

Automation Controller から Ansible Automation Platform Operator インストールによって提供される以下の Operator Pod が実行されていることを確認します。

Operator マネージャーコントローラー	Automation Controller	Automation Hub
<p>3つの各 Operator の Operator マネージャーコントローラーには、以下が含まれます。</p> <ul style="list-style-type: none"> <li>● automation-controller-operator-controller-manager</li> <li>● automation-hub-operator-controller-manager</li> <li>● resource-operator-controller-manager</li> </ul>	<p>Automation Controller のデプロイ後に、これらの Pod が追加されていることを確認できます。</p> <ul style="list-style-type: none"> <li>● controller</li> <li>● controller-postgres</li> </ul>	<p>Automation Hub のデプロイ後に、これらの Pod が追加されていることを確認できます。</p> <ul style="list-style-type: none"> <li>● hub-api</li> <li>● hub-content</li> <li>● hub-postgres</li> <li>● hub-redis</li> <li>● hub-worker</li> </ul>



## 注記

Pod が見つからない場合は、プルシークレットが必要であることを示している可能性があります。プルシークレットは、保護されたイメージレジストリーまたはプライベートイメージレジストリーに必要です。詳細は、[イメージプルシークレットの使用](#) を参照してください。**oc describe pod <pod-name>** を実行して、その Pod に ImagePullBackOff エラーがあるかどうかを確認することで、この問題をさらに診断できます。

## 3.3. RED HAT ANSIBLE AUTOMATION PLATFORM オペレーターでの AUTOMATION CONTROLLER 用の外部データベースの設定

外部データベースを使用して Ansible Automation Platform をデプロイすることを希望するユーザーは、インスタンスの認証情報と接続情報を使用してシークレットを設定し、**oc create** コマンドを使用してクラスターに適用するとデプロイできるようになります。

デフォルトでは、Red Hat Ansible Automation Platform オペレーターは、Ansible Automation Platform デプロイメントと同じ名前空間でマネージド PostgreSQL Pod を自動的に作成および設定します。Red Hat Ansible Automation Platform Operator が自動作成するマネージド PostgreSQL Pod ではなく、外



外部データベースで Ansible Automation Platform をデプロイできます。

外部データベースを使用すると、リソースを共有して再利用でき、バックアップ、アップグレード、およびパフォーマンスの最適化を手動で管理できます。



### 注記

データベース名が異なる限り、Automation Hub と Automation Controller の両方に同じ外部データベース (PostgreSQL インスタンス) を使用できます。つまり、単一の PostgreSQL インスタンス内に異なる名前のデータベースを複数指定できます。

次のセクションでは、Ansible Automation Platform オペレーターで Automation Controller の外部データベースを設定する手順の概要を説明します。

### 前提条件

外部データベースは、Ansible Automation Platform の現在のリリースでサポートされているバージョンの PostgreSQL データベースである必要があります。



### 注記

Ansible Automation Platform 2.2 は PostgreSQL 13 をサポートします。

### 手順

外部の postgres インスタンスのクレデンシャルと接続情報はシークレットに保存する必要があります。シークレットは Automation Controller の仕様に設定されます。

1. 以下のテンプレートに従って、.yaml ファイル **postgres\_configuration\_secret** を作成します。

```
apiVersion: v1
kind: Secret
metadata:
  name: external-postgres-configuration
  namespace: <target_namespace> ❶
stringData:
  host: "<external_ip_or_url_resolvable_by_the_cluster>" ❷
  port: "<external_port>" ❸
  database: "<desired_database_name>"
  username: "<username_to_connect_as>"
  password: "<password_to_connect_with>" ❹
  sslmode: "prefer" ❺
  type: "unmanaged"
type: Opaque
```

- ❶ シークレットを作成する名前空間。これは、デプロイ先の名前空間と同じにする必要があります。
- ❷ データベースノードの解決可能なホスト名です。
- ❸ 外部ポートのデフォルトは **5432** です。
- ❹
- ❺

変数 **password** の値には、デプロイメント、バックアップ、または復元中の問題を回避するために、一重引用符 (')、二重引用符 (")、またはバックスラッシュ (\) を含めないでく

- 5 変数 **sslmode** は、**external** データベースに対してのみ有効です。使用できる値は、**prefer**、**disable**、**allow**、**require**、**verify-ca**、および **verify-full** です。

2. **oc create** コマンドを使用して、**external-postgres-configuration-secret.yml** をクラスターに適用します。

```
$ oc create -f external-postgres-configuration-secret.yml
```

3. **AutomationController** カスタムリソースオブジェクトを作成するときは、以下の例に従って、仕様にシークレットを指定します。

```
apiVersion: awx.ansible.com/v1beta1
kind: AutomationController
metadata:
  name: controller-dev
spec:
  postgres_configuration_secret: external-postgres-configuration
```

### 3.4. PVC の検索および削除

Persistent Volume Claim (PVC) は、Automation Hub および Automation Controller アプリケーションが使用するデータを保存するために使用されるストレージボリュームです。これらの PVC はアプリケーションから独立しており、アプリケーションが削除されてもそのまま残ります。PVC が必要なくなるか、PVC のバックアップを別の場所に保存している場合は手動で削除できます。

#### 手順

1. デプロイメント namespace にある既存 PVC を一覧表示します。

```
oc get pvc -n <namespace>
```

2. 以前のデプロイメント名と PVC 名を比較して、以前のデプロイメントに関連付けられている PVC を特定します。
3. 以前の PVC を削除します。

```
oc delete pvc -n <namespace> <pvc-name>
```

### 3.5. 関連情報

- OpenShift Container Platform における Operator の実行に関する詳細は、[OpenShift Container Platform 製品ドキュメント](#) に移動し、**Operator - OpenShift Container Platform での Operator の使用** をクリックしてください。

## 第4章 RED HAT OPENSIFT CONTAINER PLATFORM WEB コンソールへの AUTOMATION HUB のインストールおよび設定

これらの手順を使用して、Automation Hub Operator を Red Hat OpenShift Container Platform にインストールし、カスタムリソースを指定し、外部データベースを使用して Ansible Automation Platform をデプロイできます。



### 注記

Automation Hub のインスタンスが削除されると、PVC は自動的に削除されません。これにより、新しいデプロイメントの名前が前のデプロイメントと同じ場合は、移行中に問題が発生する可能性があります。したがって、同じ namespace に新規 Automation Hub インスタンスをデプロイする前に、古い PVC を手動で削除することが推奨されます。詳細は、[PVC の検索および削除](#) を参照してください。

### 4.1. 前提条件

- Red Hat Ansible Automation Platform オペレーターを Operator Hub にインストールしている。

### 4.2. AUTOMATION HUB OPERATOR のインストール

1. Operators → Installed Operators に移動します。
2. Automation hub エントリーを見つけて、**Create instance** をクリックします。

#### 4.2.1. Red Hat OpenShift Container Platform への Ansible Automation Platform Operator インストールのストレージオプション

Automation Hub には、複数の Pod がコレクションなどの共有コンテンツにアクセスできるように、**ReadWriteMany** ファイルベースのストレージ、Azure Blob ストレージ、または Amazon S3 準拠のストレージが必要です。

**AutomationHub** CR でオブジェクトストレージを設定するプロセスは、Amazon S3 および Azure Blob Storage に類似しています。

ファイルベースのストレージを使用し、インストールシナリオに Automation Hub が含まれる場合は、Ansible Automation Platform Operator のストレージオプションが **ReadWriteMany** に設定されていることを確認します。**ReadWriteMany** はデフォルトのストレージオプションです。

さらに、OpenShift Data Foundation は **ReadWriteMany** または S3 準拠の実装を提供します。また、**ReadWriteMany** をサポートするように NFS ストレージ設定をセットアップすることもできます。ただし、これが原因で、NFS サーバーが単一障害点となる可能性があります。

#### 関連情報

- OpenShift Container Platform [ストレージ ガイド](#) の [NFS を使用した永続ストレージ](#)
- IBM の [How do I create a storage class for NFS dynamic storage provisioning in an OpenShift environment?](#)

#### 4.2.1.1. ReadWriteMany アクセスモードを使用した OCP ストレージのプロビジョニング

Ansible Automation Platform Operator を正常にインストールするには、最初に Automation Hub のストレージタイプを **ReadWriteMany** アクセスモードにプロビジョニングする必要があります。

## 手順

1. [プロビジョニング](#) をクリックして、アクセスモードを更新します。
2. 最初のステップでは、**accessModes** をデフォルトの **ReadWriteOnce** から **ReadWriteMany** に更新します。
3. 永続ボリューム要求 (PVC) を作成するには、本セクションの追加手順を実施します。

### 4.2.1.2. Amazon S3 でのオブジェクトストレージの設定

Red Hat は、Automation Hub 用の Amazon Simple Storage Service (S3) をサポートします。**AutomationHub** カスタムリソース (CR) のデプロイ時に設定するか、既存のインスタンスに対して設定できます。

## 前提条件

- オブジェクトを保存するために Amazon S3 バケットを作成します。
- S3 バケットの名前を書き留めておきます。

## 手順

1. AWS 認証情報および接続の詳細、ならびに Amazon S3 バケットの名前を含む Kubernetes シークレットを作成します。以下の例では、**test-s3** というシークレットを作成します。

```
$ oc -n $HUB_NAMESPACE apply -f- <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: 'test-s3'
stringData:
  s3-access-key-id: $S3_ACCESS_KEY_ID
  s3-secret-access-key: $S3_SECRET_ACCESS_KEY
  s3-bucket-name: $S3_BUCKET_NAME
  s3-region: $S3_REGION
EOF
```

2. シークレットを Automation Hub カスタムリソース (CR) **spec** に追加します。

```
spec:
  object_storage_s3_secret: test-s3
```

3. このシークレットを既存のインスタンスに適用している場合は、API Pod を再起動して変更を反映します。**<hub-name>** はハブインスタンスの名前です。

```
$ oc -n $HUB_NAMESPACE delete pod -l app.kubernetes.io/name=<hub-name>-api
```

### 4.2.1.3. Azure Blob でのオブジェクトストレージの設定

Red Hat は、Automation Hub 用の Azure Blob Storage をサポートします。**AutomationHub** カスタムリソース (CR) のデプロイ時に設定するか、既存のインスタンスに対して設定できます。

### 前提条件

- オブジェクトを保存するために Azure Storage Blob コンテナを作成します。
- Blob コンテナの名前を書き留めておきます。

### 手順

1. Azure アカウントの認証情報および接続の詳細、および Azure Storage blob コンテナの名前を含む Kubernetes シークレットを作成します。以下の例では、**test-azure** という名前のシークレットを作成します。

```
$ oc -n $HUB_NAMESPACE apply -f <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: 'test-azure'
stringData:
  azure-account-name: $AZURE_ACCOUNT_NAME
  azure-account-key: $AZURE_ACCOUNT_KEY
  azure-container: $AZURE_CONTAINER
  azure-container-path: $AZURE_CONTAINER_PATH
  azure-connection-string: $AZURE_CONNECTION_STRING
EOF
```

2. シークレットを Automation Hub カスタムリソース (CR) **spec** に追加します。

```
spec:
  object_storage_azure_secret: test-azure
```

3. このシークレットを既存のインスタンスに適用している場合は、API Pod を再起動して変更を反映します。**<hub-name>** はハブインスタンスの名前です。

```
$ oc -n $HUB_NAMESPACE delete pod -l app.kubernetes.io/name=<hub-name>-api
```

## 4.2.2. Automation Hub Operator ルートオプションの設定

Red Hat Ansible Automation Platform Operator のインストール形式では、**Advanced configuration** で Automation Hub Operator ルートオプションをさらに設定できます。

1. **Advanced configuration** をクリックします。
2. **Ingress type** でドロップダウンメニューをクリックし、**Route** を選択します。
3. **Route DNS host** で、ルートの応答先となる共通のホスト名を入力します。
4. **Route TLS termination mechanism** ドロップダウンメニューをクリックし、**Edge** または **Passthrough** を選択します。
5. **Route TLS credential secret** で、ドロップダウンメニューをクリックし、一覧からシークレットを選択します。

### 4.2.3. Automation Hub Operator の Ingress タイプの設定

Red Hat Ansible Automation Platform Operator のインストールフォームでは、**Advanced configuration** で Automation Hub Operator Ingress をさらに設定できます。

#### 手順

1. **Advanced Configuration** をクリックします。
2. **Ingress type** でドロップダウンメニューをクリックし、**Ingress** を選択します。
3. **Ingress annotations** で、Ingress に追加するアノテーションを入力します。
4. **Ingress TLS secret** でドロップダウンメニューをクリックし、一覧からシークレットを選択します。

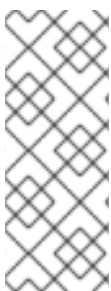
Automation Hub Operator を設定したら、フォームビューの下部にある **Create** をクリックします。Red Hat OpenShift Container Platform は Pod を作成します。これには数分の時間がかかる場合があります。

**Workloads** → **Pod** に移動し、新たに作成されたインスタンスを見つけることで、進捗を表示できます。

#### 検証

Automation Hub から Ansible Automation Platform Operator のインストールによって提供される以下の Operator Pod が実行されていることを確認します。

Operator マネージャーコントローラー	Automation Controller	Automation Hub
<p>3つの各 Operator の Operator マネージャーコントローラーには、以下が含まれます。</p> <ul style="list-style-type: none"> <li>● automation-controller-operator-controller-manager</li> <li>● automation-hub-operator-controller-manager</li> <li>● resource-operator-controller-manager</li> </ul>	<p>Automation Controller のデプロイ後に、これらの Pod が追加されていることを確認できます。</p> <ul style="list-style-type: none"> <li>● controller</li> <li>● controller-postgres</li> </ul>	<p>Automation Hub のデプロイ後に、これらの Pod が追加されていることを確認できます。</p> <ul style="list-style-type: none"> <li>● hub-api</li> <li>● hub-content</li> <li>● hub-postgres</li> <li>● hub-redis</li> <li>● hub-worker</li> </ul>



#### 注記

Pod が見つからない場合は、プルシークレットが必要であることを示している可能性があります。プルシークレットは、保護されたイメージレジストリーまたはプライベートイメージレジストリーに必要です。詳細は、[イメージプルシークレットの使用](#) を参照してください。**oc describe pod <pod-name>** を実行して、その Pod に ImagePullBackOff エラーがあるかどうかを確認することで、この問題をさらに診断できます。

### 4.3. AUTOMATION HUB ユーザーインターフェイスへのアクセス

すべての Pod が正常に起動されると、Automation Hub インターフェイスにアクセスできます。

1. **Networking** → **Routes** に移動します。
2. **Location** で、Automation Hub インスタンスの URL をクリックします。

Automation Hub のユーザーインターフェイスが起動し、Operator の設定プロセス中に指定された管理者の認証情報を使用してサインインできます。



#### 注記

設定中に管理者パスワードを指定しなかった場合は、自動的に作成されます。このパスワードを確認するには、プロジェクトに移動し、**Workloads** → **Secrets** を選択して、`controller-admin-password` を開きます。そこからパスワードをコピーして、Automation Hub のパスワードフィールドに貼り付けることができます。

### 4.4. RED HAT ANSIBLE AUTOMATION PLATFORM オペレーターでの AUTOMATION HUB 用の外部データベースの設定

外部データベースを使用して Ansible Automation Platform をデプロイすることを希望するユーザーは、インスタンスの認証情報と接続情報を使用してシークレットを設定し、**oc create** コマンドを使用してクラスターに適用するとデプロイできるようになります。

デフォルトでは、Red Hat Ansible Automation Platform オペレーターは、Ansible Automation Platform デプロイメントと同じ名前空間でマネージド PostgreSQLPod を自動的に作成および設定します。

専用のノードを使用して専用のリソースを確保する場合や、バックアップ、アップグレード、またはパフォーマンス調整を手動で管理する場合は、代わりに外部データベースを選択できます。



#### 注記

データベース名が異なる限り、Automation Hub と Automation Controller の両方に同じ外部データベース (PostgreSQL インスタンス) を使用できます。つまり、単一の PostgreSQL インスタンス内に異なる名前のデータベースを複数指定できます。

次のセクションでは、Ansible Automation Platform オペレーターで Automation Hub の外部データベースを設定する手順の概要を説明します。

#### 前提条件

外部データベースは、Ansible Automation Platform の現在のリリースでサポートされているバージョンの PostgreSQL データベースである必要があります。



#### 注記

Ansible Automation Platform 2.2 は PostgreSQL 13 をサポートします。

#### 手順

外部の postgres インスタンスのクレデンシャルと接続情報はシークレットに保存する必要があります。シークレットは Automation Hub の仕様に設定されます。

- 以下のテンプレートに従って、.yaml ファイル **postgres\_configuration\_secret** を作成します。

```

apiVersion: v1
kind: Secret
metadata:
  name: external-postgres-configuration
  namespace: <target_namespace> ❶
stringData:
  host: "<external_ip_or_url_resolvable_by_the_cluster>" ❷
  port: "<external_port>" ❸
  database: "<desired_database_name>"
  username: "<username_to_connect_as>"
  password: "<password_to_connect_with>" ❹
  sslmode: "prefer" ❺
  type: "unmanaged"
type: Opaque

```

- ❶ シークレットを作成する名前空間。これは、デプロイ先の名前空間と同じにする必要があります。
  - ❷ データベースノードの解決可能なホスト名です。
  - ❸ 外部ポートのデフォルトは **5432** です。
  - ❹ 変数 **password** の値には、デプロイメント、バックアップ、または復元中の問題を回避するために、一重引用符 (')、二重引用符 (")、またはバックスラッシュ (\) を含めないでください。
  - ❺ 変数 **sslmode** は、**external** データベースに対してのみ有効です。使用できる値は、**prefer**、**disable**、**allow**、**require**、**verify-ca**、および **verify-full** です。
- oc create** コマンドを使用して、**external-postgres-configuration-secret.yml** をクラスターに適用します。

```
$ oc create -f external-postgres-configuration-secret.yml
```

- AutomationHub** カスタムリソースオブジェクトを作成するときは、以下の例に従って、仕様にシークレットを指定します。

```

apiVersion: awx.ansible.com/v1beta1
kind: AutomationHub
metadata:
  name: hub-dev
spec:
  postgres_configuration_secret: external-postgres-configuration

```

## 4.5. PVC の検索および削除

Persistent Volume Claim (PVC) は、Automation Hub および Automation Controller アプリケーションが使用するデータを保存するために使用されるストレージボリュームです。これらの PVC はアプリケーションから独立しており、アプリケーションが削除されてもそのまま残ります。PVC が必要なくなるか、PVC のバックアップを別の場所に保存している場合は手動で削除できます。



## 手順

1. デプロイメント namespace にある既存 PVC を一覧表示します。

```
oc get pvc -n <namespace>
```

2. 以前のデプロイメント名と PVC 名を比較して、以前のデプロイメントに関連付けられている PVC を特定します。
3. 以前の PVC を削除します。

```
oc delete pvc -n <namespace> <pvc-name>
```

## 4.6. 関連情報

- OpenShift Container Platform における Operator の実行に関する詳細は、[OpenShift Container Platform 製品ドキュメント](#) に移動し、**Operator - OpenShift Container Platform での Operator の使用** をクリックしてください。

## 第5章 OPENSIFT CONTAINER PLATFORM CLI からの ANSIBLE AUTOMATION PLATFORM OPERATOR のインストール

以下の手順に従って、**oc** コマンドを使用して、OpenShift Container Platform コマンドラインインターフェイス (CLI) から Red Hat OpenShift Container Platform に Ansible Automation Platform Operator をインストールします。

### 5.1. 前提条件

- Operator をインストールするパーミッションを持つアカウントを使用して Red Hat OpenShift Container Platform にアクセスできる。
- OpenShift Container Platform CLI **oc** コマンドがローカルシステムにインストールされている。詳細は、Red Hat OpenShift Container Platform 製品ドキュメントの [OpenShift CLI のインストール](#) を参照してください。

### 5.2. OPENSIFT CONTAINER PLATFORM CLI を使用した NAMESPACE の OPERATOR へのサブスクリプション

1. Operator にプロジェクトを作成します。

```
oc new-project ansible-automation-platform
```

2. **sub.yaml** という名前のファイルを作成します。
3. 以下の YAML コードを **sub.yaml** ファイルに追加します。

```
---
apiVersion: v1
kind: Namespace
metadata:
  labels:
    openshift.io/cluster-monitoring: "true"
  name: ansible-automation-platform
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ansible-automation-platform-operator
  namespace: ansible-automation-platform
spec:
  targetNamespaces:
    - ansible-automation-platform
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ansible-automation-platform
  namespace: ansible-automation-platform
spec:
  channel: 'stable-2.2'
  installPlanApproval: Automatic
```

```

name: ansible-automation-platform-operator
source: redhat-operators
sourceNamespace: openshift-marketplace
---
apiVersion: automationcontroller.ansible.com/v1beta1
kind: AutomationController
metadata:
  name: example
  namespace: ansible-automation-platform
spec:
  replicas: 1

```

このファイルは、**ansible-automation-platform** namespace を **ansible-automation-platform-operator** Operator にサブスクライブする **ansible-automation-platform** という **Subscription** オブジェクトを作成します。

次に、**ansible-automation-platform** 名前空間に **example** という名前の **AutomationController** オブジェクトを作成します。

**example** から Automation Controller 名を変更するには、**sub.yaml** の **kind: AutomationController** セクションの **name** フィールドを編集し、**<automation\_controller\_name>** を使用する名前に置き換えてください。

```

apiVersion: automationcontroller.ansible.com/v1beta1
kind: AutomationController
metadata:
  name: <automation_controller_name>
  namespace: ansible-automation-platform

```

4. **oc apply** コマンドを実行して、**sub.yaml** ファイルで指定されたオブジェクトを作成します。

```
oc apply -f sub.yaml
```

namespace が **ansible-automation-platform-operator** Operator に正常にサブスクライブされたことを確認するには、**oc get subs** コマンドを実行します。

```
$ oc get subs -n ansible-automation-platform
```

namespace を Operator にサブスクライブする方法は、Red Hat OpenShift Container Platform **Operators** ガイドの [CLI を使用した OperatorHub からのインストール](#) を参照してください。

OpenShift Container Platform CLI を使用して、作成した Automation Controller の Web アドレスとパスワードを取得できます。

## 5.3. OPENSIFT CONTAINER PLATFORM CLI からの AUTOMATION CONTROLLER ログインの詳細の取得

Automation Controller にログインするには、Web アドレスとパスワードが必要です。

### 5.3.1. Automation Controller の Web アドレスの取得

Red Hat OpenShift Container Platform ルートは、外部クライアントが名前前でサービスに到達できるように、ホスト名でサービスを公開します。Automation Controller インスタンスを作成した時に、そのルートが作成されました。ルートは、YAML ファイルで Automation Controller オブジェクトに割り当



```
app.kubernetes.io/component: automationcontroller
app.kubernetes.io/managed-by: automationcontroller-operator
app.kubernetes.io/name: example
app.kubernetes.io/operator-version: ""
app.kubernetes.io/part-of: example
name: example-admin-password
namespace: ansible-automation-platform
resourceVersion: "185185"
uid: 39393939-5252-4242-b929-665f665f665f
```

この例では、パスワードは **88TG88TG88TG88TG88TG88TG88TG88TG** です。

## 5.4. 関連情報

- OpenShift Container Platform における Operator の実行に関する詳細は、[OpenShift Container Platform 製品ドキュメント](#) に移動し、**Operator - OpenShift Container Platform での Operator の使用** をクリックしてください。

## 第6章 AUTOMATION HUB での RED HAT SINGLE SIGN-ON OPERATOR の使用

Private Automation Hub は、認証に Red Hat Single Sign-On を使用します。

Red Hat Single Sign-On Operator はリソースを作成および管理します。この Operator を使用してカスタムリソースを作成し、OpenShift で Red Hat Single Sign-On 管理を自動化します。

- **仮想マシン (VM)** に Ansible Automation Platform をインストールする場合、インストーラーは Red Hat Single Sign-On を Private Automation Hub で使用するために自動的にインストールおよび設定できます。
- Ansible Automation Platform を Red Hat OpenShift Container Platform にインストールする場合は、Single Sign-On を個別にインストールする必要があります。

この章では、Red Hat Single Sign-On を設定し、Ansible Automation Platform が OpenShift Container Platform にインストールされている場合に Private Automation Hub と統合するプロセスについて説明します。

### 前提条件

- Operator をインストールするパーミッションを持つアカウントを使用して Red Hat OpenShift Container Platform にアクセスできる。
- Red Hat Ansible Automation Platform Operator が含まれるカタログをインストールしている。
- Red Hat Single Sign-On Operator をインストールしている。Red Hat Single Sign-On Operator をインストールするには、Red Hat Single Sign-On ドキュメントの [カスタムリソースを使用した Red Hat Single Sign-On のインストール](#) の手順に従います。

### 6.1. KEYCLOAK インスタンスの作成

Red Hat Single Sign-On Operator のインストール時に、Ansible Automation Platform で使用する Keycloak インスタンスを作成できます。

ここから、外部の Postgres を指定するか、指定しないは作成されます。

#### 手順

1. **Operator** → **Installed Operators** に移動します。
2. **rh-ss** プロジェクトを選択します。
3. **Red Hat Single Sign-On Operator** を選択します。
4. Red Hat Single Sign-On Operator の詳細ページで、**Keycloak** を選択します。
5. **Create instance** をクリックします。
6. **YAML ビュー** をクリックします。  
デフォルトの Keycloak カスタムリソースは次のとおりです。

```
apiVersion: keycloak.org/v1alpha1
kind: Keycloak
metadata:
```

```

name: example-keycloak
labels:
app: sso
namespace: aap
spec:
  externalAccess:
  enabled: true
  instances: 1

```

7. **Create** をクリックします。
8. デプロイメントが完了すると、この認証情報を使用して管理コンソールにログインできます。
9. 管理者の認証情報は、namespace の **credential-`<custom-resource>`** (keycloak の例) シークレットにあります。

## 6.2. ANSIBLE AUTOMATION PLATFORM の KEYCLOAK レルムの作成

ユーザー、認証情報、ルール、およびグループのセットを管理するレルムを作成します。ユーザーはレルムに属し、レルムにログインします。レルムは相互に分離され、制御するユーザーのみを管理および認証できます。

### 手順

1. **Operator** → **Installed Operators** に移動します。
2. **Red Hat Single Sign-On Operator** プロジェクトを選択します。
3. **Keycloak Realm** タブを選択し、**Create Keycloak Realm** をクリックします。
4. **Keycloak Realm** フォームで、**YAML view** を選択します。YAML ファイルを次のように編集します。

```

kind: KeycloakRealm
apiVersion: keycloak.org/v1alpha1
metadata:
  name: ansible-automation-platform-keycloakrealm
  namespace: rh-sso
  labels:
    app: sso
    realm: ansible-automation-platform
spec:
  realm:
    id: ansible-automation-platform
    realm: ansible-automation-platform
    enabled: true
    displayName: Ansible Automation Platform
  instanceSelector:
    matchLabels:
      app: sso

```

フィールド	説明
-------	----

<b>metadata.name</b>	設定リソース (CR) の名前のメタデータに一意の値を設定します。
<b>metadata.namespace</b>	設定リソース (CR) の名前のメタデータに一意の値を設定します。
<b>metadata.labels.app</b>	ラベルを一意の値に設定します。これは、クライアント CR を作成するときに使用されます。
<b>metadata.labels.realm</b>	ラベルを一意の値に設定します。これは、クライアント CR を作成するときに使用されます。
<b>spec.realm.id</b>	レルム名と ID を設定します。同じものを設定する必要があります。
<b>spec.realm.realm</b>	レルム名と ID を設定します。同じものを設定する必要があります。
<b>spec.realm.displayName</b>	表示する名前を設定します。

5. **Create** をクリックし、プロセスが完了するまで待ちます。

### 6.3. KEYCLOAK クライアントの作成

Keycloak クライアントは、Red Hat Single Sign-On を使用してハブユーザーを認証します。ユーザーが認証すると、リクエストは Keycloak クライアントを通過します。Single Sign-On が **OAuth** トークンを検証または発行すると、クライアントは Automation Hub に応答を提供し、ユーザーはログインできるようになります。

#### 手順

1. **Operator** → **Installed Operators** に移動します。
2. Red Hat Single Sign-On Operator プロジェクトを選択します。
3. **Keycloak Client** タブを選択し、**Create Keycloak Client** をクリックします。
4. Keycloak Realm フォームで、**YAML view** を選択します。
5. デフォルトの YAML ファイルを次のものに置き換えます。

```
kind: KeycloakClient
apiVersion: keycloak.org/v1alpha1
metadata:
  name: automation-hub-client-secret
  labels:
    app: sso
    realm: ansible-automation-platform
  namespace: rh-sso
spec:
  realmSelector:
    matchLabels:
```



```
app: sso
realm: ansible-automation-platform
client:
  name: Automation Hub
  clientId: automation-hub
  secret: <client-secret> ①
  clientAuthenticatorType: client-secret
  description: Client for automation hub
  attributes:
    user.info.response.signature.alg: RS256
    request.object.signature.alg: RS256
  directAccessGrantsEnabled: true
  publicClient: true
  protocol: openid-connect
  standardFlowEnabled: true
  protocolMappers:
    - config:
      access.token.claim: "true"
      claim.name: "family_name"
      id.token.claim: "true"
      jsonType.label: String
      user.attribute: lastName
      userinfo.token.claim: "true"
      consentRequired: false
      name: family name
      protocol: openid-connect
      protocolMapper: oidc-usermodel-property-mapper
    - config:
      userinfo.token.claim: "true"
      user.attribute: email
      id.token.claim: "true"
      access.token.claim: "true"
      claim.name: email
      jsonType.label: String
      name: email
      protocol: openid-connect
      protocolMapper: oidc-usermodel-property-mapper
      consentRequired: false
    - config:
      multivalued: "true"
      access.token.claim: "true"
      claim.name: "resource_access.${client_id}.roles"
      jsonType.label: String
      name: client roles
      protocol: openid-connect
      protocolMapper: oidc-usermodel-client-role-mapper
      consentRequired: false
    - config:
      userinfo.token.claim: "true"
      user.attribute: firstName
      id.token.claim: "true"
      access.token.claim: "true"
      claim.name: given_name
      jsonType.label: String
      name: given name
      protocol: openid-connect
```

```

protocolMapper: oidc-usermodel-property-mapper
consentRequired: false
- config:
  id.token.claim: "true"
  access.token.claim: "true"
  userinfo.token.claim: "true"
  name: full name
  protocol: openid-connect
  protocolMapper: oidc-full-name-mapper
  consentRequired: false
- config:
  userinfo.token.claim: "true"
  user.attribute: username
  id.token.claim: "true"
  access.token.claim: "true"
  claim.name: preferred_username
  jsonType.label: String
  name: <username>
  protocol: openid-connect
  protocolMapper: oidc-usermodel-property-mapper
  consentRequired: false
- config:
  access.token.claim: "true"
  claim.name: "group"
  full.path: "true"
  id.token.claim: "true"
  userinfo.token.claim: "true"
  consentRequired: false
  name: group
  protocol: openid-connect
  protocolMapper: oidc-group-membership-mapper
- config:
  multivalued: 'true'
  id.token.claim: 'true'
  access.token.claim: 'true'
  userinfo.token.claim: 'true'
  usermodel.clientRoleMapping.clientId: '{HubName}'
  claim.name: client_roles
  jsonType.label: String
  name: client_roles
  protocolMapper: oidc-usermodel-client-role-mapper
  protocol: openid-connect
- config:
  id.token.claim: "true"
  access.token.claim: "true"
  included.client.audience: '{HubName}'
  protocol: openid-connect
  name: audience mapper
  protocolMapper: oidc-audience-mapper
roles:
- name: "hubadmin"
  description: "An administrator role for automation hub"

```

**1** これを一意の値に置き換えます。

6. **Create** をクリックし、プロセスが完了するまで待ちます。

Automation Hub がデプロイされたら、[Red Hat Single Sign-On クライアントの更新](#) で説明されているように、有効なリダイレクト URI と Web オリジンでクライアントを更新する必要があります。さらに、クライアントにはトークンマッパーが事前設定されています。認証プロバイダーがグループデータを Red Hat SSO に提供しない場合は、その情報がどのように渡されるかを反映するようにグループマッピングを更新する必要があります。これは通常、ユーザー属性によるものです。

## 6.4. KEYCLOAK ユーザーの作成

この手順では、スーパー管理者権限で Automation Hub にログインできる、**hubadmin** ロールを持つ Keycloak ユーザーを作成します。

### 手順

1. **Operator** → **Installed Operators** に移動します。
2. Red Hat Single Sign-On Operator プロジェクトを選択します。
3. **Keycloak Realm** タブを選択し、**Create Keycloak User** をクリックします。
4. **Keycloak User** フォームで、**YAML view** を選択します。
5. デフォルトの YAML ファイルを次のものに置き換えます。

```
apiVersion: keycloak.org/v1alpha1
kind: KeycloakUser
metadata:
  name: hubadmin-user
  labels:
    app: sso
    realm: ansible-automation-platform
  namespace: rh-sso
spec:
  realmSelector:
    matchLabels:
      app: sso
      realm: ansible-automation-platform
  user:
    username: hub_admin
    firstName: Hub
    lastName: Admin
    email: hub_admin@example.com
    enabled: true
    emailVerified: false
    credentials:
      - type: password
        value: <ch8ngeme>
    clientRoles:
      automation-hub:
        - hubadmin
```

6. **Create** をクリックし、プロセスが完了するまで待ちます。

ユーザーが作成されると、Operator は **credential-`<realm name>`-`<username>`-`<namespace>`** という

命名パターンを使用してユーザー名とパスワードの両方を含む Secret を作成します。この例では、認証情報は **credential-ansible-automation-platform-hub-admin-rh-sso** と呼ばれます。ユーザーが作成されると、Operator はユーザーのパスワードを更新しません。パスワードの変更はシークレットに反映されません。

## 6.5. ANSIBLE AUTOMATION PLATFORM OPERATOR のインストール

### 手順

1. **Operator** → **Operator Hub** に移動し、Ansible Automation Platform Operator を検索します。
2. Ansible Automation Platform Operator プロジェクトを選択します。
3. Operator タイルをクリックします。
4. **Install** をクリックします。
5. Operator をインストールするプロジェクトを選択します。Red Hat は、Operator が推奨する名前スペース名を使用することを推奨しています。
  - a. Operator を推奨プロジェクト以外のプロジェクトにインストールする場合は、ドロップダウンメニューから **Create Project** を選択します。
  - b. プロジェクト名を入力します。
  - c. **Create** をクリックします。
6. **Install** をクリックします。
7. Operator がインストールされたら、**View Operator** をクリックします。

## 6.6. RED HAT SINGLE SIGN-ON 接続シークレットの作成

### 手順

1. **https://<sso\_host>/auth/realms/ansible-automation-platform** に移動します。
2. **public\_key** 値をコピーします。
3. OpenShift Web UI で、**Workloads** → **Secrets** に移動します。
4. **ansible-automation-platform** プロジェクトを選択します。
5. **Create** をクリックし、**From YAML** を選択します。
6. 次の YAML を編集してシークレットを作成します。

```

apiVersion: v1
kind: Secret
metadata:
  name: automation-hub-sso
  namespace: ansible-automation-platform
type: Opaque
stringData:
  keycloak_host: "keycloak-rh-sso.apps-crc.testing"

```

1

```

keycloak_port: "443"
keycloak_protocol: "https"
keycloak_realm: "ansible-automation-platform"
keycloak_admin_role: "hubadmin"
social_auth_keycloak_key: "automation-hub"
social_auth_keycloak_secret: "client-secret" 2
social_auth_keycloak_public_key: >- 3

```

- 1 この名前は、Automation Hub インスタンスを作成する際の次のステップで使用されません。
- 2 Automation Hub 用の Keycloak クライアントを作成するときにシークレットが変更した場合は、必ずこの値を変更して一致させます。
- 3 [Ansible Automation Platform Operator のインストール](#) でコピーした **public\_key** の値を入力します。

7. **Create** をクリックし、プロセスが完了するまで待ちます。

## 6.7. OPERATOR を使用した AUTOMATION HUB のインストール

Operator を使用して Automation Hub をインストールするには、以下の手順を使用します。

### 手順

1. **Operator** → **Installed Operators** に移動します。
2. **Ansible Automation Platform** を選択します。
3. **Automation Hub** タブを選択し、**Automation Hub** の作成をクリックします。
4. **YAML view** を選択します。YAML は次のようになります。

```

apiVersion: automationhub.ansible.com/v1beta1
kind: AutomationHub
metadata:
  name: private-ah 1
  namespace: ansible-automation-platform
spec:
  sso_secret: automation-hub-sso 2
  pulp_settings:
    verify_ssl: false
  route_tls_termination_mechanism: Edge
  ingress_type: Route
  loadbalancer_port: 80
  file_storage_size: 100Gi
  image_pull_policy: IfNotPresent
  web:
    replicas: 1
  file_storage_access_mode: ReadWriteMany
  content:
    log_level: INFO
    replicas: 2
  postgres_storage_requirements:

```

```

limits:
  storage: 50Gi
requests:
  storage: 8Gi
api:
  log_level: INFO
  replicas: 1
postgres_resource_requirements:
  limits:
    cpu: 1000m
    memory: 8Gi
  requests:
    cpu: 500m
    memory: 2Gi
loadbalancer_protocol: http
resource_manager:
  replicas: 1
worker:
  replicas: 2

```

- 1 インスタンスに使用する名前に `metadata.name` を設定します。
- 2 `spec.sso_secret` を、[Red Hat Single Sign On 接続の詳細](#)を保持するためにシークレットを [作成](#) で作成したシークレットの名前に設定します。



### 注記

この YAML は SSL 検証をオフにします (**`ssl_verify: false`**)。OpenShift に自己署名証明書を使用していない場合は、この設定を削除できます。

5. **Create** をクリックし、プロセスが完了するまで待ちます。

## 6.8. AUTOMATION HUB ルートの決定

ハブルートを決定するには、次の手順を使用します。

### 手順

1. **Networking** → **Routes** に移動します。
2. インストールに使用したプロジェクトを選択します。
3. **private-ah-web-svc** サービスの場所をコピーします。Automation Hub インスタンスの作成時に別の名前を使用した場合は、サービスの名前が異なります。これは、後で Red Hat Single Sign-On クライアントを更新するために使用されます。

## 6.9. RED HAT SINGLE SIGN-ON クライアントの更新

Automation Hub がインストールされ、インスタンスの URL がわかっている場合は、Red Hat Single Sign-On を更新して、有効なリダイレクト URI と Web オリジンの設定を設定する必要があります。

### 手順

1. **Operator** → **Installed Operators** に移動します。
2. RH-SSO プロジェクトを選択します。
3. **Red Hat Single Sign-On Operator** をクリックします。
4. **Keycloak Client** を選択します。
5. Automation-hub-client-secret クライアントをクリックします。
6. **YAML** を選択します。
7. クライアント YAML を更新して、有効なリダイレクト URI と Web オリジンの設定を追加します。

```

redirectUris:
  - 'https://private-ah-ansible-automation-platform.apps-crc.testing/*'
webOrigins:
  - 'https://private-ah-ansible-automation-platform.apps-crc.testing'

```

フィールド	説明
<b>redirectURIs</b>	これは、Automation Hub ルートの決定で決定された場所です。/* を <b>redirectUris</b> 設定の末尾に必ず追加してください。
<b>webOrigins</b>	これは、Automation Hub ルートの決定で決定された場所です。



#### 注記

これらの設定を入力するときは、インデントが正しいことを確認してください。

8. **Save** をクリックします。

#### 接続を確認するには

1. Automation Hub ルートに移動します。
2. **hub\_admin** ユーザー認証情報を入力してサインインします。
3. Red Hat Single Sign-On は認証を処理し、Automation Hub にリダイレクトします。

## 6.10. 関連情報

- OpenShift Container Platform でのオペレーターの実行に関する詳細は、OpenShift Container Platform 製品ドキュメントの [OpenShift Container Platform での Operator の使用](#) を参照してください。

## 第7章 RED HAT ANSIBLE AUTOMATION PLATFORM の ANSIBLE AUTOMATION PLATFORM OPERATOR への移行

Red Hat Ansible Automation Platform デプロイメントを Ansible Automation Platform Operator に移行すると、Kubernetes ネイティブ Operator が提供する利点を活用できます。これには、Red Hat Ansible Automation Platform デプロイメントの簡単なアップグレードや完全なライフサイクルサポートが含まれます。

以下の手順を使用して、以下のデプロイのいずれかを Ansible Automation Platform Operator に移行します。

- Ansible Tower 3.8.6、Automation Controller、または Automation Hub の仮想マシンベースのインストール
- Ansible Tower 3.8.6 の Openshift インスタンス (Ansible Automation Platform 1.2)

### 7.1. 移行に関する考慮事項

OpenShift Container Platform 3 で Ansible Automation Platform 1.2 から OpenShift Container Platform 4 の Ansible Automation Platform 2.x にアップグレードする場合は、新規の OpenShift Container Platform バージョン 4 クラスターをプロビジョニングし、Ansible Automation Platform を新規クラスターに移行する必要があります。

### 7.2. 移行の準備

現在の Ansible Automation Platform デプロイメントを Ansible Automation Platform Operator に移行する前に、既存のデータのバックアップを作成し、シークレットキーおよび postgresql 設定の k8s シークレットを作成します。



#### 注記

Automation Controller と Automation Hub インスタンスの両方を移行する場合は、[秘密鍵のシークレットの作成](#)と [postgresql 設定シークレットの作成](#)の手順を繰り返して、[データの Ansible Automation Platform Operator への移行](#)に進みます。

#### 7.2.1. 前提条件

Ansible Automation Platform デプロイメントを Ansible Automation Platform Operator に移行するには、以下が必要です。

- 秘密鍵のシークレット
- PostgreSQL の設定
- 新しい OpenShift クラスターの namespace のロールベースアクセス制御
- 新しい OpenShift クラスターは、以前の PostgreSQL データベースに接続可能である必要があります。





## 注記

秘密鍵情報は、Red Hat Ansible Automation Platform の初回インストール時に作成されたインベントリーファイルにあります。秘密鍵を覚えていない場合や、インベントリーファイルを見つけられない場合は、Red Hat カスタマーポータルから [Ansible サポート](#) にお問い合わせください。

Ansible Automation Platform 2.x 以前からデータを移行する前に、失われた防止のためにデータのバックアップを作成する必要があります。データをバックアップするには、以下の手順を実施します。

### 手順

1. 現在のデプロイメントプロジェクトにログインします。
2. `setup.sh` を実行して、現在のデータ/デプロイメントのバックアップを作成します。  
バージョン 2.x 以前のオンプレミスデプロイメントの場合:

```
$ ./setup.sh -b
```

バージョン 2.0 より前の OpenShift デプロイメントの場合 (operator デプロイメント以外):

```
./setup_openshift.sh -b
```

### 7.2.2. 秘密鍵のシークレットの作成

データを OpenShift Container Platform の Ansible Automation Platform Operator に移行するには、初回インストール時にインベントリーファイルに定義された秘密鍵に一致する秘密鍵を作成する必要があります。そうしないと、移行されたデータは暗号化され、移行後に使用できなくなります。

### 手順

1. 以前のインストールで AAP のデプロイに使用したインベントリーファイルで古い秘密鍵を見つけます。
2. 秘密鍵の yml ファイルを作成します。

```
apiVersion: v1
kind: Secret
metadata:
  name: <resourcename>-secret-key
  namespace: <target-namespace>
stringData:
  secret_key: <old-secret-key>
type: Opaque
```

3. 秘密鍵 yml をクラスターに適用します。

```
oc apply -f <secret-key.yml>
```

### 7.2.3. postgresql 設定シークレットの作成

移行を成功させるには、既存のデプロイメントのデータベースへのアクセスを提供する必要があります。

## 手順

1. postgresql 設定シークレットの yml ファイルを作成します。

```

apiVersion: v1
kind: Secret
metadata:
  name: <resourcename>-old-postgres-configuration
  namespace: <target namespace>
stringData:
  host: "<external ip or url resolvable by the cluster>"
  port: "<external port, this usually defaults to 5432>"
  database: "<desired database name>"
  username: "<username to connect as>"
  password: "<password to connect with>"
type: Opaque

```

2. postgresql 設定の yml をクラスターに適用します。

```
oc apply -f <old-postgres-configuration.yml>
```

### 7.2.4. ネットワーク接続の確認

データの移行を成功させるには、新しい Operator デプロイメントから以前のデプロイメントデータベースへのネットワーク接続があることを確認します。

#### 前提条件

既存のデプロイメントのホストおよびポート情報を書き留めておきます。この情報は、conf.d ディレクトリーにある postgres.py ファイルにあります。

## 手順

1. yml ファイルを作成して、新しいデプロイメントと以前のデプロイメントデータベース間の接続を確認します。

```

apiVersion: v1
kind: Pod
metadata:
  name: dbchecker
spec:
  containers:
  - name: dbchecker
    image: registry.redhat.io/rhel8/postgresql-12:latest
    command: ["sleep"]
    args: ["600"]

```

2. 接続チェッカーの yml ファイルを新規プロジェクトのデプロイメントに適用します。

```

oc project ansible-automation-platform
oc apply -f connection_checker.yml

```

3. 接続チェッカー Pod が実行されていることを確認します。

```
oc get pods
```

- Pod シェルに接続します。

```
oc rsh dbchecker
```

- Pod でシェルセッションが開いたら、新規プロジェクトが古いプロジェクトクラスターに接続できることを確認します。

```
pg_isready -h <old-host-address> -p <old-port-number> -U awx
```

#### 例

```
<old-host-address>:<old-port-number> - accepting connections
```

## 7.3. ANSIBLE AUTOMATION PLATFORM OPERATOR へのデータの移行

秘密鍵 postgresql 認証情報を設定し、ネットワーク接続を検証し、Ansible Automation Platform Operator をインストールした後に、データを移行する前にカスタムリソースコントローラーオブジェクトを作成する必要があります。

### 7.3.1. AutomationController オブジェクトの作成

以下の手順を使用して AutomationController カスタムリソースオブジェクトを作成します。

#### 手順

- Red Hat OpenShift Container Platform にログインします。
- Operators → Installed Operators に移動します。
- プロジェクトの namespace にインストールされている Ansible Automation Platform Operator を選択します。
- Automation Controller タブを選択します。
- Create AutomationController をクリックします。
- 新しいデプロイメントの名前を入力します。
- Advanced configurations で、secret key secret と postgres configuration secret を選択します。
- Create をクリックします。

### 7.3.2. AutomationHub オブジェクトの作成

AutomationHub カスタムリソースオブジェクトを作成するには、以下の手順を使用します。

#### 手順

- Red Hat OpenShift Container Platform にログインします。

2. **Operators** → **Installed Operators** に移動します。
3. プロジェクトの namespace にインストールされている Ansible Automation Platform Operator を選択します。
4. **Automation Hub** タブを選択します。
5. **Create AutomationHub** をクリックします。
6. 新しいデプロイメントの名前を入力します。
7. **Advanced configurations** で、[secret key secret](#) と [postgres configuration secret](#) を選択します。
8. **Create** をクリックします。

## 7.4. 移行後のクリーンアップ

データの移行が完了したら、不要になった InstanceGroups を削除する必要があります。

### 手順

1. 移行時に作成したパスワードを使用して、管理者として Red Hat Ansible Automation Platform にログインします。

注記：移行時に管理者パスワードを作成しなかった場合、パスワードは自動的に作成されています。このパスワードを確認するには、プロジェクトに移動し、**Workloads** → **Secrets** を選択して、`controller-admin-password` を開きます。そこから、パスワードをコピーして、Red Hat Ansible Automation Platform password フィールドに貼り付けることができます。

1. **Administration** → **InstanceGroups** を選択します。
2. コントロールプレーンおよび default 以外のすべての InstanceGroups を選択します。
3. **Delete** をクリックします。

## 第8章 OPENSIFT CONTAINER PLATFORM での ANSIBLE AUTOMATION PLATFORM OPERATOR のアップグレード

Ansible Automation Platform Operator は、OpenShift Container Platform 環境での新しい Red Hat Ansible Automation Platform インスタンスのインストール、アップグレード、およびデプロイメントを簡素化します。

### 8.1. アップグレードに関する考慮事項

Red Hat Ansible Automation Platform バージョン 2.0 は、Ansible Automation Platform Operator の最初のリリースです。バージョン 2.0 からアップグレードする場合は、[Ansible Automation Platform Operator のアップグレード](#) の手順に進みます。

アップグレード先の Red Hat Ansible Automation Platform のバージョンでサポートされていないバージョンの OpenShift Container Platform を使用している場合は、アップグレードを行う前に、OpenShift Container Platform クラスタをサポートされているバージョンにアップグレードする必要があります。

必要な OpenShift Container Platform のバージョンを判断するには、[Red Hat Ansible Automation Platform のライフサイクル](#) を参照してください。

クラスタのアップグレードに関する詳細は、[クラスタの更新](#) を参照してください。

### 8.2. 前提条件

新しいバージョンの Ansible Automation Platform Operator にアップグレードするには、以下を行うことが推奨されます。

- AutomationControllerBackup および AutomationHubBackup オブジェクトを作成します。
- アップグレードする新しい Ansible Automation Platform バージョンと中間バージョンのリリースノートを確認してください。

### 8.3. ANSIBLE AUTOMATION PLATFORM OPERATOR のアップグレード

OpenShift Container Platform で最新バージョンの Ansible Automation Platform Operator にアップグレードするには、以下を実行します。

#### 手順

1. OpenShift Container Platform にログインします。
2. **Operators** → **Installed Operators** に移動します。
3. **Subscriptions** タブを選択します。
4. **Upgrade status** で、**Upgrade Available** をクリックします。
5. **Preview InstallPlan** をクリックします。
6. **Approve** をクリックします。

