



# Red Hat Ansible Automation Platform 2.4

## Ansible plug-ins for Red Hat Developer Hub の インストール

Ansible plug-ins for Red Hat Developer Hub のインストールと設定



# Red Hat Ansible Automation Platform 2.4 Ansible plug-ins for Red Hat Developer Hub のインストール

---

Ansible plug-ins for Red Hat Developer Hub のインストールと設定

## 法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

Linux<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

このガイドでは、ユーザーが Ansible について学習し、厳選されたコレクションの詳細を理解して、自動化プロジェクトを開発できるように、Ansible plug-ins for Red Hat Developer Hub をインストールして設定する方法を説明します。

---

## 目次

はじめに .....	3
RED HAT ドキュメントへのフィードバック (英語のみ) .....	4
<b>第1章 ANSIBLE PLUG-INS FOR RED HAT DEVELOPER HUB .....</b>	<b>5</b>
1.1. RED HAT DEVELOPER HUB .....	5
1.2. ANSIBLE PLUG-INS FOR RED HAT DEVELOPER HUB .....	5
1.3. アーキテクチャー .....	5
<b>第2章 ANSIBLE プラグインの OPENSIFT CONTAINER PLATFORM へのインストール .....</b>	<b>7</b>
2.1. 前提条件 .....	7
2.2. ANSIBLE プラグインファイルのダウンロード .....	7
2.3. プラグインレジストリーの作成 .....	8
2.4. ANSIBLE プラグイン設定の追加 .....	9
2.5. ANSIBLE DEVELOPMENT TOOLS サイドカーコンテナの追加 .....	10
2.6. ANSIBLE DEV TOOLS SERVER の設定 .....	12
2.7. ANSIBLE AUTOMATION PLATFORM の詳細の設定 .....	13
2.8. ANSIBLE プラグインサブスクリプションの警告メッセージ .....	13
<b>第3章 OPENSIFT CONTAINER PLATFORM での ANSIBLE プラグインのアップグレード .....</b>	<b>17</b>
3.1. ANSIBLE プラグインファイルのダウンロード .....	17
3.2. プラグインレジストリーの更新 .....	17
3.3. ANSIBLE プラグインのバージョン番号の更新 .....	18
<b>第4章 ANSIBLE プラグインの OPENSIFT CONTAINER PLATFORM からのアンインストール .....</b>	<b>20</b>
4.1. アンインストール .....	20
<b>第5章 ANSIBLE PLUG-INS FOR RED HAT DEVELOPER HUB の設定 .....</b>	<b>22</b>
5.1. RED HAT DEVELOPER HUB 認証の有効化 .....	22
5.2. ロールベースアクセス制御の設定 .....	22
5.3. ANSIBLE プラグインのオプションのインテグレーションの設定 .....	22
5.4. RED HAT DEVELOPER HUB データ TELEMETRY のキャプチャー .....	24



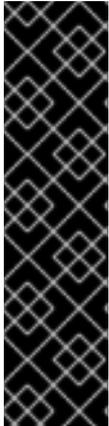
## はじめに

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。

このガイドでは、Ansible plug-ins for Red Hat Developer Hub をインストールする方法を説明します。このガイドの更新により、Ansible Automation Platform の最新リリースの情報が追加されました。

## 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。



### 重要

Ansible plug-ins for Red Hat Developer Hub はテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能では、近々発表予定の製品機能をリリースに先駆けて提供します。これにより、お客様は機能性をテストし、開発プロセス中にフィードバックを提供することができます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

# 第1章 ANSIBLE PLUG-INS FOR RED HAT DEVELOPER HUB

## 1.1. RED HAT DEVELOPER HUB

Red Hat Developer Hub (RHDH) は、開発者ポータルをビルドするために設計されたオープンな開発者プラットフォームです。

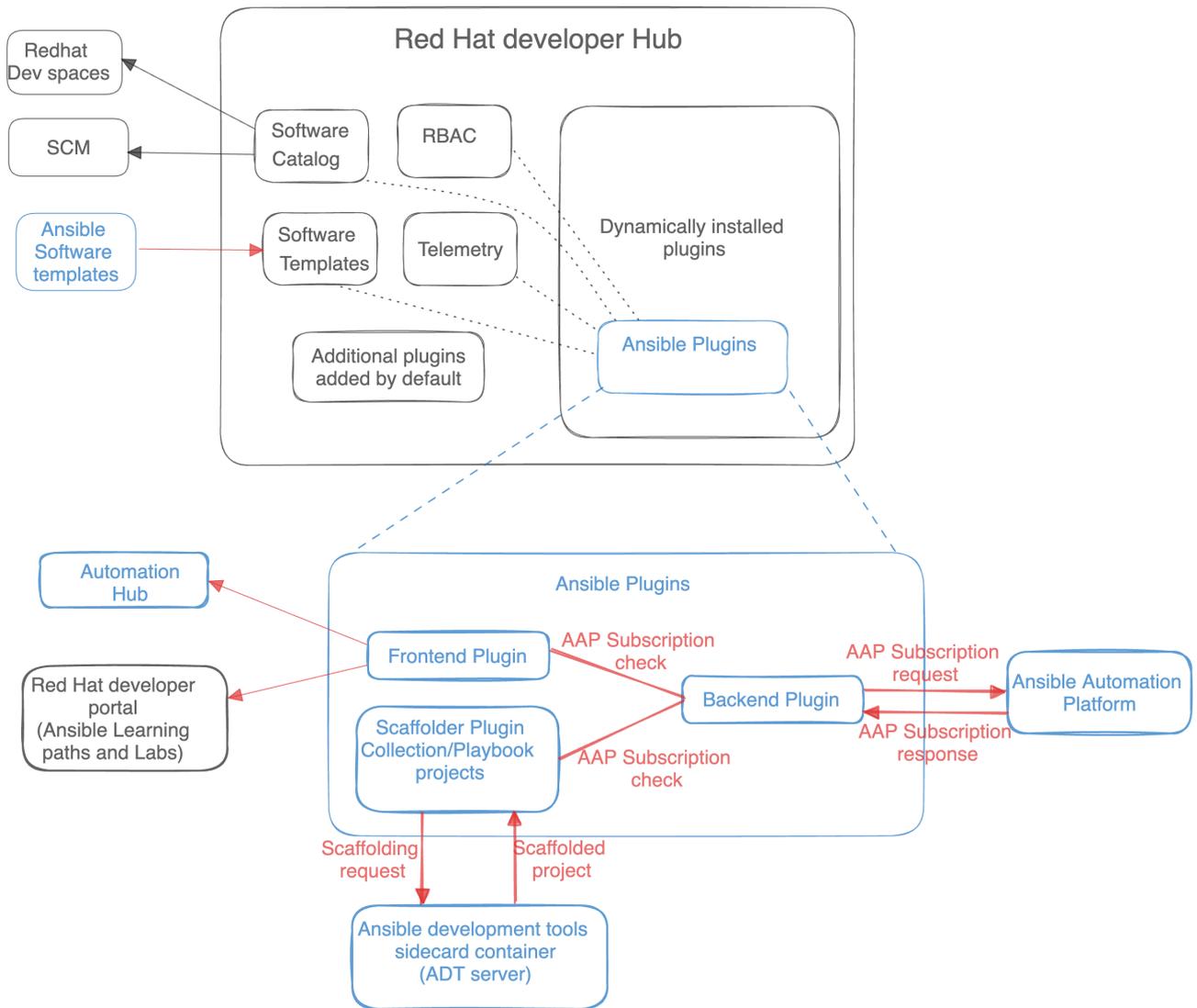
## 1.2. ANSIBLE PLUG-INS FOR RED HAT DEVELOPER HUB

Ansible plug-ins for Red Hat Developer Hub は、Ansible ファーストの Red Hat Developer Hub のユーザーエクスペリエンスを提供し、すべてのスキルレベルの Ansible ユーザーの自動化エクスペリエンスを簡素化します。Ansible プラグインは、Ansible 学習者のオンボーディングを加速し、組織全体で Ansible ユースケースの導入を効率化するための厳選されたコンテンツと機能を提供します。

Ansible プラグインは以下を提供します。

- Ansible ユーザー向けにカスタマイズされたホームページとナビゲーション。
- Ansible を初めて使用するユーザーを支援するために、厳選された Ansible ラーニングパス。
- ベストプラクティスに従う Ansible Playbook およびコレクションプロジェクトを作成するためのソフトウェアテンプレート。
- 独自の設定を持つ、サポートされている開発環境とツールへのリンク。

## 1.3. アーキテクチャー



## 第2章 ANSIBLE プラグインの OPENSIFT CONTAINER PLATFORM へのインストール

以下の手順では、Red Hat OpenShift Container Platform の Red Hat Developer Hub インスタンスに Ansible プラグインをインストールする方法を説明します。

### 2.1. 前提条件

- Helm チャートを使用して Red Hat OpenShift Container Platform に Red Hat Developer Hub をインストールしている。[Red Hat Developer Hub 管理ガイドの Helm チャートを使用して OpenShift Container Platform に Red Hat Developer Hub をデプロイする](#) セクションの手順に従ってください。
- Red Hat Ansible Automation Platform への有効なサブスクリプション。
- OpenShift Container Platform 管理者ロールを持つユーザーが、アプリケーションを作成するためにプロジェクト内で適切なロールと権限を設定している。
- Red Hat Developer Hub インスタンスは、Automation Controller API をクエリーできます。

### 2.2. ANSIBLE プラグインファイルのダウンロード

1. [Red Hat Ansible Automation Platform 製品ソフトウェアのダウンロードページ](#) から、プラグインの最新の `.tar` ファイルをダウンロードします。ファイル名の形式は `ansible-backstage-rhaap-bundle-x.y.z.tar.gz` です。Ansible プラグインリリースバージョン (`1.0.0` など) を、`x.y.z` に置き換えます。
2. `.tar` ファイルを保存するディレクトリーをローカルマシンに作成します。

```
$ mkdir /path/to/<ansible-backstage-plugins-local-dir-changeme>
```

3. ディレクトリーパスを表す環境変数 (`$DYNAMIC_PLUGIN_ROOT_DIR`) を設定します。

```
$ export DYNAMIC_PLUGIN_ROOT_DIR=/path/to/<ansible-backstage-plugins-local-dir-changeme>
```

4. `ansible-backstage-rhaap-bundle-<version-number>.tar.gz` の内容を `$DYNAMIC_PLUGIN_ROOT_DIR` に展開します。

```
$ tar --exclude="*code*" -xzf ansible-backstage-rhaap-bundle-x.y.z.tar.gz -C  
$DYNAMIC_PLUGIN_ROOT_DIR
```

Ansible プラグインリリースバージョン (`1.0.0` など) を、`x.y.z` に置き換えます。

#### 検証

`ls` を実行して、展開されたファイルが `$DYNAMIC_PLUGIN_ROOT_DIR` ディレクトリーにあることを確認します。

```
$ ls $DYNAMIC_PLUGIN_ROOT_DIR  
ansible-plugin-backstage-rhaap-x.y.z.tgz  
ansible-plugin-backstage-rhaap-x.y.z.tgz.integrity  
ansible-plugin-backstage-rhaap-backend-x.y.z.tgz
```

```
ansible-plugin-backstage-rhaap-backend-x.y.z.tgz.integrity
ansible-plugin-scaffolder-backend-module-backstage-rhaap-x.y.z.tgz
ansible-plugin-scaffolder-backend-module-backstage-rhaap-x.y.z.tgz.integrity
```

**.integrity** ファイルタイプのファイルには、プラグインの SHA 値が含まれています。SHA 値は、プラグインの設定時に使用されます。

## 2.3. プラグインレジストリーの作成

Ansible プラグインをホストするために OpenShift クラスターでプラグインレジストリーをセットアップし、Red Hat Developer Hub でのインストール用に利用できるようにします。

### 手順

1. 認証情報を使用して OpenShift Container Platform インスタンスにログインし、新しいアプリケーションを作成します。
2. Red Hat Developer Hub OpenShift プロジェクトを開きます。

```
$ oc project <YOUR_DEVELOPER_HUB_PROJECT>
```

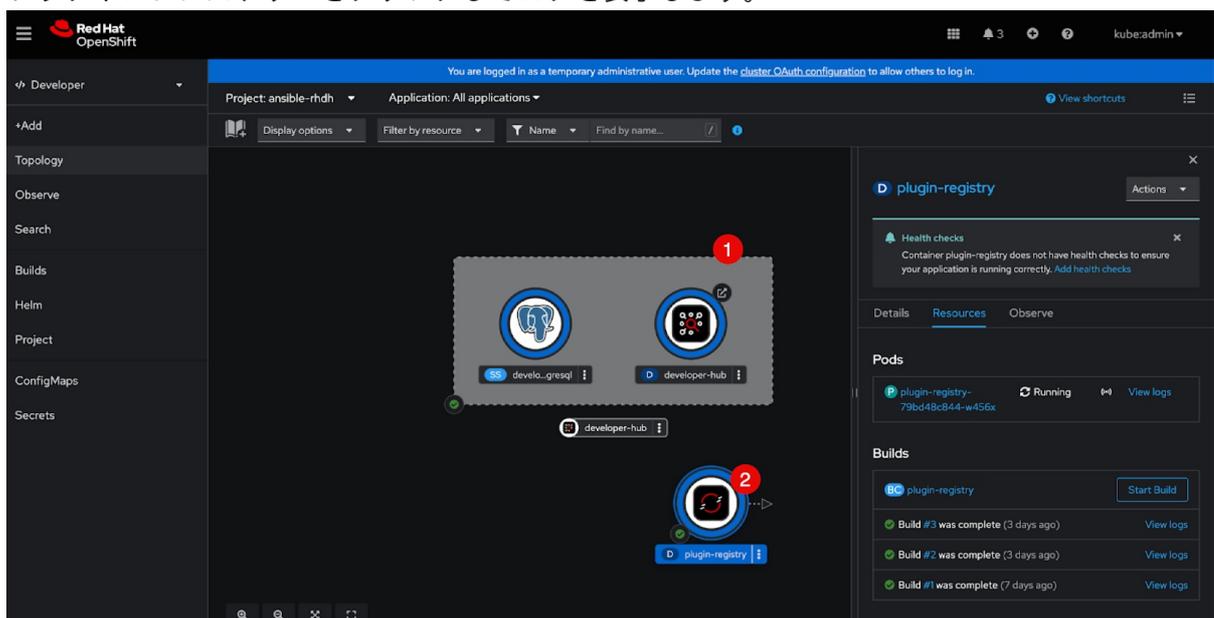
3. 以下のコマンドを実行して、OpenShift クラスターにプラグインレジストリービルドを作成します。

```
$ oc new-build httpd --name=plugin-registry --binary
$ oc start-build plugin-registry --from-dir=$DYNAMIC_PLUGIN_ROOT_DIR --wait
$ oc new-app --image-stream=plugin-registry
```

### 検証

plugin-registry が正常にデプロイされたことを確認するには、OpenShift Web コンソールで Red Hat Developer Hub アプリケーションの Developer パースペクティブを開きます。

1. プラグインレジストリーをクリックしてログを表示します。



(1) Developer Hub インスタンス

## (2) プラグインレジストリー

2. ターミナルのタブをクリックし、コンテナにログインします。
3. ターミナルで **ls** を実行して、.tar ファイルがプラグインレジストリーにあることを確認します。

```
sh-4.45 $ls -l
total 3392
-rw-rw-r-- 1 default root 3376269 Jul 22 13:48 ansible-plugin-backstage-rhaap-1.0.0.tgz
-rw-rw-r-- 1 default root 25156 Jul 22 13:48 ansible-plugin-backstage-rhaap-backend-1.0.0.tgz
-rw-rw-r-- 1 default root 61520 Jul 22 13:48 ansible-plugin-scaffolder-backend-module-backstage-rhaap-1.0.0.tgz
```

バージョン番号とファイル名は異なる場合があります。

## 2.4. ANSIBLE プラグイン設定の追加

1. OpenShift Developer UI で、**Helm** → **developer-hub** → **Actions** → **Upgrade** → **Yaml view** に移動します。
2. Helm チャート設定を更新して、RHDH インスタンスに動的プラグインを追加します。YAML ファイルの **plugins** セクションで、有効にする動的プラグインを追加します。

```
...
global:
...
  plugins:
    - disabled: false
      integrity: <SHA512 value>
      package: 'http://plugin-registry:8080/ansible-plugin-backstage-rhaap-x.y.z.tgz'
      pluginConfig:
        dynamicPlugins:
          frontend:
            ansible.plugin-backstage-rhaap:
              applcons:
                - importName: AnsibleLogo
                  name: AnsibleLogo
              dynamicRoutes:
                - importName: AnsiblePage
                  menuItem:
                    icon: AnsibleLogo
                    text: Ansible
                    path: /ansible
            - disabled: false
              integrity: <SHA512 value>
              package: >-
                http://plugin-registry:8080/ansible-plugin-scaffolder-backend-module-backstage-rhaap-x.y.z.tgz
              pluginConfig:
                dynamicPlugins:
                  backend:
                    ansible.plugin-scaffolder-backend-module-backstage-rhaap: null
            - disabled: false
```

```

integrity: <SHA512 value>
package: >-
  http://plugin-registry:8080/ansible-plugin-backstage-rhaap-backend-x.y.z.tgz
pluginConfig:
dynamicPlugins:
  backend:
    ansible.plugin-backstage-rhaap-backend: null

```

3. **パッケージ** セクションで、プラグインファイル名の **x.y.z** を、Ansible プラグインの正しいバージョン番号に置き換えます。
4. 各 Ansible プラグインの対応する **.integrity** ファイルの内容を使用して、インテグリティの値を更新します。
5. **Upgrade** をクリックします。  
開発者ハブ Pod が再起動し、プラグインがインストールされます。

## 検証

プラグインがインストールされていることを確認します。

1. OpenShift Web コンソールで Red Hat Developer Hub アプリケーションの Developer パースペクティブを開きます。
2. **install-dynamic-plugin** コンテナログを確認します。Ansible プラグインは Red Hat Developer Hub に表示されます。

```

174 ***** Skipping disabled dynamic plugin ./dynamic-plugins/dist/backstage-plugin-lighthouse
175
176 ***** Skipping disabled dynamic plugin ./dynamic-plugins/dist/backstage-plugin-tech-zader
177
178 ***** Skipping disabled dynamic plugin ./dynamic-plugins/dist/janus-ldp-backstage-plugin-analytics-provider-segment
179
180 ***** Installing dynamic plugin http://plugin-registry:8080/janus-ldp-backstage-plugin-ansible-0.0.1.tgz
181 ==> Grabbing package archive through npm pack
182 ==> Removing previous plugin directory /dynamic-plugins-root/janus-ldp-backstage-plugin-ansible-0.0.1
183 ==> Extracting package archive /dynamic-plugins-root/janus-ldp-backstage-plugin-ansible-0.0.1.tgz
184 ==> Removing package archive /dynamic-plugins-root/janus-ldp-backstage-plugin-ansible-0.0.1.tgz
185 ==> Merging plugin-specific configuration
186 ==> Successfully installed dynamic plugin http://plugin-registry:8080/janus-ldp-backstage-plugin-ansible-0.0.1.tgz
187
188 ***** Installing dynamic plugin http://plugin-registry:8080/janus-ldp-backstage-plugin-scaffolder-backend-module-ansible-0.0.0.tgz
189 ==> Grabbing package archive through npm pack
190 ==> Removing previous plugin directory /dynamic-plugins-root/janus-ldp-backstage-plugin-scaffolder-backend-module-ansible-0.0.0
191 ==> Extracting package archive /dynamic-plugins-root/janus-ldp-backstage-plugin-scaffolder-backend-module-ansible-0.0.0.tgz
192 ==> Removing package archive /dynamic-plugins-root/janus-ldp-backstage-plugin-scaffolder-backend-module-ansible-0.0.0.tgz
193 ==> Merging plugin-specific configuration
194 ==> Successfully installed dynamic plugin http://plugin-registry:8080/janus-ldp-backstage-plugin-scaffolder-backend-module-ansible-0.0.0.tgz

```

## 2.5. ANSIBLE DEVELOPMENT TOOLS サイドカーコンテナの追加

プラグインがロードされたら、Red Hat Developer Hub Pod に Ansible Development Container (**ansible-devtools-server**) をサイドカーコンテナとして追加します。

### 2.5.1. Red Hat Developer Hub Helm 設定へのプルシークレットの追加

#### 前提条件

Ansible Development Container のダウンロードには、Red Hat カスタマーポータルアカウントおよび Red Hat Service Registry アカウントが必要です。

## 手順

1. 必要に応じて、新しい [Red Hat Registry Service アカウント](#) を作成します。
2. **Account name** 列の下にあるトークン名をクリックします。
3. **OpenShift Secret** タブを選択し、指示に従ってプルシークレットを Red Hat Developer Hub OpenShift プロジェクトに追加します。
4. 新しいシークレットを Red Hat Developer Hub の Helm 設定に追加します。

```
upstream:
  backstage: |
    ...
  image:
    ...
  pullSecrets:
    - rhdh-secret-registry
    ...
```

詳細は、[Red Hat Container Registry のドキュメント](#) を参照してください。

### 2.5.2. Ansible Developer Tools コンテナの追加

コンテナを追加するには、Helm チャート設定を更新する必要があります。

## 手順

1. OpenShift UI にログインします。
2. **Helm** → **developer-hub** → **Actions** → **upgrade** → **Yaml view** に移動し、Helm チャートを開きます。
3. YAML ファイルの **extraContainers** セクションを更新します。  
次のコードを追加します。

```
upstream:
  backstage: |
    ...
  extraContainers:
    - command:
      - adt
      - server
      image: >-
        registry.redhat.io/ansible-automation-platform-25/ansible-dev-tools-rhel8:latest
      imagePullPolicy: IfNotPresent
      name: ansible-devtools-server
      ports:
        - containerPort: 8000
  image:
    pullPolicy: Always
    pullSecrets:
```

```
- ...
- rhdh-secret-registry
...
```



### 注記

イメージプルポリシーは **imagePullPolicy: IfNotPresent** です。イメージは、そのイメージがノード上にない場合にのみプルされます。常に最新のイメージを使用する場合は、**imagePullPolicy: Always** に更新してください。

4. **Upgrade** をクリックします。

### 検証

コンテナが実行中であることを確認するには、コンテナログを確認します。

The screenshot shows the 'Logs' tab for a pod. The pod is named 'developer-hub-stable-7b885ffc58-z94r7' and is in a 'Running' state. The logs show the following output:

```
1 [2024-07-23 06:37:24 +0000] [1] [INFO] Starting gunicorn 22.0.0
2 [2024-07-23 06:37:24 +0000] [1] [INFO] Listening at: http://0.0.0.0:8000 (1)
3 [2024-07-23 06:37:24 +0000] [1] [INFO] Using worker: sync
4 [2024-07-23 06:37:24 +0000] [7] [INFO] Booting worker with pid: 7
5 Note: ansible project created at
6 /tmp/tmpns47apin/my-galaxy-username-rhel-firewall-config
7 Note: ansible project created at
8 /tmp/tmpxakf1ree/my-galaxy-username-rhel-firewall-config
9 Note: ansible project created at
10 /tmp/tmpfjh8xy59/my-galaxy-username-rhel-firewall-config
11 Note: ansible project created at
12 /tmp/tmp7od35ut0/my-galaxy-username-rhel-firewall-config
```

## 2.6. ANSIBLE DEV TOOLS SERVER の設定

Ansible プラグインが提供されるソフトウェアテンプレートを使用して新しいプロジェクトをプロビジョニングするには、**creatorService** URL が必要です。

### 手順

1. カスタム Red Hat Developer Hub config map (例: **app-config-rhdh**) を編集します。
2. 以下のコードを Red Hat Developer Hub の **app-config-rhdh.yaml** ファイルに追加します。

```
data:
  app-config-rhdh.yaml: |-
  ansible:
    creatorService:
      baseUrl: 127.0.0.1
      port: '8000'
```

## 2.7. ANSIBLE AUTOMATION PLATFORM の詳細の設定

Ansible プラグインはトークンを使用して、コントローラー API で Ansible Automation Platform サブスクリプションステータスをクエリーします。



### 注記

Ansible プラグインは、Ansible Automation Platform のサブスクリプションステータスに関係なく引き続き機能します。

### 手順

1. **Automation Controller ユーザーガイド**の [アプリケーション](#) セクションに従って、Automation Controller で「読み取り」スコープを持つ Personal Access Token (PAT) を作成します。
2. カスタム Red Hat Developer Hub config map (例: **app-config-rhdh**) を編集します。
3. Ansible Automation Platform の詳細を **app-config-rhdh.yaml** に追加します。
  - a. Automation Controller URL で **baseURL** キーを設定します。
  - b. 手順1で作成したトークン値を使用して **token** キーを設定します。
  - c. **checkSSL** キーを **true** または **false** に設定します。  
**checkSSL** が **true** に設定されている場合、Ansible プラグインは SSL 証明書が有効であるかどうかを確認します。

```
data:
  app-config-rhdh.yaml: |
    ...
  ansible:
    ...
  rhaap:
    baseUrl: '<https://MyControllerUrl>'
    token: '<AAP Personal Access Token>'
    checkSSL: true
```



### 注記

Red Hat Developer Hub インストールは、ユーザー自身で外部および不正アクセスから保護する必要があります。バックエンド認証キーを他のシークレットと同様に管理します。強力なパスワード要件を満たし、パスワードを設定ファイルで公開せず、環境変数としてのみ設定ファイルに挿入します。

## 2.8. ANSIBLE プラグインサブスクリプションの警告メッセージ

Ansible プラグインは、以下のシナリオのユーザーインターフェイスで、サブスクリプション警告バーを表示します。

- [Ansible Automation Platform に接続できない](#)
- [Ansible Automation Platform に対して認証できない](#)

- [無効な Ansible Automation Platform 設定](#)
- [Ansible Automation Platform サブスクリプションがコンプライアンス違反である](#)
- [無効な Ansible Automation Platform サブスクリプション](#)

### 2.8.1. Ansible Automation Platform に接続できない

以下の警告は、Automation Controller の詳細が設定されていないか、コントローラーインスタンス API がサブスクリプションステータスをクエリーできないことを示しています。

Unable to connect to Ansible Automation Platform  
Verify that Ansible Automation Platform is reachable and correctly configured in the Ansible plug-ins.  
To get help, please refer to the Ansible plug-ins installation guide.

#### 修復手順

1. Ansible Automation Platform に到達可能で、ConfigMap の **rhaap** セクションで正しく設定されていることを確認します。
2. ご使用の環境に対して **checkSSL** キーが正しく設定されていることを確認します。
3. 設定の詳細を修正したら、Red Hat Developer Hub Pod を再起動してサブスクリプションクエリーを開始します。

### 2.8.2. Ansible Automation Platform に対して認証できない

以下の警告は、Ansible プラグインがサブスクリプションステータスをクエリーするために Ansible Automation Platform で認証できなかったことを示しています。

Unable to authenticate to Ansible Automation Platform  
Verify that the authentication details for Ansible Automation Platform are correctly configured in the Ansible plug-ins.  
For help, please refer to the Ansible plug-ins installation guide.

#### 修復手順

1. Ansible プラグインで設定された Automation Controller の Personal Access Token (PAT) が正しいことを確認します。詳細は、[Automation Controller ユーザーガイドの アプリケーション](#) セクションを参照してください。
2. 認証情報の修正後に、Red Hat Developer Hub Pod を再起動してサブスクリプションクエリーを開始します。

### 2.8.3. 無効な Ansible Automation Platform 設定

以下の警告は、Ansible Automation Platform 設定セクションが無効または不完全であることを示しています。

Invalid resource for Ansible Automation Platform  
Verify that the resource url for Ansible Automation Platform are correctly configured in the Ansible plug-ins.  
For help, please refer to the Ansible plug-ins installation guide.

#### 修復手順

1. Ansible プラグイン ConfigMap の **rhaap** セクションが正しく設定されており、必要なすべてのエントリーが含まれていることを確認します。詳細は、[Ansible Automation Platform の詳細の設定](#) を参照してください。
2. 設定の修正後に、Red Hat Developer Hub Pod を再起動してサブスクリプションクエリーを開始します。

#### 2.8.4. Ansible Automation Platform サブスクリプションがコンプライアンス違反である

以下の警告は、Ansible プラグインが Ansible Automation Platform サブスクリプションステータスを正常に取得したことを示しています。ただし、このサブスクリプションはコンプライアンス違反になります。

```
Subscription non-compliant
The connected Ansible Automation Platform subscription is out of compliance.
Contact your Red Hat account team to obtain a new subscription entitlement.
Learn more about account compliance.
```

##### 修復手順

1. 新しいサブスクリプションエンタイトルメントを取得するには、Red Hat Account チームにお問い合わせください。
2. [アカウントコンプライアンス](#) の詳細を確認してください。
3. サブスクリプションがコンプライアンスに準拠している場合は、Red Hat Developer Hub Pod を再起動して、新しいサブスクリプションクエリーを開始します。

#### 2.8.5. 無効な Ansible Automation Platform サブスクリプション

以下の警告は、Ansible プラグインが Ansible Automation Platform サブスクリプションステータスを正常に取得したことを示しています。ただし、Ansible Automation Platform ではサブスクリプションタイプが無効です。

```
Invalid subscription
The connected Ansible Automation Platform subscription is invalid.
Contact your Red Hat account team, or start an Ansible Automation Platform trial.
```

##### 修復手順

1. 新しいサブスクリプションエンタイトルメントを取得するか、[Ansible Automation Platform のトライアルを開始する](#) には、Red Hat Account チームにお問い合わせください。
2. サブスクリプションを更新したら、Red Hat Developer Hub Pod を再起動して、新しいサブスクリプションクエリーを開始します。

#### 2.8.6. Ansible プラグインソフトウェアテンプレートの追加

Red Hat Ansible は、Ansible のベストプラクティスに基づいて新しい Playbook とコレクションプロジェクトをプロビジョニングする Red Hat Developer Hub のソフトウェアテンプレートを提供します。

##### 手順

1. カスタム Red Hat Developer Hub config map (例: **app-config-rhdh**) を編集します。

2. 以下のコードを Red Hat Developer Hub の **app-config-rhdh.yaml** ファイルに追加します。

```
data:
  app-config-rhdh.yaml: |
    catalog:
      ...
    locations:
      ...
      - type: url
        target: https://github.com/ansible/ansible-rhdh-templates/blob/main/all.yaml
    rules:
      - allow: [Template]
```

詳細は、Red Hat Developer Hub の管理ガイドの [テンプレートの管理](#) セクションを参照してください。

## 第3章 OPENSIFT CONTAINER PLATFORM での ANSIBLE プラグインのアップグレード

Ansible プラグインをアップグレードするには、最新の Ansible プラグインファイルで **plugin-registry** アプリケーションを更新する必要があります。

### 3.1. ANSIBLE プラグインファイルのダウンロード

1. [Red Hat Ansible Automation Platform 製品ソフトウェアのダウンロードページ](#) から、プラグインの最新の **.tar** ファイルをダウンロードします。ファイル名の形式は **ansible-backstage-rhaap-bundle-x.y.z.tar.gz** です。Ansible プラグインリリースバージョン (**1.0.0** など) を、**x.y.z** に置き換えます。
2. **.tar** ファイルを保存するディレクトリーをローカルマシンに作成します。

```
$ mkdir /path/to/<ansible-backstage-plugins-local-dir-changeme>
```

3. ディレクトリーパスを表す環境変数 (**\$DYNAMIC\_PLUGIN\_ROOT\_DIR**) を設定します。

```
$ export DYNAMIC_PLUGIN_ROOT_DIR=/path/to/<ansible-backstage-plugins-local-dir-changeme>
```

4. **ansible-backstage-rhaap-bundle-<version-number>.tar.gz** の内容を **\$DYNAMIC\_PLUGIN\_ROOT\_DIR** に展開します。

```
$ tar --exclude='*code*' -xzf ansible-backstage-rhaap-bundle-x.y.z.tar.gz -C  
$DYNAMIC_PLUGIN_ROOT_DIR
```

Ansible プラグインリリースバージョン (**1.0.0** など) を、**x.y.z** に置き換えます。

#### 検証

**ls** を実行して、展開されたファイルが **\$DYNAMIC\_PLUGIN\_ROOT\_DIR** ディレクトリーにあることを確認します。

```
$ ls $DYNAMIC_PLUGIN_ROOT_DIR  
ansible-plugin-backstage-rhaap-x.y.z.tgz  
ansible-plugin-backstage-rhaap-x.y.z.tgz.integrity  
ansible-plugin-backstage-rhaap-backend-x.y.z.tgz  
ansible-plugin-backstage-rhaap-backend-x.y.z.tgz.integrity  
ansible-plugin-scaffolder-backend-module-backstage-rhaap-x.y.z.tgz  
ansible-plugin-scaffolder-backend-module-backstage-rhaap-x.y.z.tgz.integrity
```

**.integrity** ファイルタイプのファイルには、プラグインの SHA 値が含まれています。SHA 値は、プラグインの設定時に使用されます。

### 3.2. プラグインレジストリーの更新

最新の Ansible プラグインファイルを使用して、OpenShift クラスターでプラグインレジストリーアプリケーションを再ビルドします。

#### 手順

1. 認証情報を使用して OpenShift Container Platform インスタンスにログインし、新しいアプリケーションを作成します。
2. Red Hat Developer Hub OpenShift プロジェクトを開きます。

```
$ oc project <YOUR_DEVELOPER_HUB_PROJECT>
```

3. 次のコマンドを実行して、OpenShift クラスターのプラグインレジストリービルドを更新します。

```
$ oc start-build plugin-registry --from-dir=$DYNAMIC_PLUGIN_ROOT_DIR --wait
```

4. レジストリーが起動すると、出力に次のメッセージが表示されます。

```
Uploading directory "/path/to/dynamic_plugin_root" as binary input for the build ...
Uploading finished
build.build.openshift.io/plugin-registry-1 started
```

## 検証

plugin-registry が更新されたことを確認します。

1. OpenShift UI で、**トポロジー** をクリックします。
2. **redhat-developer-hub** アイコンをクリックして、プラグインレジストリーの Pod を表示します。
3. プラグインレジストリー Pod の **View logs** をクリックします。
4. ターミナル タブを開き、**ls** を実行してプラグインレジストリー内の **.tar** ファイルを表示します。
5. 新しい **.tar** ファイルがアップロードされていることを確認します。

## 3.3. ANSIBLE プラグインのバージョン番号の更新

### 手順

1. OpenShift Container Platform インスタンスにログインします。
2. OpenShift Developer UI で、**Helm** → **developer-hub** → **Actions** → **Upgrade** → **Yaml view** に移動します。
3. Ansible プラグインのバージョン番号と関連する **.integrity** ファイル値を更新します。

```
...
global:
...
  plugins:
    - disabled: false
      integrity: <SHA512 value>
      package: 'http://plugin-registry:8080/ansible-plugin-backstage-rhaap-x.y.z.tgz'
      pluginConfig:
        dynamicPlugins:
```

```
frontend:
  ansible.plugin-backstage-rhaap:
    applcons:
      - importName: AnsibleLogo
        name: AnsibleLogo
    dynamicRoutes:
      - importName: AnsiblePage
        menuItem:
          icon: AnsibleLogo
          text: Ansible
          path: /ansible
    - disabled: false
    integrity: <SHA512 value>
    package: >-
      http://plugin-registry:8080/ansible-plugin-scaffolder-backend-module-backstage-rhaap-
x.y.z.tgz
    pluginConfig:
      dynamicPlugins:
        backend:
          ansible.plugin-scaffolder-backend-module-backstage-rhaap: null
    - disabled: false
    integrity: <SHA512 value>
    package: >-
      http://plugin-registry:8080/ansible-plugin-backstage-rhaap-backend-x.y.z.tgz
    pluginConfig:
      dynamicPlugins:
        backend:
          ansible.plugin-backstage-rhaap-backend: null
```

4. **Upgrade** をクリックします。  
開発者ハブ Pod が再起動し、プラグインがインストールされます。

## 検証

1. OpenShift UI で、**トポロジー** をクリックします。
2. Red Hat Developer Hub インスタンスが利用可能であることを確認します。

## 第4章 ANSIBLE プラグインの OPENSIFT CONTAINER PLATFORM からのアンインストール

Ansible プラグインをアンインストールするには、Red Hat Developer Hub から **ansible:content:create** アクションを使用するソフトウェアテンプレートをすべて削除し、OpenShift の Helm チャートからプラグイン設定を削除する必要があります。

### 4.1. アンインストール

#### 手順

1. Red Hat Developer Hub で、**ansible:content:create** アクションを使用するソフトウェアテンプレートを削除します。
2. OpenShift Developer UI で、**Helm** → **developer-hub** → **Actions** → **Upgrade** → **Yaml view** に移動します。
3. **plugins** セクションで Ansible プラグイン設定を削除します。

```

...
global:
...
  plugins:
    - disabled: false
      integrity: <SHA512 value>
      package: 'http://plugin-registry:8080/ansible-plugin-backstage-rhaap-x.y.z.tgz'
      pluginConfig:
        dynamicPlugins:
          frontend:
            ansible.plugin-backstage-rhaap:
              applcons:
                - importName: AnsibleLogo
                  name: AnsibleLogo
              dynamicRoutes:
                - importName: AnsiblePage
                  menuItem:
                    icon: AnsibleLogo
                    text: Ansible
                    path: /ansible
            - disabled: false
              integrity: <SHA512 value>
              package: >-
                http://plugin-registry:8080/ansible-plugin-scaffolder-backend-module-backstage-rhaap-
x.y.z.tgz
              pluginConfig:
                dynamicPlugins:
                  backend:
                    ansible.plugin-scaffolder-backend-module-backstage-rhaap: null
            - disabled: false
              integrity: <SHA512 value>
              package: >-
                http://plugin-registry:8080/ansible-plugin-backstage-rhaap-backend-x.y.z.tgz
              pluginConfig:

```

```
dynamicPlugins:
  backend:
    ansible.plugin-backstage-rhaap-backend: null
```

4. **extraContainers** セクションを削除します。

```
upstream:
  backstage: |
    ...
  extraContainers:
    - command:
      - adt
      - server
    image: >-
      registry.redhat.io/ansible-automation-platform-25/ansible-dev-tools-rhel8:latest
    imagePullPolicy: IfNotPresent
    name: ansible-devtools-server
    ports:
      - containerPort: 8000
  image:
    pullPolicy: Always
    pullSecrets:
      - ...
      - rhdh-secret-registry
    ...
```

5. **Upgrade** をクリックします。
6. カスタム Red Hat Developer Hub config map (例: **app-config-rhdh**) を編集します。
7. **ansible** セクションを削除します。

```
data:
  app-config-rhdh.yaml: |
    ...
  ansible:
    analytics:
      enabled: true
    devSpaces:
      baseUrl: '<https://MyOwnDevSpacesUrl/>'
    creatorService:
      baseUrl: '127.0.0.1'
      port: '8000'
    rhaap:
      baseUrl: '<https://MyAapSubscriptionUrl>'
      token: '<TopSecretAAPToken>'
      checkSSL: true
    automationHub:
      baseUrl: '<https://MyOwnPAHUrl/>'
```

8. Red Hat Developer Hub のデプロイメントを再起動します。
9. **plugin-registry** OpenShift アプリケーションを削除します。

```
oc delete all -l app=plugin-registry
```

## 第5章 ANSIBLE PLUG-INS FOR RED HAT DEVELOPER HUB の設定

### 5.1. RED HAT DEVELOPER HUB 認証の有効化

Red Hat Developer Hub (RHDH) は、複数のソースコントロール管理 (SCM) システムのインテグレーションを提供します。これは、プラグインがリポジトリを作成するために必要です。

Red Hat Developer Hub の管理ガイドの [Red Hat Developer Hub での認証の有効化](#) の章を参照してください。

### 5.2. ロールベースアクセス制御の設定

Red Hat Developer Hub は、ロールベースアクセス制御 (RBAC) 機能を提供します。その後、RBAC を Ansible プラグインコンテンツに適用できます。

次の例では、Red Hat Developer Hub に RBAC を追加します。

```
data:
  app-config-rhdh.yaml: |
    plugins:
    ...
    permission:
      enabled: true
      rbac:
        admin:
          users:
            - name: user:default/<user-scm-ida>
          superUsers:
            - name: user:default/<user-admin-idb>
```

権限ポリシーと RBAC の管理に関する詳細は、Red Hat Developer Hub の管理ガイドの [Red Hat Developer Hub での認証の有効化](#) の章を参照してください。

### 5.3. ANSIBLE プラグインのオプションのインテグレーションの設定

Ansible プラグインは、Ansible Automation Platform およびその他のオプションの Red Hat 製品とのインテグレーションを提供します。

Red Hat Developer Hub の管理ガイドの [Helm チャートを使用して OpenShift Container Platform にカスタムアプリケーション設定ファイルを追加する](#) で説明されているように、**app-config-rhdh** というカスタム ConfigMap を作成します。

ConfigMap を編集するには、OpenShift UI にログインし、**Select Project ( developerHubProj ) → ConfigMaps → {developer-hub}-app-config → EditConfigMaps → app-config-rhdh** に移動します。

#### 5.3.1. OpenShift Dev Space の設定

OpenShift Dev Spaces が Ansible プラグイン用に設定されている場合、ユーザーは Red Hat Developer Hub のカタログ項目ビューからリンクをクリックし、Dev Spaces を使用してプロビジョニングされた Ansible Git プロジェクトを編集できます。



## 注記

OpenShift Dev Spaces は別の製品であり、オプションです。プラグインはこれがない場合でも機能します。

これは別の Red Hat 製品であり、Ansible Automation Platform または Red Hat Developer Hub サブスクリプションには含まれていません。

Ansible プラグインで OpenShift Dev Spaces リンクが設定されていない場合は、Ansible プラグインランディングページの **4. DEVELOP** セクションにある **OpenShift Dev Spaces ダッシュボードに移動** リンクをクリックすると、ユーザーは [Ansible 開発ツールのホームページ](#) にリダイレクトされます。

## 前提条件

- Dev Spaces のインストール。Red Hat OpenShift Dev Spaces 管理ガイドの [Dev Spaces のインストール](#) セクションを参照してください。

## 手順

1. カスタム Red Hat Developer Hub config map (例: **app-config-rhdh**) を編集します。
2. 以下のコードを Red Hat Developer Hub の **app-config-rhdh.yaml** ファイルに追加します。

```
data:
  app-config-rhdh.yaml: |-
    ansible:
      devSpaces:
        baseUrl: >-
          https://<Your OpenShift Dev Spaces URL>
```

3. **<Your OpenShift Dev Spaces URL>** を OpenShift Dev Spaces URL に置き換えます。
4. OpenShift Developer UI で、**Red Hat Developer Hub** Pod を選択します。
5. **Actions** を開きます。
6. **Restart rollout** をクリックします。

### 5.3.2. Private Automation Hub URL の設定

Private Automation Hub は、認定された Ansible コレクション、実行環境、および組織によって提供される追加の検証済みコンテンツ用の集中型オンプレミスリポジトリを提供します。

Ansible プラグインで Private Automation Hub URL が設定されていない場合、ユーザーは [Red Hat Hybrid Cloud Console Automation Hub](#) にリダイレクトされます。



## 注記

Private Automation Hub 設定はオプションですが、推奨されます。Ansible プラグインはこれがない場合でも機能します。

## 前提条件:

- Private Automation Hub インスタンス。

Private Automation Hub のインストールに関する詳細は、Ansible Automation Platform ドキュメントの [インストールとアップグレード](#) ガイドを参照してください。

#### 手順:

1. カスタム Red Hat Developer Hub config map (例: **app-config-rhdh**) を編集します。
2. 以下のコードを Red Hat Developer Hub の **app-config-rhdh.yaml** ファイルに追加します。

```
data:
  app-config-rhdh.yaml: |-
    ansible:
    ...
    automationHub:
      baseUrl: '<https://MyOwnPAHUrl>'
    ...
```

3. **<https://MyOwnPAHUrl/>** を Private Automation Hub URL に置き換えます。
4. OpenShift Developer UI で、**Red Hat Developer Hub** Pod を選択します。
5. **Actions** を開きます。
6. **Restart rollout** をクリックします。

## 5.4. RED HAT DEVELOPER HUB データ TELEMETRY のキャプチャー

Red Hat Developer Hub (RHDH) は、デフォルトで有効になっている **backstage-plugin-analytics-provider-segment** プラグインを使用して、telemetry データを Red Hat に送信します。これには、Ansible プラグインからの telemetry データが含まれます。

Red Hat は、以下のデータを収集および分析して、Red Hat Developer Hub のエクスペリエンスを改善します。

- ページ訪問やリンクまたはボタンのクリックのイベント。
- システム関連の情報 (ロケール、タイムゾーン、ブラウザや OS の詳細を含むユーザーエージェントなど)。
- ページ関連の情報 (タイトル、カテゴリー、拡張機能名、URL、パス、リファラー、検索パラメーターなど)。
- 匿名の IP アドレス (0.0.0.0 として記録)。
- 匿名化されたユーザー名ハッシュ。これは、RHDH アプリケーションの一意のユーザー数を識別するためにのみ使用される一意の識別子です。
- Ansible プラグインのフィードバックフォームで提供されるフィードバックと感想。

Red Hat Developer Hub では、Telemetry データ収集機能を無効化またはカスタマイズできます。詳細は、[Red Hat Developer Hub の管理ガイド](#)の [Telemetry データコレクション](#) セクションを参照してください。

