Red Hat Advanced Cluster Management for Kubernetes 2.2

クラスターの管理

詳細は、クラウドプロバイダー全体でクラスターを作成、インポート、および管理する方法を説明します。

Last Updated: 2023-02-22

Red Hat Advanced Cluster Management for Kubernetes 2.2 クラスターの管理

詳細は、クラウドプロバイダー全体でクラスターを作成、インポート、および管理する方法を説明します。

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

詳細は、クラウドプロバイダー全体でクラスターを作成、インポート、および管理する方法を説明します。

目次

第1章 クラスターの管理

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用した、クラウドプロバイダー全体におけるクラスターの作成、インポート、管理の方法を説明します。

以下のトピックでは、クラウドプロバイダー全体でクラスターを管理する方法について説明します。

- サポート対象のクラウド
- クラスターのサイズ調整
- プロバイダー接続の作成
- クラスターの作成
- ハブクラスターへのターゲットのマネージドクラスターのインポート
- ManagedClusterSets
- マネージドクラスターの AnsibleJob の作成
- Submariner
- クラスターのアップグレード
- マネージメントからのクラスターの削除

第2章 サポート対象のクラウド

Red Hat Advanced Cluster Management for Kubernetes で利用可能なクラウドプロバイダーについて説明します。また、利用可能なマネージドプロバイダーに関するドキュメントも参照してください。

- サポート対象のハブクラスタープロバイダー
- サポート対象のマネージドクラスタープロバイダー
- kubectl の設定

ベストプラクティス: マネージドクラスターのプロバイダーには、最新版の Kubernetes を使用してください。

2.1. サポート対象のハブクラスタープロバイダー

ハブクラスターとしてサポートされるのは、Red Hat OpenShift Container Platform 4.5.2 以降、4.6.1 以降、4.7.0 以降です。

- OpenShift on Amazon Web Services を参照してください。
- Azure on Red Hat OpenShift を参照してください。
- Red Hat OpenShift Dedicated を参照してください。
- Red Hat OpenShift Container Platform on OpenStack (OpenStack バージョン 16.1 以降) を参照してください。
- Red Hat OpenShift Container Platform on VMware vSphere を参照してください。
- Red Hat OpenShift Container Platform on IBM Cloud (ROKS) (Red Hat OpenShift Container Platform バージョン 4.5 以降) を参照してください。

2.2. サポート対象のマネージドクラスタープロバイダー

マネージドクラスターとしてサポートされるのは、Red Hat OpenShift Container Platform 3.11.200 以降、4.4.3 以降、4.5.2 以降、4.6.1 以降、4.7.0 以降です。

利用可能なマネージドクラスターのオプションおよびドキュメントは以下を参照してください。

- OpenShift on Amazon Web Services を参照してください。
- Red Hat OpenShift Container Platform on IBM Cloud (ROKS) (Kubernetes 1.17 以降) を参照してください。
- About Red Hat OpenShift Kubernetes Engine を参照してください。
- Red Hat OpenShift Container Platform (4.6.1 以降) on IBM Z を参照してください。
- IBM Cloud Kubernetes Service 概説 (Kubernetes 1.18 以降) を参照してください。
- Google Kubernetes Engine (Kubernetes 1.17 以降) を参照してください。
- Azure Kubernetes Service (Kubernetes 1.19.6 以降) を参照してください。
- Amazon Elastic Kubernetes Service (Kubernetes 1.17.6 以降) を参照してください。

- Red Hat OpenShift Container Platform on VMware vSphere を参照してください。
- Azure on Red Hat OpenShift を参照してください。
- Red Hat OpenShift Dedicated (Red Hat OpenShift Container Platform バージョン 4.5.16 以降) を参照してください。
- Red Hat OpenShift Container Platform on OpenStack (OpenStack バージョン 16.1 以降) を参照してください。

2.3. KUBECTL の設定

前述したベンダーのドキュメントを参照し、**kubectl** の設定方法を確認してください。マネージドクラスターをハブクラスターにインポートする場合には **kubectl** をインストールしておく必要があります。詳細は、「ハブクラスターへのターゲットのマネージドクラスターのインポート」を参照してください。

第3章 クラスターのサイズ調整

仮想マシンのサイズやノード数などのスケーリングの手順を使用して、Red Hat OpenShift Container Platform マネージドクラスターの仕様をカスタマイズできます。クラスターのサイズ変更についての詳細は、「クラスターの OpenShift Container Platform バージョンについての推奨されるクラスタースケーリングプラクティス」を参照してください。

ヒント: Red Hat Advanced Cluster Management for Kubernetes コンソールを使用してクラスターを作成した場合には、これは OpenShift Container Platform クラスターになります。

第4章 リリースイメージ

Red Hat Advanced Cluster Management for Kubernetes を使用してプロバイダーでクラスターを作成する場合は、新規クラスターに使用するリリースイメージを指定する必要があります。リリースイメージでは、クラスターのビルドに使用する Red Hat OpenShift Container Platform のバージョンを指定します。

acm-hive-openshift-releases GitHub リポジトリーの **yaml** ファイルを使用して、リリースイメージを参照します。Red Hat Advanced Cluster Management はこれらのファイルを使用して、コンソールで利用可能なリリースイメージの一覧を作成します。これには、OpenShift Container Platform における最新の fast チャネルイメージが含まれます。コンソールには、OpenShift Container Platform の 3 つの最新バージョンの最新リリースイメージのみが表示されます。たとえば、コンソールオプションに以下のリリースイメージが表示される可能性があります。

- quay.io/openshift-release-dev/ocp-release:4.5.36-x86_64
- quay.io/openshift-release-dev/ocp-release:4.6.23-x86_64
- quay.io/openshift-release-dev/ocp-release:4.7.4-x86_64

追加のリリースイメージは保管されますが、コンソールには表示されません。利用可能なすべてのリリースイメージを表示するには、CLI で **kubectl get clusterimageset** を実行します。最新のリリースイメージでクラスターを作成することが推奨されるため、コンソールには最新バージョンのみがあります。特定バージョンのクラスターを作成する必要がある場合があります。そのため、古いバージョンが利用可能となっています。

リポジトリーには、**clusterImageSets** ディレクトリーと **subscription** ディレクトリーが含まれます。これらのディレクトリーは、リリースイメージの操作時に使用します。

clusterImageSets ディレクトリーには以下のディレクトリーが含まれます。

- Fast: サポート対象の各 OpenShift Container Platform バージョンのリリースイメージの内、最新バージョンを参照するファイルが含まれます。このフォルダー内のリリースイメージはテストされ、検証されており、サポートされます。
- Releases - 各 OpenShift Container Platform バージョン (stable、fast、および candidate チャネル) のリリースイメージをすべて参照するファイルが含まれます。**注記:** このリリースはすべてテストされ、安定していると判別されているわけではありません。
- Stable: サポート対象の各 OpenShift Container Platform バージョンのリリースイメージの内、最新の安定版 2 つを参照するファイルが含まれます。このフォルダー内のリリースイメージはテストされ、検証されており、サポートされます。

独自の **ClusterImageSets** は以下の 3 つの方法でキュレートできます。

この 3 つの方法で最初のステップは、最新の fast チャネルイメージの自動更新を実行するのに含まれるサブスクリプションを無効にすることです。最新の fast の **ClusterImageSets** の自動キュレーションを無効にするには、multiclusterhub リソースでインストーラーパラメーターを使用します。**spec.disableUpdateClusterImageSets** パラメーターを **true** と **false** の間で切り替えることにより、Red Hat Advanced Cluster Management でインストールしたサブスクリプションが、それぞれ無効または有効になります。独自のイメージをキューレートする場合は、**spec.disableUpdateClusterImageSets** を **true** に設定してサブスクリプションを無効にします。

オプション 1: クラスターの作成時にコンソールで使用する特定の **ClusterImageSet** のイメージ参照を指定します。指定する新規エントリーはそれぞれ保持され、将来のすべてのクラスタープロビジョニングで利用できます。たとえば、エントリーは **quay.io/openshift-release-dev/ocp-release:4.6.8-x86_64** のようになります。

オプション 2: GitHub リポジトリー **acm-hive-openshift-releases** から YAML ファイル **ClusterImageSets** を手動で作成し、適用します。

オプション 3: GitHub リポジトリー **acm-hive-openshift-releases** の **README.md** に従って、フォークした GitHub リポジトリーから **ClusterImageSets** の自動更新を有効にします。

subscription ディレクトリーには、リリースイメージの一覧がプルされる場所を指定するファイルが含まれます。Red Hat Advanced Cluster Management のデフォルトのリリースイメージは、Quay.io デフォルトで提供されます。イメージは、GitHub リポジトリー acm-hive-openshift-releases のファイルで参照されます。

4.1. 利用可能なリリースイメージの同期

リリースイメージは頻繁に更新されるため、リリースイメージの一覧を同期して、利用可能な最新バージョンを選択できるようにする必要があります。リリースイメージは、GitHub リポジトリー acm-hive-openshift-releases にあります。

リリースイメージの安定性には、以下の 3 つのレベルがあります。

表4.1 リリースイメージの安定性レベル

カテゴリー	説明
stable	完全にテストされたイメージで、クラスターを正常にインストールしてビルドできることが確認されています。
fast	部分的にテスト済みですが、stable バージョンよりも安定性が低い可能性があります。
candidate	テストはしていませんが、最新のイメージです。バグがある可能性もあります。

一覧を更新するには、以下の手順を実行します。

1. インストーラーが管理する **acm-hive-openshift-releases** サブスクリプションが有効になっている場合は、**disableUpdateClusterImageSets** の値を **true** に設定してサブスクリプションを無効にします。以下のコマンドのようなコマンドを入力して、サブスクリプションを削除できます。

   ```
   oc delete -f subscription/subscription-stable
   ```

2. acm-hive-openshift-releases GitHub repository のクローンを作成します。
3. 以下のコマンドを入力して、stable リリースイメージに接続し、Red Hat Advanced Cluster Management for Kubernetes のハブクラスターに同期します。

   ```
   make subscribe-stable
   ```

   注記: この **make** コマンドは、Linux または MacOS のオペレーティングシステムを使用している場合のみ実行できます。

   約 1 分後に、**安定** 版のリリースイメージの最新の一覧が利用可能になります。

- Fast リリースイメージを同期して表示するには、以下のコマンドを実行します。

```
make subscribe-fast
```

注記: この **make** コマンドは、Linux または MacOS のオペレーティングシステムを使用している場合のみ実行できます。

このコマンド実行の約 1 分後に、利用可能な **stable** と **fast** のリリースイメージの一覧が、現在利用可能なイメージに更新されます。

- **candidate** リリースイメージを同期して表示するには、以下のコマンドを実行します。

```
make subscribe-candidate
```

注記: この **make** コマンドは、Linux または MacOS のオペレーティングシステムを使用している場合のみ実行できます。

このコマンド実行の約 1 分後に、利用可能な **stable**、**fast**、および **candidate** のリリースイメージの一覧が、現在利用可能なイメージに更新されます。

4. クラスターの作成時に、Red Hat Advanced Cluster Management コンソールで現在利用可能なリリースイメージの一覧を表示します。

5. 以下の形式でコマンドを入力して、これらのチャネルのサブスクライブを解除して更新の表示を停止することができます。

```
oc delete -f subscription/subscription-stable
```

4.1.1. 接続時におけるリリースイメージのカスタム一覧の管理

すべてのクラスターに同じリリースイメージが使用されるようにします。クラスターの作成時に利用可能なリリースイメージのカスタム一覧を作成し、作業を簡素化します。利用可能なリリースイメージを管理するには、以下の手順を実行します。

1. インストーラーが管理する **acm-hive-openshift-releases** サブスクリプションが有効になっている場合は、**disableUpdateClusterImageSets** の値を **true** に設定して無効にします。

2. acm-hive-openshift-releases GitHub repository をフォークします。

3. **./subscription/channel.yaml** ファイルを更新して、sto **lostron** ではなくフォークしたリポジトリーの GitHub 名にアクセスするように **spec: pathname** を変更します。この手順では、ハブクラスターによるリリースイメージの取得先を指定します。更新後の内容は以下の例のようになります。

```
spec:
  type: GitHub
  pathname: https://github.com/<forked_content>/acm-hive-openshift-releases.git
```

forked_content はフォークしたリポジトリーのへのパスに置き換えます。

4. Red Hat Advanced Cluster Management for Kubernetes を使用してクラスターを作成する時に利用できるようにイメージの YAML ファイルを **./clusterImageSets/stable/ または ./clusterImageSets/fast/* ディレクトリーに追加します。*ヒント: フォークしたリポジトリーに変更をマージすることで、利用可能な YAML ファイルはメインのリポジトリーから取得できます。**

5. フォークしたリポジトリーに変更をコミットし、マージします。

6. **acm-hive-openshift-releases** リポジトリーをクローンした後に stable リリースイメージの一覧を同期するには、以下のコマンドを入力して stable イメージを更新します。

```
make subscribe-stable
```

   注記: この make コマンドは、Linux または MacOS のオペレーティングシステムを使用している場合のみ実行できます。

   このコマンドを実行後に、利用可能な安定版のリリースイメージの一覧が、現在利用可能なイメージに約 1 分ほどで更新されます。

7. **デフォルトでは、安定版のイメージのみが一覧表示されます。Fast リリースイメージを同期して表示するには、以下のコマンドを実行します。**

```
make subscribe-fast
```

   注記: この make コマンドは、Linux または MacOS のオペレーティングシステムを使用している場合のみ実行できます。

   このコマンドを実行後に、利用可能な fast リリースイメージの一覧が、現在利用可能なイメージに約 1 分ほどで更新されます。

8. **デフォルトではRed Hat Advanced Cluster Management は ClusterImageSets を複数事前に読み込みます。以下のコマンドを使用して、利用可能ものを表示し、デフォルトの設定を削除します。**

```
oc get clusterImageSets
oc delete clusterImageSet <clusterImageSet_NAME>
```

   注記: disableUpdateClusterImageSets の値を true に設定して、インストーラー管理の ClusterImageSets の自動更新をまだ無効にしていない場合は、削除するイメージが自動的に再作成されます。

9. **クラスターの作成時に、Red Hat Advanced Cluster Management コンソールで現在利用可能なリリースイメージの一覧を表示します。**

4.1.2. 非接続時におけるリリースイメージのカスタム一覧の管理

ハブクラスターにインターネット接続がない場合に、リリースイメージのカスタムリストを管理する必要がある場合があります。クラスターの作成時に利用可能なリリースイメージのカスタム一覧を作成します。非接続時に、利用可能なリリースイメージを管理するには、以下の手順を実行します。

1. **オンライン接続されているシステムを使用している場合は、GitHub リポジトリー acm-hive-openshift-releases に移動します。**

2. **clusterImageSets ディレクトリーを、非接続の Red Hat Advanced Cluster Management for Kubernetes ハブクラスターにアクセス可能なシステムにコピーします。**

3. **clusterImageSet YAML を手作業で追加し、Red Hat Advanced Cluster Management for Kubernetes コンソールを使用してクラスターを作成する時に利用できるようにイメージの YAML ファイルを追加します。**

4. **clusterImageSets コマンドを作成します。**

```
oc create -f <clusterImageSet_FILE>
```

追加するリソース毎にこのコマンドを実行すると、利用可能なリリースイメージの一覧が使用できるようになります。

5. または Red Hat Advanced Cluster Management のクラスター作成のコンソールに直接イメージの URL を貼り付けることもできます。これにより、clusterImageSets が存在しない場合には、新しいものが作成されます。

6. クラスターの作成時に、Red Hat Advanced Cluster Management コンソールで現在利用可能なリリースイメージの一覧を表示します。

第5章 ベアメタルアセットの作成および変更

ベアメタルアセットとは、クラウドオペレーションを実行するように設定された仮想サーバーまたは物理サーバーのことです。Red Hat Advanced Cluster Management for Kubernetes は管理者が作成するベアメタルアセットに接続してクラスターを作成できます。

Red Hat Advanced Cluster Management for Kubernetes でベアメタルアセットを作成して、ベアメタルアセットでクラスターを作成する必要があります。以下の手順を使用して、Red Hat Advanced Cluster Management for Kubernetes が管理するクラスターをホストできるベアメタルアセットを作成します。

5.1. 前提条件

ベアメタルアセットを作成する前に、以下の前提条件を満たす必要があります。

- OpenShift Container Platform バージョン 4.5 以降に、Red Hat Advanced Cluster Management for Kubernetes ハブクラスターをデプロイしておく。
- Red Hat Advanced Cluster Management for Kubernetes ハブクラスターがベアメタルアセットに接続できるようにアクセスを設定しておく。
- ベアメタルアセットおよび、ベアメタルアセットへのログインまたは管理に必要なパーミッションを指定したログイン認証情報を設定しておく。注記: ベアメタルアセットへのログイン認証情報には、管理者が提供する以下のアセットの項目が含まれます。
  - ユーザー名
  - パスワード
  - ベースボード管理コントローラー (BMC) アドレス
  - ブート NIC MAC アドレス

5.2. コンソールを使用したベアメタルアセットの作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用してベアメタルアセットを作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Bare metal assetsに移動します。
2. Bare metal assetsページで Create bare metal assetをクリックします。
3. クラスターの作成時に識別できるようにアセット名を入力します。
4. ベアメタルアセットを作成する namespace を入力します。
   注記: ベアメタルアセット、ベアメタルのマネージドクラスター、および関連シークレットは同じ namespace に配置する必要があります。

   この namespace にアクセスできるユーザーは、クラスターの作成時にこのアセットをクラスターに関連付けることができます。
5. ベースボード管理コントローラー (BMC) アドレスを入力します。このコントローラーで、ホストとの通信が可能になります。以下のプロトコルがサポートされます。
   - IPMI。詳細は、IPMI 2.0 Specificationを参照してください。

- iDRAC。詳細は、Support for Integrated Dell Remote Access Controller 9 (iDRAC9)を参照してください。
- iRMC。詳細は、Data Sheet: FUJITSU Software ServerView Suite integrated Remote Management Controller - iRMC S5 を参照してください。
- Redfish。詳細は、Redfish specification を参照してください。

6. ベアメタルアセットのユーザー名とパスワードを入力します。
7. ベアメタルアセットのブート NIC MAC アドレスを追加します。これは、ネットワーク接続されたホストの NIC の MAC アドレスで、ベアメタルアセットにホストをプロビジョニングする時に使用します。

「ベアメタルでのクラスターの作成」に進んでください。

5.3. ベアメタルアセットの変更

ベアメタルアセットの設定を変更する必要がある場合は、以下の手順を実行します。

1. Red Hat Advanced Cluster Management for Kubernetes コンソールのナビゲーションで、Automate infrastructure >Bare metal assetsを選択します。
2. テーブルで変更するアセットのオプションメニューを選択します。
3. Edit assetを選択します。

5.4. ベアメタルアセットの削除

ベアメタルアセットがどのクラスターにも使用されなくなった場合には、利用可能なベアメタルアセット一覧から削除できます。使用されていないアセットを削除することで、利用可能なアセット一覧が簡素化されて、対象のアセットが誤って選択されないようにします。

ベアメタルアセットを削除するには、以下の手順を実行します。

1. Red Hat Advanced Cluster Management for Kubernetes コンソールのナビゲーションで、Automate infrastructure >Bare metal assetsを選択します。
2. テーブルで削除するアセットのオプションメニューを選択します。
3. Delete assetを選択します。

第6章 プロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes でクラウドサービスプロバイダーに Red Hat OpenShift Container Platform クラスターを作成するには、プロバイダー接続 が必要です。

プロバイダー接続には、アクセス用の認証情報と、プロバイダーの設定情報が保存されます。プロバイダーアカウントごとに独自のプロバイダー接続が、プロバイダーごとにドメインが必要です。

以下のファイルには、サポートされる各プロバイダーの接続のドキュメントを作成するのに必要な情報がまとめられています。

- Amazon Web Services のプロバイダー接続の作成
- Microsoft Azure のプロバイダー接続の作成
- Google Cloud Platform のプロバイダー接続の作成
- VMware vSphere のプロバイダー接続の作成
- ベアメタルのプロバイダー接続の作成

6.1. AMAZON WEB SERVICES のプロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、Amazon Web Services (AWS) に OpenShift クラスターをデプロイして管理するには、プロバイダー接続が必要です。

注記: Red Hat Advanced Cluster Management for Kubernetes でクラスターを作成する前に、以下の手順を実行する必要があります。

6.1.1. 前提条件

プロバイダー接続を作成する前に、以下の前提条件を満たす必要があります。

- Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをデプロイしておく。
- Amazon Web Services で Kubernetes クラスターを作成できるようにする Red Hat Advanced Cluster Management for Kubernetes ハブクラスターでのインターネットアクセスがある。
- アクセスキー ID およびシークレットアクセスキーなど、Amazon Web Services (AWS) のログイン認証情報。Understanding and getting your AWS credentialsを参照してください。
- AWS でクラスターをインストールできるようにするアカウントのパーミッション。設定の方法は、「AWS アカウントの設定」を参照してください。

6.1.2. コンソールを使用したプロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes コンソールからプロバイダー接続を作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Clusters ページで Provider connections タブを選択します。
   既存のプロバイダー接続が表示されます。

3. Add a connectionを選択します。
4. Amazon Web Servicesをプロバイダーとして選択します。
5. プロバイダー接続の名前を追加します。
6. 一覧からプロバイダー接続の namespace を選択します。
ヒント: 便宜上およびセキュリティー上、プロバイダー接続のホスト専用の namespace を作成します。
7. オプションで、プロバイダー接続の ベース DNS ドメイン を追加できます。ベース DNS ドメインをプロバイダー接続に追加した場合には、このプロバイダー接続でクラスターを作成すると、このベース DNS ドメインは自動的に正しいフィールドに設定されます。
8. Amazon Web Service アカウントの AWS アクセスキー IDを追加します。AWS にログインして、ID を検索します。
9. AWS Secret Access Keyを追加します。
10. Red Hat OpenShift pull secretを入力します。Pull secretからプルシークレットをダウンロードします。
11. SSH 秘密鍵 と SSH 公開鍵 を追加し、クラスターに接続できるようにします。既存のキーペアを使用するか、キー生成プログラムで新しいキーを作成できます。キー生成の方法は、「SSH プライベートキーの生成およびエージェントへの追加」を参照してください。
12. Createをクリックします。プロバイダー接続を作成すると、プロバイダー接続の一覧に追加されます。

「Amazon Web Services でのクラスターの作成」の手順を実行して、このプロバイダー接続を使用するクラスターを作成します。

6.1.3. プロバイダー接続の削除

プロバイダー接続を使用するクラスターを管理しなくなった場合には、そのプロバイダー接続を削除して、プロバイダー接続の情報を保護します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Provider connections を選択します。
3. 削除するプロバイダー接続の横にあるオプションメニューを選択します。
4. Delete connectionを選択します。

6.2. MICROSOFT AZURE のプロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、Microsoft Azure で Red Hat OpenShift Container Platform クラスターを作成して管理するには、プロバイダー接続が必要です。

注記 以下の手順は、Red Hat Advanced Cluster Management for Kubernetes でクラスターを作成するための前提条件となっています。

6.2.1. 前提条件

プロバイダー接続を作成する前に、以下の前提条件を満たす必要があります。

- Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをデプロイしておく。
- Azure で Kubernetes クラスターを作成できるようにする Red Hat Advanced Cluster Management for Kubernetes ハブクラスターでのインターネットアクセスがある。
- ベースドメインのリソースグループおよび Azure Service Principal JSON などの Azure ログイン認証情報。azure.microsoft.com を参照してください。
- Azre でクラスターがインストールできるようにするアカウントのパーミッション。詳細は、「How to configure Cloud Services」および「Azure アカウントの設定」を参照してください。

6.2.2. コンソールを使用したプロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes コンソールからプロバイダー接続を作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Clusters ページで Provider connections タブを選択します。
   既存のプロバイダー接続が表示されます。
3. Add a connection を選択します。
4. Microsoft Azure をプロバイダーとして選択します。
5. プロバイダー接続の名前を追加します。
6. 一覧からプロバイダー接続の namespace を選択します。
   ヒント: 便宜上およびセキュリティー上、プロバイダー接続のホスト専用の namespace を作成できます。
7. オプションで、プロバイダー接続の ベース DNS ドメイン を追加できます。ベース DNS ドメインをプロバイダー接続に追加した場合には、このプロバイダー接続でクラスターを作成すると、このベース DNS ドメインは自動的に正しいフィールドに設定されます。
8. Azure アカウントの ベースドメインリソースグループ名 を追加します。このエントリーは、Azure アカウントで作成したリソース名です。Azure インターフェースで Home >DNS Zones を選択することで、ベースドメインのリソースグループ名を検索できます。ベースドメインのリソースグループ名は、アカウントに適用するベース DNS ドメインが含まれるエントリーの Resource Group コラムにあります。
9. Client ID を追加します。この値は、以下のコマンドを使用してサービスプリンシパルを作成すると、 **appId** プロパティーとして設定されます。

   ```
   az ad sp create-for-rbac --role Contributor --name <service_principal>
   ```

   service_principal は、お使いのサービスプリンシパル名に置き換えます。
10. Client Secret を追加します。この値は、以下のコマンドを使用してサービスプリンシパルを作成すると、 **password** プロパティーとして設定されます。

    ```
    az ad sp create-for-rbac --role Contributor --name <service_principal>
    ```

service_principal は、お使いのサービスプリンシパル名に置き換えます。

11. Subscription ID を追加します。以下のコマンドの出力では、この値は、id プロパティーになります。

```
az account show
```

12. Tenant ID を追加します。以下のコマンドの出力では、この値は、tenantId プロパティーになります。

```
az account show
```

13. Red Hat OpenShift pull secretを入力します。Pull secretからプルシークレットをダウンロードします。

14. クラスターへの接続に使用する SSH 秘密鍵 と SSH 公開鍵 を追加します。既存のキーペアを使用するか、キー生成プログラムで新しいキーを作成できます。キー生成の方法は、「SSH プライベートキーの生成およびエージェントへの追加」を参照してください。

15. Create をクリックします。プロバイダー接続を作成すると、プロバイダー接続の一覧に追加されます。

「Microsoft Azure でのクラスターの作成」の手順を実行して、このプロバイダー接続を使用するクラスターを作成します。

6.2.3. プロバイダー接続の削除

プロバイダー接続を使用するクラスターを管理しなくなった場合には、そのプロバイダー接続を削除して、プロバイダー接続の情報を保護します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Provider connections を選択します。
3. 削除するプロバイダー接続のオプションメニューを選択します。
4. Delete connection を選択します。

6.3. GOOGLE CLOUD PLATFORM のプロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、Google Cloud Platform (GCP) で Red Hat OpenShift Container Platform クラスターを作成して管理するには、プロバイダー接続が必要です。

注記 以下の手順は、Red Hat Advanced Cluster Management for Kubernetes でクラスターを作成するための前提条件となっています。

6.3.1. 前提条件

プロバイダー接続を作成する前に、以下の前提条件を満たす必要があります。

- Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをデプロイしておく。

- GCP で Kubernetes クラスターを作成できるようにする Red Hat Advanced Cluster Management for Kubernetes ハブクラスターでのインターネットアクセスがある。
- ユーザーの Google Cloud Platform プロジェクト ID および Google Cloud Platform サービスアカウント JSON キーなど、GCP ログインの認証情報。「Creating and managing projects」を参照してください。
- GCP でクラスターがインストールできるようにするアカウントのパーミッション。アカウントの設定方法は、「GCP プロジェクトの設定」を参照してください。

6.3.2. コンソールを使用したプロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes コンソールからプロバイダー接続を作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Clusters ページで Provider connections タブを選択します。
   既存のプロバイダー接続が表示されます。
3. Add a connection を選択します。
4. Google Cloud Platform をプロバイダーとして選択します。
5. プロバイダー接続の名前を追加します。
6. 一覧からプロバイダー接続の namespace を選択します。

   ヒント

   便宜上およびセキュリティー上、プロバイダー接続のホスト専用の namespace を作成します。

7. オプションで、プロバイダー接続の ベース DNS ドメイン を追加できます。ベース DNS ドメインをプロバイダー接続に追加した場合には、このプロバイダー接続でクラスターを作成すると、このベース DNS ドメインは自動的に正しいフィールドに設定されます。
8. GCP アカウントの Google Cloud Platform project IDを追加します。GCP にログインして設定を取得します。
9. Google Cloud Platform service account JSON keyを追加します。適切なパーミッションでキーを作成するには、以下の手順を実行します。
   a. GCP のメインメニューで IAM & Adminを選択して、Service Accounts appletを起動します。
   b. Create Service Accountを選択します。
   c. サービスアカウントの Name、Service account IDおよび Description を指定します。
   d. Create を選択してサービスアカウントを作成します。
   e. Owner のロールを選択し、Continue をクリックします。
   f. Create Key をクリックします。

g. JSON を選択して、Create をクリックします。

h. 作成されたファイルをコンピューターに保存します。

i. Google Cloud Platform service account JSON keyのコンテンツを指定します。

10. Red Hat OpenShift pull secretを入力します。Pull secretからプルシークレットをダウンロードします。

11. クラスターにアクセスできるように SSH 秘密鍵 と SSH 公開鍵 を追加します。既存のキーペアを使用するか、キー生成プログラムで新しいキーを作成できます。キー生成の方法は、「SSH プライベートキーの生成およびエージェントへの追加」を参照してください。

12. Create をクリックします。プロバイダー接続を作成すると、プロバイダー接続の一覧に追加されます。

「Google Cloud Platform でのクラスターの作成」の手順を実行して、 クラスターの作成時にこの接続を使用できます。

6.3.3. プロバイダー接続の削除

プロバイダー接続を使用するクラスターを管理しなくなった場合には、そのプロバイダー接続を削除して、プロバイダー接続の情報を保護します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Provider connections を選択します。
3. 削除するプロバイダー接続の横にあるオプションメニューを選択します。
4. Delete connection を選択します。

6.4. VMWARE VSPHERE のプロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、VMware vSphere で Red Hat OpenShift Container Platform クラスターを作成して管理するには、プロバイダー接続が必要です。注記: OpenShift Container Platform バージョン 4.5.x 以降のみがサポートされます。

注記: Red Hat Advanced Cluster Management でクラスターを作成する前に、以下の手順を実行する必要があります。

6.4.1. 前提条件

プロバイダー接続を作成する前に、以下の前提条件を満たす必要があります。

- OpenShift Container Platform バージョン 4.5 以降に、Red Hat Advanced Cluster Management ハブクラスターをデプロイしておく。
- VMware vSphere で Kubernetes クラスターを作成できるようにする Red Hat Advanced Cluster Management ハブクラスターでのインターネットアクセスがある。
- インストーラーでプロビジョニングされるインフラストラクチャーを使用する場合に OpenShift Container Platform 向けに設定された VMware vSphere ログイン認証情報および vCenter 要件。「クラスターの vSphere へのインストール」を参照してください。これらの認証除法には、以下の情報が含まれます。

- vCenter アカウントの権限
- クラスターリソース
- HDCP が利用できる
- 時間を同期した ESXi ホスト (例: NTP)

6.4.2. コンソールを使用したプロバイダー接続の作成

Red Hat Advanced Cluster Management コンソールからプロバイダー接続を作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Clusters ページで Provider connections タブを選択します。
   既存のプロバイダー接続が表示されます。
3. Add a connection を選択します。
4. VMware vSphere をプロバイダーとして選択します。
5. プロバイダー接続の名前を追加します。
6. 一覧からプロバイダー接続の namespace を選択します。
   ヒント: 便宜上およびセキュリティー上、プロバイダー接続のホスト専用の namespace を作成します。
7. オプションで、プロバイダー接続のベース DNS ドメインを追加できます。ベース DNS ドメインをプロバイダー接続に追加した場合には、このプロバイダー接続でクラスターを作成すると、このベース DNS ドメインは自動的に正しいフィールドに設定されます。
8. VMware vCenter サーバーの完全修飾ホスト名または IP アドレス を追加します。値は vCenter サーバーのルート CA 証明書に定義する必要があります。可能な場合は、完全修飾ホスト名を使用します。
9. VMware vCenter のユーザー名を追加します。
10. VMware vCenter パスワードを追加します。
11. VMware vCenter ルート CA 証明書を追加します。
    a. VMware vCenter サーバー (https://<vCenter_address>/certs/download.zip) から **download.zip** として証明書をダウンロードできます。**vCenter_address** は、vCenter サーバーのアドレスに置き換えます。
    b. **download.zip** のパッケージを展開します。
    c. 拡張が **.0** の **certs/<platform>** ディレクトリーから証明書を使用します。ヒント: **ls certs/<platform>** コマンドを使用して、お使いのプラットフォームで使用可能な全証明書を一覧表示できます。
       <platform> は、**lin**、**mac** または **win** など、お使いのプラットフォームに置き換えます。

       例: **certs/lin/3a343545.0**
12. VMware vSphere クラスター名を追加します。

13. VMware vSphere データセンターを追加します。
14. VMware vSphere デフォルトデータストアを追加します。
15. OpenShift Container Platform プルシークレットを入力します。Pull secretからプルシークレットをダウンロードします。
16. SSH 秘密鍵と SSH 公開鍵を追加し、クラスターに接続できるようにします。既存のキーペアを使用するか、キー生成プログラムで新しいキーを作成できます。詳細は、「SSH プライベートキーの生成およびエージェントへの追加」を参照してください。
17. Create をクリックします。プロバイダー接続を作成すると、プロバイダー接続の一覧に追加されます。

「VMware vSphere でのクラスターの作成」の手順を実行して、このプロバイダー接続を使用するクラスターを作成します。

6.4.3. プロバイダー接続の削除

プロバイダー接続を使用するクラスターを管理しなくなった場合には、そのプロバイダー接続を削除して、プロバイダー接続の情報を保護します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Provider connections を選択します。
3. 削除するプロバイダー接続のオプションメニューを選択します。
4. Delete connection を選択します。

6.5. ベアメタルのプロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、ベアメタル環境で Red Hat OpenShift Container Platform クラスターを作成して管理するには、プロバイダー接続が必要です。

6.5.1. 前提条件

プロバイダー接続を作成する前に、以下の前提条件を満たす必要があります。

- Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをデプロイしておく。ベアメタルクラスターを管理する場合は、Red Hat OpenShift Container Platform バージョン 4.5 以降に、ハブクラスターを インストールする必要があります。
- ベアメタルサーバーで Kubernetes クラスターを作成できるようにする Red Hat Advanced Cluster Management for Kubernetes ハブクラスターでのインターネットアクセスがある。
- libvirt URI、SSH、秘密鍵、SSH の既知のホスト一覧など、ベアメタルサーバーのログイン認証情報。「SSH プライベートキーの生成およびエージェントへの追加」を参照してください。
- 切断された環境では、クラスター作成用のリリースイメージをコピーできるミラーレジストリーを設定しておく。詳細は、OpenShift Container Platformドキュメントの「Mirroring images for a disconnected installation」を参照してください。
- ベアメタルインフラストラクチャーでクラスターがインストールできるようにするアカウントのパーミッション。

6.5.2. プロビジョニングホストの準備

ベアメタルの認証情報とクラスターを作成する場合には、プロビジョニングホストが必要となります。プロビジョニングホストは、インストールに使用できるブートストラップホストの仮想マシンです。これは、KVM (Kernel-based Virtual Machine) を実行している仮想マシンまたはサービスです。認証情報やクラスターを作成する時に、このホストの詳細が必要になります。以下の手順でプロビジョニングホストを設定します。

1. **プロビジョナーノードには SSH を使用してログインします。**
2. **root 以外のユーザー (user-name) を作成し、そのユーザーに sudo 権限に割り当てて、以下のコマンドを実行します。**

```
useradd <user-name>
passwd <password>
echo "<user-name> ALL=(root) NOPASSWD:ALL" | tee -a /etc/sudoers.d/<user-name>
chmod 0440 /etc/sudoers.d/<user-name>
```

3. **次のコマンドを入力して、新しいユーザーの SSH キーを作成します。**

```
su - <user-name> -c "ssh-keygen -t rsa -f /home/<user-name>/.ssh/id_rsa -N ""
```

4. **プロビジョナーノードに、新しいユーザーとしてログインします。**

```
su - <user-name>
[user-name@provisioner ~]$
```

5. **Red Hat Subscription Manager を使用して、以下のコマンドを入力してプロビジョナーノードを登録します。**

```
sudo subscription-manager register --username=<user-name> --password=<password> --auto-attach
sudo subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms --enable=rhel-8-for-x86_64-baseos-rpms
```

Red Hat Subscription Manager の詳細は、Red Hat OpenShift Container Platform ドキュメントの「Subscription Manager の使用および設定」を参照してください。

6. **以下のコマンドを実行して、必要なパッケージをインストールします。**

```
sudo dnf install -y libvirt qemu-kvm mkisofs python3-devel jq ipmitool
```

7. **ユーザーを変更して、新たに作成したユーザーに libvirt グループを追加します。**

```
sudo usermod --append --groups libvirt <user-name>
```

8. **以下のコマンドを入力して、firewalld を再起動し、http サービスを有効にします。**

```
sudo systemctl start firewalld
sudo firewall-cmd --zone=public --add-service=http --permanent
sudo firewall-cmd --add-port=5000/tcp --zone=libvirt  --permanent
sudo firewall-cmd --add-port=5000/tcp --zone=public   --permanent
sudo firewall-cmd --reload
```

9. 以下のコマンドを入力して、libvirtd サービスを起動し、有効にします。

```
sudo systemctl start libvirtd
sudo systemctl enable libvirtd --now
```

10. 次のコマンドを入力して、デフォルトのストレージプールを作成し、起動します。

```
sudo virsh pool-define-as --name default --type dir --target /var/lib/libvirt/images
sudo virsh pool-start default
sudo virsh pool-autostart default
```

11. ネットワーク設定の以下の例を参照してください。

- プロビジョニングネットワーク (IPv4アドレス)

```
sudo nohup bash -c """
    nmcli con down "$PROV_CONN"
    nmcli con delete "$PROV_CONN"
    # RHEL 8.1 appends the word "System" in front of the connection, delete in case it
exists
    nmcli con down "System $PROV_CONN"
    nmcli con delete "System $PROV_CONN"
    nmcli connection add ifname provisioning type bridge con-name provisioning
    nmcli con add type bridge-worker ifname "$PROV_CONN" master provisioning
    nmcli connection modify provisioning ipv4.addresses 172.22.0.1/24 ipv4.method
manual
    nmcli con down provisioning
    nmcli con up provisioning"""
```

この手順を完了すると、SSH 接続が切断される場合があります。

IPv4 アドレスは、ベアメタルネットワークでルーティングできなければ、どのようなアドレスでも構いません。

- プロビジョニングネットワーク (IPv6 アドレス)

```
sudo nohup bash -c """
    nmcli con down "$PROV_CONN"
    nmcli con delete "$PROV_CONN"
    # RHEL 8.1 appends the word "System" in front of the connection, delete in case it
exists
    nmcli con down "System $PROV_CONN"
    nmcli con delete "System $PROV_CONN"
    nmcli connection add ifname provisioning type bridge con-name provisioning
    nmcli con add type bridge-worker ifname "$PROV_CONN" master provisioning
    nmcli connection modify provisioning ipv6.addresses fd00:1101::1/64 ipv6.method
manual
    nmcli con down provisioning
    nmcli con up provisioning"""
```

この手順を完了すると、SSH 接続が切断される場合があります。

IPv6 アドレスは、ベアメタルネットワークでルーティングできなければ、どのようなアドレスでも構いません。

IPv6 アドレスを使用する場合に UEFI PXE 設定が有効にされており、UEFI PXE 設定が IPv6 プロトコルに設定されていることを確認します。

12. **ssh を使用してプロビジョナーノードに再接続します (必要な場合)。**

```
# ssh <user-name>@provisioner.<cluster-name>.<domain>
```

13. **以下のコマンドを実行して、接続ブリッジが正しく作成されていることを確認します。**

```
nmcli con show
```

検索結果は、以下のような内容になります。

名前	UUID	TYPE	DEVICE
baremetal	4d5133a5-8351-4bb9-bfd4-3af264801530	bridge	baremetal
provisioning	43942805-017f-4d7d-a2c2-7cb3324482ed	bridge	provisioning
virbr0	d9bca40f-eee1-410b-8879-a2d4bb0465e7	bridge	virbr0
bridge-worker-eno1	76a8ed50-c7e5-4999-b4f6-6d9014dd0812	ethernet	eno1
bridge-worker-eno2	f31c3353-54b7-48de-893a-02d2b34c4736	ethernet	eno2

14. 以下の手順で **pull-secret.txt** ファイルを作成します。

```
vim pull-secret.txt
```

   a. Webブラウザーで「Install OpenShift on Bare Metal with user-provisioned infrastructure」にアクセスし、「Downloads」セクションまでスクロールします。

   b. Copy pull secretをクリックします。

   c. その内容を **pull-secret.txt** に貼り付けて、**user-name** ユーザーのホームディレクトリーに保存します。

これで、ベアメタルの認証情報を作成する準備が整いました。

6.5.3. コンソールを使用したプロバイダー接続の作成

Red Hat Advanced Cluster Management for Kubernetes コンソールからプロバイダー接続を作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Clusters ページで Provider connections タブを選択します。
   既存のプロバイダー接続が表示されます。
3. Add connection を選択します。
4. プロバイダーとして Bare metal を選択します。
5. プロバイダー接続の名前を追加します。
6. 一覧からプロバイダー接続の namespace を選択します。
   ヒント: 便宜上およびセキュリティー上、プロバイダー接続のホスト専用の namespace を作成します。
7. オプションで、プロバイダー接続の ベース DNS ドメイン を追加できます。ベース DNS ドメインをプロバイダー接続に追加した場合には、このプロバイダー接続でクラスターを作成すると、このベース DNS ドメインは自動的に正しいフィールドに設定されます。
8. libvirt URI を追加します。libvirt URI は、ブートストラップノード向けに作成したプロビジョニングノードを追加してください。libvirt URI は以下の例のようになります。

```
<qemu+ssh>://<user-name>@<provision-host.com>/system
```

   - **qemu+ssh** は、プロビジョニングホスト上の libvirt デーモンに接続する方法に置き換えてください。

- **user-name** は、プロビジョニングホストにブートストラップノードを作成するアクセス権があるユーザー名に置き換えてください。

- **provision-host.com** は、プロビジョニングホストへのリンクに置き換えてください。これは、IP アドレスまたは完全修飾ドメイン名アドレスのいずれかです。
  詳細は、Connection URIs を参照してください。

9. プロビジョニングホストに SSH の既知ホストのリストを追加します。この値には、IP アドレスまたは完全修飾ドメイン名アドレスを指定できますが、libvirt URI 値で使用したのと同じ形式を使用することをお勧めします。

10. Red Hat OpenShift pull secretを入力します。Pull secretからプルシークレットをダウンロードします。

11. クラスターにアクセスできるように SSH 秘密鍵 と SSH 公開鍵 を追加します。既存のキーを使用するか、キー生成プログラムを使用して新しいキーを作成できます。キー生成の方法は、「SSH プライベートキーの生成およびエージェントへの追加」を参照してください。

12. オフラインインストールのみ: Configuration for disconnected installationサブセクションのフィールドに必要な情報を入力します。

    - Image registry mirror: この値には、オフラインのレジストリーパスを含みます。このパスには、オフラインインストールに使用する全インストールイメージのホスト名、ポート、レジストリーパスが含まれます。例: **repository.com:5000/openshift/ocp-release**.
      このパスは、Red Hat OpenShift Container Platform リリースイメージに対して、**install-config.yaml** のイメージコンテンツソースポリシーのマッピングを作成します。たとえば、**repository.com:5000** は以下の **imageContentSource** コンテンツを作成します。

      ```
      imageContentSources:
      - mirrors:
        - registry.example.com:5000/ocp4
        source: quay.io/openshift-release-dev/ocp-release-nightly
      - mirrors:
        - registry.example.com:5000/ocp4
        source: quay.io/openshift-release-dev/ocp-release
      - mirrors:
        - registry.example.com:5000/ocp4
        source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
      ```

    - Bootstrap OS image: この値には、ブートストラップマシンに使用するイメージの URL が含まれます。

    - Cluster OS image: この値には、Red Hat OpenShift Container Platform クラスターマシンに使用するイメージの URL が含まれます。

    - Additional trust bundle: この値で、ミラーレジストリーへのアクセスに必要な証明書ファイルのコンテンツを指定します。
      注記: 非接続環境にあるハブクラスターからマネージドクラスターをデプロイして、インストール後の設定を自動的にインポートする場合には、YAML エディターを使用してイメージコンテンツソースポリシーを **install-config.yaml** ファイルに追加します。エントリーの例を以下に示します。

      ```
      imageContentSources:
      - mirrors:
        - registry.example.com:5000/rhacm2
        source: registry.redhat.io/rhacm2
      ```

13. Create をクリックします。プロバイダー接続を作成すると、プロバイダー接続の一覧に追加されます。

「ベアメタルでのクラスターの作成」の手順を実行して、このプロバイダー接続を使用するクラスターを作成します。

6.5.4. プロバイダー接続の削除

プロバイダー接続を使用するクラスターを管理しなくなった場合には、そのプロバイダー接続を削除して、プロバイダー接続の情報を保護します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Provider connections を選択します。
3. 削除するプロバイダー接続の横にあるオプションメニューを選択します。
4. Delete connection を選択します。

第7章 クラスターの作成

Red Hat Advanced Cluster Management for Kubernetes を使用した、クラウドプロバイダー全体にクラスターを作成する方法を説明します。

- Amazon Web Services でのクラスターの作成
- Google Cloud Platform でのクラスターの作成
- Microsoft Azure でのクラスターの作成
- VMware vSphere でのクラスターの作成
- ベアメタルでのクラスターの作成

7.1. AMAZON WEB SERVICES でのクラスターの作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、Amazon Web Services (AWS) で Red Hat OpenShift Container Platform クラスターを作成できます。

7.1.1. 前提条件

AWS でクラスターを作成する前に、以下の前提条件を満たす必要があります。

- Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをデプロイしておく。
- Amazon Web Services で Kubernetes クラスターを作成できるようにする Red Hat Advanced Cluster Management for Kubernetes ハブクラスターでのインターネットアクセスがある。
- AWS プロバイダー接続。詳細は、「 Amazon Web Services のプロバイダー接続の作成 」を参照してください。
- AWS で設定されたドメイン。ドメインの設定方法は、「AWS アカウントの設定」を参照してください。
- ユーザー名、パスワード、アクセスキー ID およびシークレットアクセスキーなど、Amazon Web Services (AWS) のログイン認証情報。Understanding and getting your AWS credentials を参照してください。
- OpenShift Container Platform イメージプルシークレット。「イメージプルシークレットの使用」を参照してください。

注記: クラウドプロバイダーのアクセスキーを変更する場合は、プロビジョニングしたクラスターアクセスキーを手動で更新する必要があります。詳細は、既知の問題「プロビジョニングしたクラスターのシークレットの自動更新はサポートされない」を参照してください。

7.1.2. Red Hat Advanced Cluster Management for Kubernetes コンソールでのクラスターの作成

Red Hat Advanced Cluster Management for Kubernetes コンソールからクラスターを作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。

2. クラスターページで Add Cluster をクリックします。

3. Create a clusterを選択します。
注記: この手順では、クラスターを作成します。既存のクラスターをインポートする場合には、「ハブクラスターへのターゲットのマネージドクラスターのインポート」の手順を参照してください。

4. クラスターの名前を入力します。この名前はクラスターのホスト名で使用されます。
ヒント: コンソールに情報を入力する時に yaml コンテンツの更新内容を表示するには、YAML を ON に切り替えるように設定します。

5. インフラストラクチャープラットフォーム向けの Amazon Web Services を選択します。

6. クラスターに使用する リリースイメージ を指定します。このリリースイメージで、クラスターの作成に使用される OpenShift Container Platform イメージのバージョンを特定します。使用するバージョンが利用可能な場合は、イメージの一覧からイメージを選択できます。使用するイメージが標準イメージではない場合は、使用するイメージへの url を入力できます。リリースイメージの詳細は、「リリースイメージ」を参照してください。

7. 利用可能な接続一覧から、お使いのプロバイダー接続を選択します。プロバイダー接続が設定されていない場合や、新規のプロバイダー接続を設定する場合には、Add connection を参照してください。プロバイダー接続の作成に関する詳細は、「Amazon Web Services のプロバイダー接続の作成」を参照してください。

8. クラスターに関連付ける 追加のラベル を追加します。これらのラベルは、クラスターを特定し、検索結果を絞り込むのに役立ちます。

9. クラスターの ノードプール を設定します。
ノードプールは、クラスターに使用されるノードの場所とサイズを定義します。

   Region は、ノードの地理的な配置場所を指定します。リージョンが近くにある場合にはパフォーマンスの速度が向上しますが、リージョンの距離が離れると、より分散されます。

   - マスタープール: マスタープールには、クラスター向けに作成されたマスターノードが 3 つあります。マスターノードは、クラスターアクティビティーの管理を共有します。より分散されているマスターノードグループでは、リージョンで複数のゾーンを選択できます。インスタンスの作成後にインスタンスのタイプやサイズを変更できますが、このセクションで指定することもできます。デフォルト値は、ルートストレージ 100 GiB の mx5.xlarge - 4 vCPU, 16 GiB RAM - General Purpose です。

   - ワーカープール: ワーカープールにワーカーノードを作成して (作成しないことも可能)、クラスターのコンテナーワークロードを実行できます。ワーカーノードは、ワーカープール 1 つに所属することも、複数のワーカープールに分散させることもできます。ワーカーノードが指定されていない場合は、マスターノードもワーカーノードとして機能します。

10. 必要なクラスターネットワークオプションを設定します。AWS アカウントに設定したベース DNS 情報を入力します。選択したプロバイダー接続にベースドメインが紐付けされている場合には、その値がこのフィールドに設定されます。値を上書きすると変更できます。詳細は、「AWS アカウントの設定」を参照してください。この名前はクラスターのホスト名で使用されます。

11. オプション: クラスターのラベルを設定します。

12. Create をクリックします。作成およびインポートプロセスを完了すると、クラスターの詳細を表示できます。

注記: クラスターのインポートには、クラスターの詳細で提示された kubectl コマンドを実行する必要はありません。クラスターを作成すると、Red Hat Advanced Cluster Management で管理されるように自動的に設定されます。

7.1.3. クラスターへのアクセス

Red Hat Advanced Cluster Management for Kubernetes で管理されるクラスターにアクセスするには、以下の手順を実行します。

1. Red Hat Advanced Cluster Management ナビゲーションメニューで Automate infrastructure > Clusters に移動します。
2. 作成したクラスターまたはアクセスするクラスターの名前を選択します。クラスターの詳細が表示されます。
3. Reveal credentials を選択し、クラスターのユーザー名およびパスワードを表示します。クラスターにログインする時に使用するので、これらの値をメモしてください。
4. クラスターにリンクする Console URL を選択します。
5. 手順 3 で確認したユーザー ID およびパスワードを使用して、クラスターにログインします。
6. アクセスするクラスターの Actions >Launch to cluster を選択します。
   ヒント: ログイン認証情報がすでにわかっている場合は、アクセスしたいクラスターの Actions > Launch to cluster を選択することで、クラスターにアクセスできます。

7.2. MICROSOFT AZURE でのクラスターの作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、Microsoft Azure で Red Hat OpenShift Container Platform クラスターを作成できます。

7.2.1. 前提条件

Azure でクラスターを作成する前に、以下の前提条件を満たす必要があります。

- Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをデプロイしておく。
- Azure で Kubernetes クラスターを作成できるようにする Red Hat Advanced Cluster Management for Kubernetes ハブクラスターでのインターネットアクセスがある。
- Azure プロバイダー接続。詳細は、「Microsoft Azure のプロバイダー接続の作成」を参照してください。
- Azure で設定されたドメイン。ドメイン設定の方法は、Configuring a custom domain name for an Azure cloud service を参照してください。
- ユーザー名とパスワードなどの Azure ログイン認証情報。azure.microsoft.com を参照してください。
- clientId、clientSecret および tenantId などの Azure サービスプリンシパル。azure.microsoft.com を参照してください。
- OpenShift Container Platform イメージプルシークレット。「イメージプルシークレットの使用」を参照してください。

注記: クラウドプロバイダーのアクセスキーを変更する場合は、プロビジョニングしたクラスターアクセスキーを手動で更新する必要があります。詳細は、既知の問題「プロビジョニングしたクラスターのシークレットの自動更新はサポートされない」を参照してください。

7.2.2. Red Hat Advanced Cluster Management for Kubernetes コンソールでのクラスターの作成

Red Hat Advanced Cluster Management for Kubernetes コンソールからクラスターを作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Clusters ページで、Add cluster をクリックします。
3. Create a cluster を選択します。
   注記: この手順では、クラスターを作成します。既存のクラスターをインポートする場合には、「ハブクラスターへのターゲットのマネージドクラスターのインポート」の手順を参照してください。
4. クラスターの名前を入力します。この名前はクラスターのホスト名で使用されます。
   ヒント: コンソールに情報を入力する時に yaml コンテンツの更新内容を表示するには、YAML を ON に切り替えるように設定します。
5. インフラストラクチャープロバイダーに Microsoft Azure を選択します。
6. クラスターに使用する リリースイメージ を指定します。このリリースイメージで、クラスターの作成に使用される OpenShift Container Platform イメージのバージョンを特定します。使用するバージョンが利用可能な場合は、イメージの一覧からイメージを選択できます。使用するイメージが標準イメージではない場合は、使用するイメージの URL を入力できます。リリースイメージの詳細は、「リリースイメージ」を参照してください。
7. 利用可能な接続一覧から、お使いのプロバイダー接続を選択します。プロバイダー接続が設定されていない場合や、新規のプロバイダー接続を設定する場合には、Add connection を参照してください。プロバイダー接続の作成に関する詳細は、「Microsoft Azure のプロバイダー接続の作成」を参照してください。
8. クラスターに関連付ける 追加のラベル を追加します。これらのラベルは、クラスターを特定し、検索結果を絞り込むのに役立ちます。
9. クラスターの ノードプール を設定します。
   ノードプールは、クラスターに使用されるノードの場所とサイズを定義します。

   Region は、ノードの地理的な配置場所を指定します。リージョンが近くにある場合にはパフォーマンスの速度が向上しますが、リージョンの距離が離れると、より分散されます。

   - マスタープール: マスタープールには、クラスター向けに作成されたマスターノードが 3 つあります。マスターノードは、クラスターアクティビティーの管理を共有します。より分散されているマスターノードグループでは、リージョンで複数のゾーンを選択できます。インスタンスの作成後にインスタンスのタイプやサイズを変更できますが、このセクションで指定することもできます。デフォルト値は、ルートストレージ 128 GiB の Standard_D4s_v3 - 4 vCPU, 16 GiB RAM - General Purposeです。
   - ワーカープール: ワーカープールにワーカーノードを作成して (作成しないことも可能)、クラスターのコンテナーワークロードを実行できます。ワーカーノードは、ワーカープール 1 つに所属することも、複数のワーカープールに分散させることもできます。ワーカーノードが指定されていない場合は、マスターノードもワーカーノードとして機能します。

10. 必要なクラスターネットワークオプションを設定します。
Azure アカウントに設定したベー DNS 情報を入力します。選択したプロバイダー接続にベース DNS が紐付けされている場合は、その値がこのフィールドに設定されます。値を上書きすると変更できます。詳細は、Configuring a custom domain name for an Azure cloud serviceを参照してください。この名前はクラスターのホスト名で使用されます。

11. オプション: クラスターのラベルを設定します。

12. Create をクリックします。作成およびインポートプロセスを完了すると、クラスターの詳細を表示できます。
注記: クラスターのインポートには、クラスターの詳細で提示された **kubectl** コマンドを実行する必要はありません。クラスターを作成すると、Red Hat Advanced Cluster Management for Kubernetes で管理されるように自動的に設定されます。

7.2.3. クラスターへのアクセス

Red Hat Advanced Cluster Management for Kubernetes で管理されるクラスターにアクセスするには、以下の手順を実行します。

1. Red Hat Advanced Cluster Management for Kubernetes ナビゲーションメニューで Automate infrastructure >Clusters に移動します。

2. 作成したクラスターまたはアクセスするクラスターの名前を選択します。クラスターの詳細が表示されます。

3. Reveal credentials を選択し、クラスターのユーザー名およびパスワードを表示します。クラスターにログインする時に使用するので、これらの値をメモしてください。

4. クラスターにリンクする Console URL を選択します。

5. 手順 3 で確認したユーザー ID およびパスワードを使用して、クラスターにログインします。

6. アクセスするクラスターの Actions >Launch to clusterを選択します。
ヒント: ログイン認証情報がすでにわかっている場合は、アクセスしたいクラスターの Actions > Launch to clusterを選択することで、クラスターにアクセスできます。

7.3. GOOGLE CLOUD PLATFORM でのクラスターの作成

Google Cloud Platform (GCP) で Red Hat OpenShift Container Platform クラスターを作成する手順に従います。Google Cloud Platform の詳細は、Google Cloud Platform を参照してください。

7.3.1. 前提条件

GCP でクラスターを作成する前に、以下の前提条件を満たす必要があります。

- Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをデプロイしておく。

- GCP で Kubernetes クラスターを作成できるようにする Red Hat Advanced Cluster Management for Kubernetes ハブクラスターでのインターネットアクセスがある。

- GCP プロバイダー接続。詳細は、「Google Cloud Platform のプロバイダー接続の作成」を参照してください。

- GCP に設定されたドメイン。ドメインの設定方法は、Setting up a custom domainを参照してください。
- ユーザー名とパスワードなどの GCP ログイン認証情報。
- OpenShift Container Platform イメージプルシークレット。「イメージプルシークレットの使用」を参照してください。

注記: クラウドプロバイダーのアクセスキーを変更する場合は、プロビジョニングしたクラスターアクセスキーを手動で更新する必要があります。詳細は、既知の問題「プロビジョニングしたクラスターのシークレットの自動更新はサポートされない」を参照してください。

7.3.2. Red Hat Advanced Cluster Management for Kubernetes コンソールでのクラスターの作成

Red Hat Advanced Cluster Management for Kubernetes コンソールからクラスターを作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Clusters ページで、Add cluster をクリックします。
3. Create a clusterを選択します。
   注記: この手順では、クラスターを作成します。既存のクラスターをインポートする場合には、「ハブクラスターへのターゲットのマネージドクラスターのインポート」の手順を参照してください。
4. クラスターの名前を入力します。この名前はクラスターのホスト名で使用されます。GCP クラスターの命名に適用される制限がいくつかあります。この制限には、名前を goog で開始しないことや、名前に google に類似する文字および数字のグループが含まれないことなどがあります。制限の完全な一覧は、「Bucket naming guidelines」を参照してください。
   ヒント: コンソールに情報を入力する時に yaml コンテンツの更新内容を表示するには、YAML を ON に切り替えるように設定します。
5. インフラストラクチャープロバイダーに Google Cloud を選択します。
6. クラスターに使用する リリースイメージ を指定します。このリリースイメージで、クラスターの作成に使用される OpenShift Container Platform イメージのバージョンを特定します。使用するバージョンが利用可能な場合は、イメージの一覧からイメージを選択できます。使用するイメージが標準イメージではない場合は、使用するイメージの URL を入力できます。リリースイメージの詳細は、「リリースイメージ」を参照してください。
7. 利用可能な接続一覧から、お使いのプロバイダー接続を選択します。プロバイダー接続が設定されていない場合や、新規のプロバイダー接続を設定する場合には、Add connection を参照してください。プロバイダー接続の作成に関する詳細は、「Google Cloud Platform のプロバイダー接続の作成」を参照してください。
8. クラスターに関連付ける 追加のラベル を追加します。これらのラベルは、クラスターを特定し、検索結果を絞り込むのに役立ちます。
9. クラスターの ノードプール を設定します。
   ノードプールは、クラスターに使用されるノードの場所とサイズを定義します。

   Region は、ノードの地理的な配置場所を指定します。リージョンが近くにある場合にはパフォーマンスの速度が向上しますが、リージョンの距離が離れると、より分散されます。

   - マスタープール: マスタープールには、クラスター向けに作成されたマスターノードが 3 つ

あります。マスターノードは、クラスターアクティビティーの管理を共有します。より分散されているマスターノードグループでは、リージョンで複数のゾーンを選択できます。インスタンスの作成後にインスタンスのタイプやサイズを変更できますが、このセクションで指定することもできます。デフォルト値は、ルートストレージ 500 GiB の n1-standard-1 - n1-standard-11 vCPU - General Purpose です。

- ワーカープール: ワーカープールにワーカーノードを作成して (作成しないことも可能)、クラスターのコンテナーワークロードを実行できます。ワーカーノードは、ワーカープール 1 つに所属することも、複数のワーカープールに分散させることもできます。ワーカーノードが指定されていない場合は、マスターノードもワーカーノードとして機能します。

10. 必要なクラスターネットワークオプションを設定します。
Google Cloud Platform アカウントに設定したベース DNS 情報を入力します。選択したプロバイダー接続にベースドメインが紐付けされている場合は、その値がこのフィールドに設定されます。値を上書きすると変更できます。詳細は、Setting up a custom domainを参照してください。この名前はクラスターのホスト名で使用されます。

11. オプション: クラスターのラベルを設定します。

12. Create をクリックします。

作成およびインポートプロセスを完了すると、クラスターの詳細を表示できます。

+注記: クラスターのインポートには、クラスターの詳細で提示された **kubectl** コマンドを実行する必要はありません。クラスターを作成すると、Red Hat Advanced Cluster Management for Kubernetes で管理されるように自動的に設定されます。

7.3.3. クラスターへのアクセス

Red Hat Advanced Cluster Management for Kubernetes で管理されるクラスターにアクセスするには、以下の手順を実行します。

1. Red Hat Advanced Cluster Management for Kubernetes ナビゲーションメニューで Automate infrastructure >Clusters に移動します。

2. 作成したクラスターまたはアクセスするクラスターの名前を選択します。クラスターの詳細が表示されます。

3. Reveal credentials を選択し、クラスターのユーザー名およびパスワードを表示します。クラスターにログインする時に使用するので、これらの値をメモしてください。

4. クラスターにリンクする Console URL を選択します。

5. 手順 3 で確認したユーザー ID およびパスワードを使用して、クラスターにログインします。

6. アクセスするクラスターの Actions >Launch to clusterを選択します。
ヒント: ログイン認証情報がすでにわかっている場合は、アクセスしたいクラスターの Actions > Launch to clusterを選択することで、クラスターにアクセスできます。

7.4. VMWARE VSPHERE でのクラスターの作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、VMware vSphere で Red Hat OpenShift Container Platform クラスターを作成できます。

7.4.1. 前提条件

vSphere でクラスターを作成する前に、以下の前提条件を満たす必要があります。

- OpenShift Container Platform バージョン 4.5 以降に Red Hat Advanced Cluster Management ハブクラスターをデプロイしておく。
- vSphere で Kubernetes クラスターを作成できるように Red Hat Advanced Cluster Management ハブクラスターでのインターネットアクセスがある。
- vSphere プロバイダー接続。詳細は、「VMware vSphere のプロバイダー接続の作成」を参照してください。
- Red Hat OpenShift イメージプルシークレット。「イメージプルシークレットの使用」を参照してください。
- デプロイする VMware インスタンスについての以下の情報。
  - API および Ingress インスタンスに必要な静的 IP アドレス
  - 以下の DNS レコード
    - api.<cluster_name>.<base_domain>。静的 API VIP を参照する必要があります。
    - *.apps.<cluster_name>.<base_domain>。Ingress VIP の静的 IP アドレスを参照する必要があります。

7.4.2. Red Hat Advanced Cluster Management for Kubernetes コンソールでのクラスターの作成

Red Hat Advanced Cluster Management コンソールからクラスターを作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Clusters ページで、Add cluster をクリックします。
3. Create a cluster を選択します。
   注記: この手順では、クラスターを作成します。既存のクラスターをインポートする場合には、「ハブクラスターへのターゲットのマネージドクラスターのインポート」の手順を参照してください。
4. クラスターの名前を入力します。この名前はクラスターのホスト名で使用されます。
   注記: 値は、プロバイダー接続の要件セクションに記載されている DNS レコードの作成に使用した名前と一致させる必要があります。

   ヒント: コンソールに情報を入力する時に yaml コンテンツの更新内容を表示するには、YAML を ON に切り替えるように設定します。
5. インフラストラクチャープロバイダーに VMware vSphere を選択します。
6. クラスターに使用する リリースイメージ を指定します。このリリースイメージで、クラスターの作成に使用される OpenShift Container Platform イメージのバージョンを特定します。使用するバージョンが利用可能な場合は、イメージの一覧からイメージを選択できます。使用するイメージが標準イメージではない場合は、使用するイメージへの URL を入力できます。詳細は、「リリースイメージ」を参照してください。注記: OpenShift Container Platform バージョン 4.5.x 以降のリリースイメージのみがサポートされます。
7. 利用可能な接続一覧から、お使いのプロバイダー接続を選択します。プロバイダー接続が設定

されていない場合や、新規のプロバイダー接続を設定する場合には、Add connection を参照してください。プロバイダー接続の作成に関する詳細は、「プロバイダー接続の作成」を参照してください。

8. vSphere アカウントに設定したベースドメイン情報を入力します。選択したプロバイダー接続にベースドメインが紐付けされている場合には、その値がこのフィールドに設定されます。値を上書きすると変更できます。注記: 値は、要件セクションに記載されている DNS レコードの作成に使用した名前と一致させる必要があります。この名前はクラスターのホスト名で使用されます。

9. クラスターに関連付ける 追加のラベル を追加します。これらのラベルは、クラスターを特定し、検索結果を絞り込むのに役立ちます。

10. クラスターの ノードプール を設定します。
ノードプールは、クラスターに使用されるノードの場所とサイズを定義します。

    ワーカープールに 1 つまたは複数のワーカーノードを作成し、クラスターのコンテナーワークロードを実行できます。ワーカーノードは、ワーカープール 1 つに所属することも、複数のワーカープールに分散させることもできます。

11. クラスターのネットワークオプションを設定します。これらのオプションについては以下のリストで紹介します。

    a. vSphere ネットワーク名: VMware vSphere ネットワーク名

    b. API VIP: 内部 API 通信に使用する IP アドレス注記:: 値は、要件セクションに記載されている DNS レコードの作成に使用した名前と一致させる必要があります。指定しない場合には、DNS を事前設定して **api.** が正しく解決されるようにします。

    c. Ingress VIP: Ingress トラフィックに使用する IP アドレス注記:: 値は、要件セクションに記載されている DNS レコードの作成に使用した名前と一致させる必要があります。指定しない場合には、DNS を事前設定して **test.apps.** が正しく解決されるようにします。

12. オプション: クラスターのラベルを設定します。

13. Create をクリックします。作成およびインポートプロセスを完了すると、クラスターの詳細を表示できます。
注記: クラスターを作成すると、Red Hat Advanced Cluster Management で管理されるように自動的に設定されます。クラスターのインポートには、クラスターの詳細で提示された **kubectl** コマンドを実行する必要はありません。

7.4.3. クラスターへのアクセス

Red Hat Advanced Cluster Management で管理されるクラスターにアクセスするには、以下の手順を実行します。

1. すでにログイン情報が分かっている場合には、クラスターの Options メニューにアクセスして、Launch to cluster を選択します。

2. ログイン認証情報が分からない場合:

    a. Red Hat Advanced Cluster Management ナビゲーションメニューで Automate infrastructure >Clusters に移動します。

    b. 作成したクラスターまたはアクセスするクラスターの名前を選択します。クラスターの詳細が表示されます。

c. Reveal credentials を選択し、クラスターのユーザー名およびパスワードを表示します。クラスターへのログイン時にこの値を使用します。

3. クラスターにリンクする Console URL を選択します。

4. 手順 3 で確認したユーザー ID およびパスワードを使用して、クラスターにログインします。

5. アクセスするクラスターの Actions >Launch to clusterを選択します。
ヒント: ログイン認証情報がすでにわかっている場合は、アクセスしたいクラスターの Actions > Launch to clusterを選択することで、クラスターにアクセスできます。

7.5. ベアメタルでのクラスターの作成

Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、ベアメタル環境で Red Hat OpenShift Container Platform クラスターを作成できます。

7.5.1. 前提条件

ベアメタル環境にクラスターを作成する前に、以下の前提条件を満たす必要があります。

- OpenShift Container Platform バージョン 4.5 以降に、Red Hat Advanced Cluster Management for Kubernetes ハブクラスターをデプロイしておく。
- クラスターを作成するために必要なイメージを取得するための、Red Hat Advanced Cluster Management for Kubernetes ハブクラスターへのインターネットアクセス (接続済み)、あるいはインターネットへの接続がある内部またはミラーレジストリーへの接続 (非接続) がある。
- Hive クラスターの作成に使用されるブートストラップ仮想マシンを実行する一時的な外部 KVM ホスト。詳細は、「 プロビジョニングホストの準備 」を参照してください。
- ベアメタルサーバーのログイン資格情報。これには、前の項目のブートストラップ仮想マシンからのlibvirt URI、SSH秘密鍵、およびSSHの既知のホストのリストが含まれます。詳細は、「 OpenShift インストール環境の設定 」を参照してください。
- ベアメタルのプロバイダー接続。詳細は、「 ベアメタルのプロバイダー接続の作成 」を参照してください。
- ユーザー名、パスワード、ベースボード管理コントローラー (BMC) アドレスなどのベアメタル環境のログイン認証情報。
- ベアメタルアセットが証明書の検証を有効にしている場合には、ベアメタルアセットを設定します。詳細は、「 ベアメタルアセットの作成および変更 」を参照してください。
- OpenShift Container Platform イメージプルシークレット。「イメージプルシークレットの使用」を参照してください。
注記:
  - ベアメタルアセット、ベアメタルのマネージドクラスター、および関連シークレットは同じ namespace に配置する必要があります。
  - クラウドプロバイダーのアクセスキーを変更する場合は、プロビジョニングしたクラスターアクセスキーを手動で更新する必要があります。詳細は、既知の問題「プロビジョニングしたクラスターのシークレットの自動更新はサポートされない」を参照してください。

7.5.2. Red Hat Advanced Cluster Management コンソールでのクラスターの作成

Red Hat Advanced Cluster Management コンソールからクラスターを作成するには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. Clusters ページで、Add cluster をクリックします。
3. Create a cluster を選択します。
   注記: この手順では、クラスターを作成します。既存のクラスターをインポートする場合には、「ハブクラスターへのターゲットのマネージドクラスターのインポート」の手順を参照してください。
4. クラスターの名前を入力します。ベアメタルクラスターの場合には、名前を任意で指定できません。この名前は、クラスター URL に関連付けられています。使用するクラスター名が DNS およびネットワーク設定と一致していることを確認します。
   ヒント: コンソールに情報を入力する時に yaml コンテンツの更新内容を表示するには、YAML を ON に切り替えるように設定します。
5. インフラストラクチャープロバイダーに Bare Metal を選択します。
6. クラスターに使用する リリースイメージ を指定します。このリリースイメージで、クラスターの作成に使用される Red Hat OpenShift Container Platform イメージのバージョンを特定します。使用するバージョンが利用可能な場合は、イメージの一覧からイメージを選択できます。使用するイメージが標準イメージではない場合は、使用するイメージの URL を入力できます。リリースイメージの詳細は、「リリースイメージ」を参照してください。
7. 利用可能な接続一覧から、お使いのプロバイダー接続を選択します。プロバイダー接続が設定されていない場合や、新規のプロバイダー接続を設定する場合には、Add connection を参照してください。プロバイダー接続の作成に関する詳細は、「ベアメタルのプロバイダー接続の作成」を参照してください。
8. オプション: クラスターの追加ラベルを設定します。
9. プロバイダー接続に関連付けられたホスト一覧から、お使いのホストを選択します。ハイパーバイザーと同じブリッジネットワークにあるアセットを 3 つ以上選択します。
   ホストの一覧は、既存のベアメタルアセットからコンパイルされます。ベアメタルアセットを作成していない場合は、作成プロセスを続行する前に作成またはインポートを行うことができます。Disable certificate verification を選択して要件を無視することができます。
10. クラスターネットワークオプションを設定します。

パラメーター	説明	必須またはオプション
ベース DNS ドメイン	プロバイダーのベースドメインは、Red Hat OpenShift Container Platform クラスターコンポーネントへのルートの作成に使用されます。これは、クラスタープロバイダーの DNS で Start of Authority (SOA) レコードとして設定されます。この設定は、クラスターの作成後に変更できません。	必須

パラメーター	説明	必須またはオプション
ネットワークタイプ	デプロイする Pod ネットワークプロバイダープラグイン。OpenShift Container Platform 4.3 でサポートされるのは、OpenShiftSDN プラグインのみです。OVNKubernetes プラグインは、OpenShift Container Platform 4.3、4.4、および 4.5 でテクノロジープレビューとしてご利用いただけます。通常、これは OpenShift Container Platform バージョン 4.6 以降で利用できます。OVNKubernetes は IPv6 と共に使用する必要があります。デフォルト値は **OpenShiftSDN** です。	必須
クラスターのネットワーク CIDR	Pod IP アドレスの割り当てに使用する IP アドレスのブロック。OpenShiftSDN ネットワークプラグインは複数のクラスターネットワークをサポートします。複数のクラスターネットワークのアドレスブロックには重複が許可されません。予想されるワークロードに適したサイズのアドレスプールを選択してください。デフォルト値は、10.128.0.0/14 です。	必須
ネットワークホストのプレフィックス	それぞれの個別ノードに割り当てるサブネットプレフィックスの長さ。たとえば、hostPrefix が 23 に設定される場合、各ノードに指定の cidr から /23 サブネットが割り当てられます (510 (2^(32 - 23) - 2) Pod IP アドレスが許可されます)。デフォルトは 23 です。	必須

パラメーター	説明	必須またはオプション
サービスネットワーク CIDR	サービスの IP アドレスのブロック。OpenShiftSDN が許可するのは serviceNetwork ブロック 1 つだけです。このアドレスは他のネットワークブロックと重複できません。デフォルト値は 172.30.0.0/16 です。	必須
マシン CIDR	OpenShift Container Platform ホストで使用される IP アドレスのブロック。このアドレスブロックは他のネットワークブロックと重複できません。デフォルト値は 10.0.0.0/16 です。	必須
プロビジョニングネットワーク CIDR	プロビジョニングに使用するネットワークの CIDR。このサンプル形式は 172.30.0.0/16 です。	必須
プロビジョニングネットワークインターフェース	プロビジョニングネットワークに接続されたコントロールプレーンノード上のネットワークインターフェース名。	必須
プロビジョニングネットワークブリッジ	プロビジョニングネットワークに接続されているハイパーバイザーのブリッジ名。	必須
外部ネットワークブリッジ	外部ネットワークに接続されているハイパーバイザーのブリッジ名。	必須
DNS VIP	内部 DNS 通信に使用する仮想 IPこのパラメーターは、OpenShift Container Platform バージョン 4.4 以前にのみ適用されます。	OpenShift Container Platform 4.4 以前のバージョンでは必要です。
API VIP	内部 API 通信に使用する仮想 IP**api.<cluster_name>.<Base DNS domain>** パスが正しく解決されるように、DNS は A/AAAA または CNAME レコードで事前設定する必要があります。	必須

パラメーター	説明	必須またはオプション
Ingress VIP	Ingress トラフィックに使用する仮想 IP。***.apps.<cluster_name>.<Base DNS domain>** パスが正しく解決されるように、DNS は A/AAAA または CNAME レコードで事前設定する必要があります。	オプション

11. **オプション: Configmap の追加設定を変更する場合には、詳細設定を更新します。**

12. **Create をクリックします。作成およびインポートプロセスを完了すると、クラスターの詳細を表示できます。**
注記: クラスターのインポートには、クラスターの詳細で提示された kubectl コマンドを実行する必要はありません。クラスターを作成すると、Red Hat Advanced Cluster Management for Kubernetes で管理されるように自動的に設定されます。

7.5.3. クラスターへのアクセス

Red Hat Advanced Cluster Management for Kubernetes で管理されるクラスターにアクセスするには、以下の手順を実行します。

1. **Red Hat Advanced Cluster Management for Kubernetes ナビゲーションメニューで Automate infrastructure >Clusters に移動します。**

2. **作成したクラスターまたはアクセスするクラスターの名前を選択します。クラスターの詳細が表示されます。**

3. **Reveal credentials を選択し、クラスターのユーザー名およびパスワードを表示します。クラスターにログインする時に使用するので、これらの値をメモしてください。**

4. **クラスターにリンクする Console URL を選択します。**

5. **手順 3 で確認したユーザー ID およびパスワードを使用して、クラスターにログインします。**

6. **アクセスするクラスターの Actions >Launch to cluster を選択します。**
ヒント: ログイン認証情報がすでにわかっている場合は、アクセスしたいクラスターの Actions > Launch to cluster を選択することで、クラスターにアクセスできます。

第8章 ハブクラスターへのターゲットのマネージドクラスターのインポート

別の Kubernetes クラウドプロバイダーからクラスターをインポートできます。インポートすると、ターゲットクラスターは Red Hat Advanced Cluster Management for Kubernetes ハブクラスターのマネージドクラスターになります。指定されていない場合には、ハブクラスターとターゲットのマネージドクラスターにアクセスできる場所で、インポートタスクを実行します。

ハブクラスターは 他 のハブクラスターの管理はできず、自己管理のみが可能です。ハブクラスターは、自動的にインポートして自己管理できるように設定されています。ハブクラスターは手動でインポートする必要はありません。

ただし、ハブクラスターを削除して、もう一度インポートする場合は、**local-cluster:true** ラベルを追加する必要があります。

コンソールまたは CLI からのマネージドクラスターの設定は、以下の手順から選択します。

必要なユーザータイプまたはアクセスレベル: クラスター管理者

- コンソールを使用した既存クラスターのインポート
- CLI を使用したマネージドクラスターのインポート
- クラスターの klusterlet アドオン設定の変更

8.1. コンソールを使用した既存クラスターのインポート

Red Hat Advanced Cluster Management for Kubernetes をインストールすると、管理するクラスターをインポートする準備が整います。コンソールと CLI の両方からインポートできます。コンソールからインポートするには、以下の手順に従います。この手順では、認証用にターミナルが必要です。

- 前提条件
- クラスターのインポート
- クラスターの削除

8.1.1. 前提条件

- Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをデプロイしておく必要がある。ベアメタルクラスターをインポートする場合には、ハブクラスターを Red Hat OpenShift Container Platform バージョン 4.5 以降にインストールする必要があります。
- 管理するクラスターとインターネット接続が必要である。
- **kubectl** をインストールしておく必要がある。**kubectl** のインストール手順は、Kubernetes ドキュメント の「Install and Set Up kubectl」 を参照してください。
- **Base64** コマンドラインツールが必要である。
- Red Hat OpenShift Dedicated 環境にインポートする場合の前提条件:
  - ハブクラスターを Red Hat OpenShift Dedicated 環境にデプロイしている必要があります。
  - Red Hat OpenShift Dedicated のデフォルトパーミッションを dedicated-admin ですが、

namespace を作成するためのパーミッションがすべて含まれているわけではありません。Red Hat Advanced Cluster Management for Kubernetes でクラスターをインポートして管理するには **cluster-admin** パーミッションが必要です。

必要なユーザータイプまたはアクセスレベル: クラスター管理者

8.1.2. クラスターのインポート

利用可能なクラウドプロバイダーごとに、Red Hat Advanced Cluster Management for Kubernetes コンソールから既存のクラスターをインポートできます。

注記: ハブクラスターは別のハブクラスターを管理できません。ハブクラスターは、自動的にインポートおよび自己管理するように設定されるため、ハブクラスターを手動でインポートして自己管理する必要はありません。

1. ナビゲーションメニューで Automate infrastructure にマウスをかざし、Clusters をクリックします。
2. Add cluster をクリックします。
3. Import an existing cluster をクリックします。
4. クラスターの名前を指定します。デフォルトで、namespace はクラスター名と namespace に使用されます。
5. オプション: ラベル を追加します。
   注記: Red Hat OpenShift Dedicated クラスターをインポートし、**vendor=OpenShiftDedicated** のラベルを追加してベンダーが指定されないようにする場合、または **vendor=auto-detect** のラベルを追加する場合には **managed-by=platform** ラベルがクラスターに自動的に追加されます。この追加ラベルを使用して、クラスターを Red Hat OpenShift Dedicated クラスターとして識別し、Red Hat OpenShift Dedicated クラスターをグループとして取得できます。
6. Save import and generate code をクリックし、**open-cluster-management-agent-addon** のデプロイに使用するコマンドを生成します。確認メッセージが表示されます。
7. オプション: **oc get managedcluster** コマンドを実行する際に、テーブルに表示される URL を設定して、クラスターの詳細ページにある Cluster API アドレス を設定します。
   a. **cluster-admin** パーミッションがある ID でハブクラスターにログインします。
   b. ターゲットのマネージドクラスターの **kubectl** を設定します。
   kubectl の設定方法は、「サポート対象のクラウド」を参照してください。
   c. 以下のコマンドを入力して、インポートしているクラスターのマネージドクラスターエントリーを編集します。

   ```
   oc edit managedcluster <cluster-name>
   ```

   cluster-name は、マネージドクラスターの名前に置き換えます。
   d. 以下の例のように、YAML ファイルの **ManagedCluster** 仕様に **manageClusterClientConfigs** セクションを追加します。

   ```
   spec:
     hubAcceptsClient: true
   ```

```
managedClusterClientConfigs:
- url: https://multicloud-console.apps.new-managed.dev.redhat.com
```

URL の値を、インポートするマネージドクラスターへの外部アクセスを提供する URL に置き換えます。

8. Import an existing clusterウィンドウで Copy command を選択し、生成されたコマンドおよびトークンをクリップボードにコピーします。
重要: コマンドには、インポートした各クラスターにコピーされるプルシークレット情報が含まれます。インポートしたクラスターにアクセスできるユーザーであれば誰でも、プルシークレット情報を表示することもできます。https://cloud.redhat.com/ で 2 つ目のプルシークレットを作成することを検討するか、サービスアカウントを作成して個人の認証情報を保護してください。プルシークレットの詳細は、「イメージプルシークレットの使用」または「サービスアカウントの概要および作成」を参照してください。

9. インポートするマネージドクラスターにログインします。

10. Red Hat OpenShift Dedicated 環境のみ対象: 以下の手順を実行します。

    a. マネージドクラスターで **open-cluster-management-agent** および **open-cluster-management** namespace またはプロジェクトを作成します。

    b. OpenShift Container Platform カタログで klusterlet Operator を検索します。

    c. 作成した **open-cluster-management** namespace またはプロジェクトにインストールします。
    重要: **open-cluster-management-agent** namespace に Operator をインストールしないでください。

    d. 以下の手順を実行して、import コマンドからブートストラップシークレットを展開します。

        i. import コマンドを生成します。

            A. Red Hat Advanced Cluster Management コンソールで、Automate infrastructure >Clusters を選択します。

            B. Add a cluster >Import an existing clusterを選択します。

            C. クラスター情報を追加し、Save import and generate codeを選択します。

        ii. import コマンドをコピーします。

        iii. **import-command** という名前で作成したファイルに、import コマンドを貼り付けます。

        iv. Red Hat Advanced Cluster Management バージョン 2.2.x のバージョンに応じて、以下のコマンドを実行して import コマンドをデコードします。

        - Red Hat Advanced Cluster Management バージョン 2.2.x のアップグレード済みの z-stream バージョン (例: 2.2.2) を実行している場合は、以下のコマンドを実行します。

```
cat import-command | awk '{split($0,a,"&&"); print a[3]}' | awk '{split($0,a,"|"); print a[1]}' | sed -e "s/^ echo //" | sed 's/\"//g' | base64 -d
```

- Red Hat Advanced Cluster Management バージョン 2.2 のバージョンを後続の z-stream にアップグレードしていない場合は、以下のコマンドを実行します (例: バージョン 2.2.0)。

```
cat import-command | awk '{split($0,a,"&&"); print a[3]}' | awk '{split($0,a,"|"); print a[1]}' | sed -e "s/^ echo //" | base64 -d
```

v. 出力で **bootstrap-hub-kubeconfig** という名前のシークレットを見つけ、コピーします。

vi. シークレットをマネージドクラスターの **open-cluster-management-agent** namespace に適用します。

vii. インストールした Operator の例を使用して klusterlet リソースを作成します。clusterName は、インポート中に設定されたクラスター名と同じ名前に変更する必要があります。
注記: **managedcluster** リソースがハブに正しく登録されると、2 つの klusterlet Operator がインストールされます。klusterlet Operator の 1 つは **open-cluster-management** namespace に、もう 1 つは**open-cluster-management-agent** namespace にあります。Operator が複数あっても klusterlet の機能には影響はありません。

11. Red OpenShift Dedicated 環境に含まれていないクラスターのインポート: 以下の手順を実行します。

a. 必要な場合は、マネージドクラスターの **kubectl** コマンドを設定します。**kubectl** コマンドラインインターフェースの設定方法は、「サポート対象のクラウド」を参照してください。

b. マネージドクラスターに **open-cluster-management-agent-addon** をデプロイするには、コピーしたトークンでコマンドを実行します。

12. View cluster をクリックして Overview ページのクラスターの概要を表示します。

クラスターがインポートされました。Import another を選択すると、さらにインポートできます。

8.1.3. インポートされたクラスターの削除

以下の手順を実行して、インポートされたクラスターと、マネージドクラスターで作成された **open-cluster-management-agent-addon** を削除します。

1. Clusters ページの表から、インポートされたクラスターを見つけます。

2. Actions >Detach cluster をクリックしてマネージメントからクラスターを削除します。

注記: **local-cluster** という名前のハブクラスターをデタッチしようとする場合には、デフォルトの **disableHubSelfManagement** 設定が **false** である点に注意してください。この設定が原因で、ハブクラスターはデタッチされると、自身を再インポートして管理し、**MultiClusterHub** コントローラーが調整されます。ハブクラスターがデタッチプロセスを完了して再インポートするのに時間がかかる場合があります。プロセスが終了するのを待たずにハブクラスターを再インポートする場合には、以下のコマンドを実行して **multiclusterhub-operator** Pod を再起動して、再インポートの時間を短縮できます。

```
oc delete po -n open-cluster-management `oc get pod -n open-cluster-management | grep multiclusterhub-operator| cut -d' ' -f1`
```

「ネットワーク接続時のオンラインインストール」で説明されているように、disableHubSelfManagement の値を true に指定して、自動的にインポートされないように、ハブクラスターの値を変更できます。

8.2. CLI を使用したマネージドクラスターのインポート

Red Hat Advanced Cluster Management for Kubernetes をインストールすると、管理するクラスターをインポートする準備が整います。コンソールと CLI の両方からインポートできます。以下の手順に従って、CLI からインポートします。

- 前提条件
- サポート対象のアーキテクチャー
- klusterlet のインポート

重要: ハブクラスターは別のハブクラスターを管理できません。ハブクラスターは、自動でインポートおよび自己管理するように設定されます。ハブクラスターは、手動でインポートして自己管理する必要はありません。

ただし、ハブクラスターを削除して、もう一度インポートする場合は、local-cluster:true ラベルを追加する必要があります。

8.2.1. 前提条件

- Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをデプロイしておく必要がある。ベアメタルクラスターをインポートする場合には、ハブクラスターを Red Hat OpenShift Container Platform バージョン 4.5 以降にインストールする必要があります。
- 管理予定の別のクラスターとインターネット接続が必要である。
- oc コマンドを実行するには、Red Hat OpenShift Container Platform の CLI バージョン 4.5 以降が必要である。Red Hat OpenShift CLI (oc) のインストールおよび設定の詳細は、「CLI の使用方法」を参照してください。
- Kubernetes CLI (kubectl) をインストールする必要がある。kubectl のインストール手順は、Kubernetes ドキュメント の「Install and Set Up kubectl」 を参照してください。
  注記: コンソールから CLI ツールのインストールファイルをダウンロードします。

8.2.2. サポート対象のアーキテクチャー

- Linux (x86_64, s390x)
- macOS

8.2.3. インポートの準備

1. ハブクラスター にログインします。以下のコマンドを実行します。

```
oc login
```

2. ハブクラスターで以下のコマンドを実行して namespace を作成します。注記: <cluster_name> で定義したクラスター名は、.yaml ファイルおよびコマンドでクラスターの namespace としても使用します。

```
oc new-project ${CLUSTER_NAME}
oc label namespace ${CLUSTER_NAME} cluster.open-cluster-
management.io/managedCluster=${CLUSTER_NAME}
```

3. 以下の YAML 例のように、ManagedCluster の例を編集します。

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: <cluster_name>
spec:
  hubAcceptsClient: true
```

4. ファイルは **managed-cluster.yaml** として保存します。

5. 以下のコマンドを使用して、YAML ファイルを適用します。

```
oc apply -f managed-cluster.yaml
```

6. klusterlet のアドオン設定ファイルを作成します。以下の YAML の例を入力します。

```
apiVersion: agent.open-cluster-management.io/v1
kind: KlusterletAddonConfig
metadata:
  name: <cluster_name>
  namespace: <cluster_name>
spec:
  clusterName: <cluster_name>
  clusterNamespace: <cluster_name>
  applicationManager:
    enabled: true
  certPolicyController:
    enabled: true
  clusterLabels:
    cloud: auto-detect
    vendor: auto-detect
  iamPolicyController:
    enabled: true
  policyController:
    enabled: true
  searchCollector:
    enabled: true
  version: 2.2.0
```

7. ファイルは **klusterlet-addon-config.yaml** として保存します。

8. YAML を適用します。以下のコマンドを実行します。

```
oc apply -f klusterlet-addon-config.yaml
```

注記: Red Hat OpenShift Dedicated クラスターをインポートし、vendor=OpenShiftDedicated のラベルを追加してベンダーが指定されないようにする場合、または vendor=auto-detect のラベルを追加する場合には managed-by=platform ラベルが

クラスターに自動的に追加されます。この追加ラベルを使用して、クラスターを Red Hat OpenShift Dedicated クラスターとして識別し、Red Hat OpenShift Dedicated クラスターをグループとして取得できます。

ManagedCluster-Import-Controller は **${CLUSTER_NAME}-import** という名前のシークレットを生成します。**${CLUSTER_NAME}-import** シークレットには、**import.yaml** が含まれており、このファイルをユーザーがマネージドクラスターに適用して klusterlet をインストールします。

8.2.4. klusterlet のインポート

重要: import コマンドには、インポートした各クラスターにコピーされるプルシークレット情報が含まれます。インポートしたクラスターにアクセスできるユーザーであれば誰でも、プルシークレット情報を表示することもできます。

1. マネージドクラスターのインポートコントローラーによって生成された **klusterlet-crd.yaml** を取得します。
   以下のコマンドを実行します。

   ```
   oc get secret ${CLUSTER_NAME}-import -n ${CLUSTER_NAME} -o jsonpath={.data.crds\\.yaml} | base64 --decode > klusterlet-crd.yaml
   ```

2. マネージドクラスターのインポートコントローラーによって生成された **import.yaml** を取得します。以下のコマンドを実行します。

   ```
   oc get secret ${CLUSTER_NAME}-import -n ${CLUSTER_NAME} -o jsonpath={.data.import\\.yaml} | base64 --decode > import.yaml
   ```

3. ターゲットの マネージド クラスターにログインします。

4. 手順 1 で生成した **klusterlet-crd.yaml** を適用します。以下のコマンドを実行します。

   ```
   kubectl apply -f klusterlet-crd.yaml
   ```

5. 手順 2 で生成した **import.yaml** ファイルを適用します。以下のコマンドを実行します。

   ```
   kubectl apply -f import.yaml
   ```

6. ターゲットのマネージドクラスターで Pod のステータスを検証します。以下のコマンドを実行します。

   ```
   kubectl get pod -n open-cluster-management-agent
   ```

7. インポートしたクラスターのステータス (**JOINED** および **AVAILABLE**) を確認します。ハブ クラスターから以下のコマンドを実行します。

   ```
   kubectl get managedcluster ${CLUSTER_NAME}
   ```

8. アドオンは、マネージドクラスターが **AVAILABLE** になってからインストールされます。ターゲットのマネージドクラスターでアドオンの Pod ステータスを確認します。以下のコマンドを実行します。

   ```
   kubectl get pod -n open-cluster-management-agent-addon
   ```

8.3. クラスターの KLUSTERLET アドオン設定の変更

ハブクラスターを使用して設定を変更するには、klusterlet addon の設定を変更します。

klusterlet addon コントローラーは、klusterletaddonconfigs.agent.open-cluster-management.io Kubernetes リソースの設定に合わせて有効化/無効化される機能を管理します。

以下の設定は、klusterletaddonconfigs.agent.open-cluster-management.io の Kubernetes リソースで更新できます。

設定名	値
applicationmanager	**true** または **false**
policyController	**true** または **false**
searchCollector	**true** または **false**
certPolicyController	**true** または **false**
iamPolicyController	**true** または **false**

8.3.1. ハブクラスターのコンソールを使用した変更

ハブクラスターを使用して、klusterletaddonconfigs.agent.open-cluster-management.io リソースの設定を変更できます。設定の変更には、以下の手順を実行します。

1. **ハブクラスターの Red Hat Advanced Cluster Management for Kubernetes コンソールへの認証を行います。**
2. **ハブクラスターのメインメニューから Search を選択します。**
3. **検索パラメーターに、kind:klusterletaddonconfigs の値を入力します。**
4. **更新するエンドポイントリソースを選択します。**
5. **spec セクションから、Edit を選択してコンテンツを編集します。**
6. **設定を変更します。**
7. **Save を選択して変更を適用します。**

8.3.2. ハブクラスターのコマンドラインを使用した変更

ハブクラスターを使用して設定を変更するには、<cluster-name> namespace へのアクセス権が必要です。以下の手順を実行します。

1. **ハブクラスターへの認証を行います。**
2. **以下のコマンドを入力してリソースを編集します。**

```
kubectl edit klusterletaddonconfigs.agent.open-cluster-management.io <cluster-name> -n <cluster-name>
```

-

3. **spec セクションを検索します。**
4. **必要に応じて設定を変更します。**

第9章 特定のクラスター管理ロールの設定

Red Hat Advanced Cluster Management for Kubernetes をインストールすると、デフォルト設定で Red Hat Advanced Cluster Management ハブクラスターに **cluster-admin** ロールが提供されます。このパーミッションを使用すると、ハブクラスターでマネージドクラスターを作成、管理、およびインポートできます。状況によっては、ハブクラスターのすべてのマネージドクラスターへのアクセスを提供するのではなく、ハブクラスターが管理する特定のマネージドクラスターへのアクセスを制限する必要がある場合があります。

クラスターロールを定義し、ユーザーまたはグループに適用することで、特定のマネージドクラスターへのアクセスを制限できます。ロールを設定して適用するには、以下の手順を実行します。

1. 以下の内容を含む YAML ファイルを作成してクラスターロールを定義します。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: <clusterrole-name>
rules:
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  resourceNames:
  - <managed-cluster-name>
  verbs:
  - get
  - list
  - watch
  - update
  - delete
  - deletecollection
  - patch
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  verbs:
  - create
- apiGroups:
  - ""
  resources:
  - namespaces
  resourceNames:
  - <managed-cluster-name>
  verbs:
  - create
  - get
  - list
  - watch
  - update
  - delete
  - deletecollection
  - patch
- apiGroups:
  - register.open-cluster-management.io
```

```
resources:
- managedclusters/accept
resourceNames:
- <managed-cluster-name>
verbs:
- update
```

clusterrole-name は、作成するクラスターロールの名前に置き換えます。

managed-cluster-name は、ユーザーにアクセスを許可するマネージドクラスターの名前に置き換えます。

2. 以下のコマンドを入力して **clusterrole** 定義を適用します。

```
oc apply <filename>
```

filename を、先の手順で作成した YAML ファイルの名前に置き換えます。

3. 以下のコマンドを入力して、**clusterrole** を指定されたユーザーまたはグループにバインドします。

```
oc adm policy add-cluster-role-to-user <clusterrole-name> <username>
```

clusterrole-name を、直前の手順で適用したクラスターロールの名前に置き換えます。**username** を、クラスターロールをバインドするユーザー名に置き換えます。

第10章 MANAGEDCLUSTERSETS

ManagedClusterSet は、マネージドクラスターのグループです。ManagedClusterSet では、グループ内の全マネージドクラスターに対するアクセス権を管理できます。ManagedClusterSetBinding リソースを作成して ManagedClusterSet リソースを namespace にバインドすることもできます。

10.1. MANAGEDCLUSTERSET の作成

ManagedClusterSet にマネージドクラスターをグループ化して、マネージドクラスターでのユーザーのアクセス権限を制限できます。

必要なアクセス権限: クラスターの管理者

ManagedClusterSet は、クラスタースコープのリソースであるため、ManagedClusterSet の作成先となるクラスターで管理者権限が必要です。マネージドクラスターは、複数の ManagedClusterSet に追加できません。ManagedClusterSet を作成するには以下の手順を実行します。

1. **yaml ファイルに以下の ManagedClusterSet 定義を追加します。**

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ManagedClusterSet
metadata:
  name: <clusterset1>
```

clusterset1 は ManagedClusterSet の名前に置き換えます。

10.2. クラスターの MANAGEDCLUSTERSET への追加

ManagedClusterSet の作成後に、1つ以上のマネージドクラスターを追加する必要があります。マネージドクラスターを追加するには以下の手順を実行します。

1. **managedclustersets/join の仮想サブリソースに 作成 できるように、RBAC ClusterRole エントリーが追加されていることを確認します。このパーミッションがない場合には、マネージドクラスターを ManagedClusterSet に割り当てることはできません。**
   このエントリーが存在しない場合は、yaml ファイルに追加します。サンプルエントリーは以下の内容のようになります。

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: clusterrole1
rules:
  - apiGroups: ["cluster.open-cluster-management.io"]
    resources: ["managedclustersets/join"]
    resourceNames: ["clusterset1"]
    verbs: ["create"]
```

   clusterset1 は ManagedClusterSet の名前に置き換えます。

   注記: マネージドクラスターを別の ManagedClusterSet に移動する場合には、いずれの ManagedClusterSets でもパーミッションの設定が必要です。

2. **yaml ファイルでマネージドクラスターの定義を検索します。ラベルの追加先のマネージドクラスター定義のセクションは、以下の内容のようになります。**

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
spec:
  hubAcceptsClient: true
```

この例では、cluster1 はマネージドクラスターの名前です。

3. ManagedClusterSet の名前を cluster.open-cluster-management.io/clusterset: clusterset1 形式で指定してラベルを追加します。
コードは以下の例のようになります。

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
  labels:
    cluster.open-cluster-management.io/clusterset: clusterset1
spec:
  hubAcceptsClient: true
```

この例では、cluster1 は、clusterset1 ManagedClusterSet に追加するクラスターです。

注記: マネージドクラスターが削除済みの ManagedClusterSet にこれまでに割り当てられていた場合には、すでに存在しないクラスターが指定された ManagedClusterSet がマネージドクラスターに設定されている可能性があります。その場合は、名前を新しい名前に置き換えます。

10.3. MANAGEDCLUSTERSET からのマネージドクラスターの削除

ManagedClusterSet からマネージドクラスターを削除して別の ManagedClusterSet に移動するか、セットの管理設定から削除する必要がある場合があります。

ManagedClusterSet からマネージドクラスターを削除するには、以下の手順を実行します。

1. 以下のコマンドを実行して、ManagedClusterSet でマネージドクラスターのリストを表示します。

```
kubectl get managedclusters -l cluster.open-cluster-management.io/clusterset=<clusterset1>
```

clusterset1 は ManagedClusterSet の名前を置き換えます。

2. 削除するクラスターのエントリーを見つけます。

3. 削除するクラスターの yaml エントリーからラベルを削除します。ラベルの例については、以下のコードを参照してください。

```
labels:
  cluster.open-cluster-management.io/clusterset: clusterset1
```

注記: マネージドクラスターを別の ManagedClusterSet に移動する場合には、いずれの ManagedClusterSets でも RBAC パーミッションの設定が必要です。

10.4. MANAGEDCLUSTERSETBINDING リソース

ManagedClusterSetBinding リソースを作成して ManagedClusterSet リソースを namespace にバインドします。同じ namespace で作成されたアプリケーションおよびポリシーがアクセスできるのは、バインドされた ManagedClusterSet リソースに含まれるマネージドクラスターだけです。

ManagedClusterSetBinding の作成時には、ManagedClusterSetBinding 名と、バインドする ManagedClusterSet 名を一致させる必要があります。

ManagedClusterSetBinding リソースは、以下の情報のようになります。

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ManagedClusterSetBinding
metadata:
  namespace: project1
  name: clusterset1
spec:
  clusterSet: clusterset1
```

ターゲットの ManagedClusterSet でのバインド権限を割り当てておく必要があります。以下の ClusterRole リソースの例を確認してください。この例には、ユーザーが **clusterset1** にバインドできるように、複数のルールが含まれています。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: clusterrole1
rules:
  - apiGroups: ["cluster.open-cluster-management.io"]
    resources: ["managedclustersets/bind"]
    resourceNames: ["clusterset1"]
    verbs: ["create"]
```

ロールアクションの詳細は、「ロールベースのアクセス制御」を参照してください。

第11章 マネージドクラスターの ANSIBLEJOB の作成

Red Hat Advanced Cluster Management for Kubernetes でクラスターを作成した場合、または Red Hat Advanced Cluster Management で管理するためにインポートした場合には、**AnsibleJob** を作成してクラスターにバインドできます。

以下の手順を実行して、Ansible ジョブを作成し、Red Hat Advanced Cluster Management でまだ管理されていないクラスターを使用して設定します。

1. アプリケーション機能でサポートされるチャネルのいずれかで、Ansible ジョブの定義ファイルを作成します。Git チャネルのみがサポートされます。
AnsibleJob は、定義の **kind** の値として使用します。

   定義ファイルの内容は以下の例のようになります。

   ```
   apiVersion: apiVersion: tower.ansible.com/v1alpha1
   kind: AnsibleJob
   metadata:
     name: hive-cluster-gitrepo
   spec:
     tower_auth_secret: my-toweraccess
     job_template_name: my-tower-template-name
     extra_vars:
       variable1: value1
       variable2: value2
   ```

   ファイルを prehook または posthook ディレクトリーに保存すると、配置ルールと一致するクラスター名の一覧が作成されます。クラスター名の一覧は、**extra_vars** の値として **AnsibleJob kind** リソースに渡すことができます。この値が **AnsibleJob** リソースに渡されると、Ansible ジョブで、新しいクラスター名が判別され、自動化での使用が可能になります。

2. Red Hat Advanced Cluster Management ハブクラスターにログインします。

3. Red Hat Advanced Cluster Management コンソールを使用して、先程作成した定義ファイルの保存先のチャネルを参照する Git サブスクリプションでアプリケーションを作成します。アプリケーションおよびサブスクリプションの作成に関する詳細は、「アプリケーションリソースの管理」を参照してください。
サブスクリプションの作成時に、ラベルを指定して、クラスターとこのサブスクリプションを連携させるために作成またはインポートするクラスターに追加できます。このラベルは、**vendor=OpenShift** などの既存のものでも、一意のラベルを作成、定義することも可能です。

   注記: すでに使用中のラベルを選択すると、Ansible ジョブは自動的に実行されます。prehook または posthook の一部ではないアプリケーションに、リソースを追加することを推奨します。

   デフォルトの配置ルールでは、**AnsibleJob** のラベルと一致するラベルの付いたクラスターが検出されると、ジョブが実行されます。ハブクラスターが管理する実行中のクラスターすべてで、自動化を実行するには、以下の内容を配置ルールに追加します。

   ```
   clusterConditions:
     - type: ManagedClusterConditionAvailable
       status: "True"
   ```

これを配置ルールの YAML コンテンツに貼り付けるか、Red Hat Advanced Cluster Management コンソールの Application create ページで Deploy to all online cluster and local cluster オプションを選択します。

4. 「クラスターの作成」または「ハブクラスターへのターゲットのマネージドクラスターのインポート」の手順にしたがって、クラスターを作成またはインポートします。
クラスターの作成またはインポート時には、サブスクリプションの作成時に使用したラベルと同じラベルを使用すると、**AnsibleJob** はクラスター上で自動的に実行されるように設定されます。

Red Hat Advanced Cluster Management は、クラスター名を **AnsibleJob.extra_vars.target_clusters** パスに自動的に挿入します。クラスター名を定義に動的に挿入できます。以下の手順を実行して、AnsibleJob を作成し、Red Hat Advanced Cluster Management で管理しているクラスターを使用して設定します。

1. Git チャネルの prehook または posthook ディレクトリーに、AnsibleJob の定義ファイルを作成します。
AnsibleJob は、定義の **kind** の値として使用します。

定義ファイルの内容は以下の例のようになります。

```
apiVersion: tower.ansible.com/v1alpha1
kind: AnsibleJob
metadata:
  name: hive-cluster-gitrepo
spec:
  tower_auth_secret: my-toweraccess
  job_template_name: my-tower-template-name
  extra_vars:
    variable1: value1
    variable2: value2
```

my-toweraccess は、Ansible Tower にアクセスするための認証シークレットに置き換えます。**my-tower-template-name** は、Ansible Tower のテンプレート名に置き換えます。

Ansible ジョブが制御するクラスターが削除されるか、追加されるたびに、AnsibleJob は自動的に **extra_vars.target_clusters** 変数を実行して更新します。この更新により、特定の自動化でクラスター名を指定したり、クラスターのグループに自動化を適用したりできるようになりました。

第12章 CLUSTERCLAIMS

ClusterClaim は、マネージドクラスター上のカスタムリソース定義 (CRD) です。ClusterClaim は、マネージドクラスターが要求する情報の一部を表します。以下の例は、YAML ファイルで特定される要求を示しています。

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ClusterClaim
metadata:
  name: id.openshift.io
spec:
  value: 95f91f25-d7a2-4fc3-9237-2ef633d8451c
```

以下の表は、Red Hat Advanced Cluster Management for Kubernetes が管理するクラスターにある、定義済みの ClusterClaim を示しています。

要求名	予約	変更可能	説明
id.k8s.io	true	false	アップストリームの提案で定義された ClusterID
kubeversion.open-cluster-management.io	true	true	Kubernetes バージョン
platform.open-cluster-management.io	true	false	AWS、GCE、Equinix Metal など、マネージドクラスターが稼働しているプラットフォーム
product.open-cluster-management.io	true	false	OpenShift、Anthos、EKS、および GKE などの製品名
id.openshift.io	false	false	OpenShift Container Platform クラスターでのみ利用できる OpenShift Container Platform 外部 ID
consoleurl.openshift.io	false	true	OpenShift Container Platform クラスターでのみ利用できる管理コンソールの URL
version.openshift.io	false	true	OpenShift Container Platform クラスターでのみ利用できる OpenShift Container Platform バージョン

マネージドクラスターで以前の要求が削除されるか、または更新されると、自動的に復元またはロールバックされます。

マネージドクラスターがハブに参加すると、マネージドクラスターで作成された ClusterClaim は、ハブの ManagedCluster リソースのステータスと同期されます。ClusterClaims のマネージドクラスターは、以下の例のようになります。

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  labels:
    cloud: Amazon
    clusterID: 95f91f25-d7a2-4fc3-9237-2ef633d8451c
    installer.name: multiclusterhub
    installer.namespace: open-cluster-management
    name: cluster1
    vendor: OpenShift
  name: cluster1
spec:
  hubAcceptsClient: true
  leaseDurationSeconds: 60
status:
  allocatable:
    cpu: '15'
    memory: 65257Mi
  capacity:
    cpu: '18'
    memory: 72001Mi
  clusterClaims:
    - name: id.k8s.io
      value: cluster1
    - name: kubeversion.open-cluster-management.io
      value: v1.18.3+6c42de8
    - name: platform.open-cluster-management.io
      value: AWS
    - name: product.open-cluster-management.io
      value: OpenShift
    - name: id.openshift.io
      value: 95f91f25-d7a2-4fc3-9237-2ef633d8451c
    - name: consoleurl.openshift.io
      value: 'https://console-openshift-console.apps.xxxx.dev04.red-chesterfield.com'
    - name: version.openshift.io
      value: '4.5'
  conditions:
    - lastTransitionTime: '2020-10-26T07:08:49Z'
      message: Accepted by hub cluster admin
      reason: HubClusterAdminAccepted
      status: 'True'
      type: HubAcceptedManagedCluster
    - lastTransitionTime: '2020-10-26T07:09:18Z'
      message: Managed cluster joined
      reason: ManagedClusterJoined
      status: 'True'
      type: ManagedClusterJoined
    - lastTransitionTime: '2020-10-30T07:20:20Z'
      message: Managed cluster is available
```

```
    reason: ManagedClusterAvailable
    status: 'True'
    type: ManagedClusterConditionAvailable
version:
  kubernetes: v1.18.3+6c42de8
```

12.1. 既存の CLUSTERCLAIM の表示

kubectl コマンドを使用して、マネージドクラスターに適用される ClusterClaim を一覧表示できます。これは、ClusterClaim をエラーメッセージと比較する場合に便利です。

注記: clusterclaims.cluster.open-cluster-management.io のリソースに list のパーミッションがあることを確認します。

以下のコマンドを実行して、マネージドクラスターにある既存の ClusterClaim の一覧を表示します。

```
kubectl get clusterclaims.cluster.open-cluster-management.io
```

12.2. カスタム CLUSTERCLAIMS の作成

ClusterClaim は、マネージドクラスターでカスタム名を使用して作成できるため、簡単に識別できます。カスタム ClusterClaim は、ハブクラスターの ManagedCluster リソースのステータスと同期されます。以下のコンテンツでは、カスタマイズされた ClusterClaim の定義例を示しています。

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ClusterClaim
metadata:
  name: <custom_claim_name>
spec:
  value: <custom_claim_value>
```

spec.value フィールドの最大長は 1024 です。ClusterClaim を作成するには clusterclaims.cluster.open-cluster-management.io リソースの create パーミッションが必要です。

第13章 SUBMARINER

submariner-addon コンポーネントはテクノロジープレビュー 機能です。

Submariner はオープンソースツールで、Red Hat Advanced Cluster Management for Kubernetes で使用すると、お使いの環境 (オンプレミスまたはクラウドのいずれか) 内の 2 つ以上の Kubernetes クラスター間の直接ネットワークを提供するために、Red Hat Advanced Cluster Management for Kubernetes で使用できます。Submariner の詳細は、Submariner を参照してください。

以下の環境でホストされる OpenShift Container Platform クラスターで Submariner を有効にできます。

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure
- IBM Cloud
- VMware vSphere
- ベアメタル
- Red Hat OpenShift Dedicated

Red Hat Advanced Cluster Management for Kubernetes は、ハブクラスターを使用してお使いの環境にデプロイできる Submariner コンポーネントを提供します。

13.1. 前提条件

Submariner を使用する前に、以下の前提条件があることを確認します。

- Red Hat OpenShift Container Platform バージョン 4.5 以降および Kubernetes バージョン 1.17 以降で実行している Red Hat Advanced Cluster Management ハブクラスター。
- **cluster administrator** のパーミッションを使用してハブクラスターにアクセスするための認証情報。
- (Kubernetes バージョン 1.17 以降を使用する) OpenShift Container Platform バージョン 4.4 以降で実行している 2 つ以上の OpenShift Container Platform マネージドクラスターは、Red Hat Advanced Cluster Management ハブクラスターによって管理されます。
- 重複しないクラスター間の Pod および Service Classless Inter-Domain Routing (CIDR)。
- ゲートウェイノード間で IP 接続を設定する必要があります。2 つのクラスターを接続する場合、1 つ以上のクラスターには、ゲートウェイノードに公開されているルーティング可能な IP アドレスが必要です。
- 各マネージドクラスターのすべてのノードでのファイアウォール設定は、両方の方向で 4800/UDP を許可する必要があります。
  注記: これは、クラスターが Amazon Web Services 環境にデプロイされる際に自動的に設定されますが、他の環境のクラスター用に手動で設定し、ベアメタルや VMware vSphere などのプライベートクラウドを保護するファイアウォール用に設定する必要があります。

- ゲートウェイノードのファイアウォール設定では、入力 8080/TCP が許可されているため、クラスター内の他のノードがアクセスできます。
- UDP/4500、UDP/500、およびゲートウェイノード上の IPsec トラフィックに使用されるその他のポート用にファイアウォール設定が開いています。

前提条件の詳細は、Submariner の前提条件 を参照してください。

13.2. SUBMARINER をデプロイするホストの準備

Red Hat Advanced Cluster Management for Kubernetes に Submariner をデプロイする前に、接続用にホスト環境でクラスターを準備する必要があります。要件はホスティング環境によって異なるため、ホスティング環境の手順に従います。

13.2.1. Amazon Web Services で Submariner をデプロイする準備

Amazon Web Services でホストされる OpenShift Container Platform クラスターを設定して、Submariner デプロイメントと統合する方法は 2 つあります。以下のいずれかのオプションを選択して、接続を準備します。

- 方法 1
  SubmarinerConfig API を使用してクラスター環境をビルドできます。この方法では、submariner-addon が環境を設定するため、SubmarinerConfig の定義で設定およびクラウドプロバイダーの認証情報を使用します。

  注記: この方法は、クラスターが Amazon Web Services にある場合に限りサポートされます。

  以下の内容を含む YAML ファイルを作成します。

```
apiVersion: v1
kind: Secret
metadata:
  name: <cloud-provider-credential-secret-name>
  namespace: <managed-cluster-namespace>
type: Opaque
data:
  aws_access_key_id: <aws-access-key-id>
  aws_secret_access_key: <aws-secret-access-key>
```

  name の形式は、Red Hat Advanced Cluster Management でのクラスターのプロビジョニングに使用した認証情報シークレット名と同じです。

  Submariner クラスター環境を手動で設定している場合は、SubmarinerConfig の追加設定にこの設定を追加します。この例では、IPSecIKEPort が 501 に設定され、IPSecNATTPort が 4501 に設定されます。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: <config-name>
  namespace: <managed-cluster-namespace>
spec:
  IPSecIKEPort: 501
  IPSecNATTPort: 4501
  ...
```

- 方法 2
  Submariner の Web サイトにあるスクリプトファイル **prep_for_subm.sh** を使用して、Submariner デプロイメント用に OpenShift Container Platform インストーラーでプロビジョニングされる Amazon Web Services インフラストラクチャーを更新できます。スクリプトおよびスクリプトの実行方法については、「Prepare AWS Clusters for Submariner」を参照してください。

13.2.2. Google Cloud Platform で Submariner をデプロイする準備

Google Cloud Platform で Submariner コンポーネントをデプロイするようにクラスターを準備するには、以下の手順を実行します。

1. Google Cloud Platform で受信および送信のファイアウォールルールを作成し、IPsec IKE (デフォルトでは 500/UDP) および NAT トラバーサルポート (デフォルトでは 4500/UDP) を開き、Submariner 通信を有効にします。

   ```
   $ gcloud compute firewall-rules create <name> --network=<network-name> --allow=udp:
   <ipsec-port> --direction=IN
   $ gcloud compute firewall-rules create <rule-name> --network=<network-name> --allow=udp:
   <ipsec-port>  --direction=OUT
   ```

   rule-name を、実際のルール名に置き換えます。

   network-name を、Google Cloud Platform クラスターネットワーク名に置き換えます。

   ipsec-port を、実際の IPsec ポートに置き換えます。

2. Google Cloud Platform で受信および送信のファイアウォールルールを作成し、4800/UDP ポートを開き、ワーカーノードおよびマスターノードから Submariner Gateway ノードに Pod トラフィックをカプセル化します。

   ```
   $ gcloud compute firewall-rules create <name> --network=<network-name> --
   allow=udp:4800 --direction=IN
   $ gcloud compute firewall-rules create <name> --network=<network-name> --
   allow=udp:4800 --direction=OUT
   ```

   name を、実際のルール名に置き換えます。

   network-name を、Google Cloud Platform クラスターネットワーク名に置き換えます。

3. Google Cloud Platform で受信および送信のファイアウォールルールを作成し、8080/TCP ポートを開き、Submariner Gateway ノードからメトリクスサービスをエクスポートします。

   ```
   $ gcloud compute firewall-rules create <name> --network=<network-name> --allow=tcp:8080
   --direction=IN
   $ gcloud compute firewall-rules create <name> --network=<network-name> --allow=tcp:8080
   --direction=OUT
   ```

   name を、実際のルール名に置き換えます。

   network-name を、Google Cloud Platform クラスターネットワーク名に置き換えます。

13.2.3. Microsoft Azure で Submariner をデプロイする準備

Submariner コンポーネントをデプロイするように Microsoft Azure でクラスターを準備するには、以下の手順を実行します。

1. Microsoft Azure 環境で受信および送信のファイアウォールルールを作成し、IP セキュリティー IKE (デフォルトでは 500/UDP) および NAT トラバーサルポート (デフォルトでは 4500/UDP) を開き、Submariner 通信を有効にします。

```
# create inbound nat rule
$ az network lb inbound-nat-rule create --lb-name <lb-name> \
--resource-group <res-group> \
--name <name> \
--protocol Udp --frontend-port <ipsec-port> \
--backend-port <ipsec-port> \
--frontend-ip-name <frontend-ip-name>

# add your vm network interface to the created inbound nat rule
$ az network nic ip-config inbound-nat-rule add \
--lb-name <lb-name> --resource-group <res-group> \
--inbound-nat-rule <nat-name> \
--nic-name <nic-name> --ip-config-name <pipConfig>
```

lb-name を、ロードバランサー名に置き換えます。

res-group を、リソースグループ名に置き換えます。

nat-name を、ロードバランシングの受信 NAT ルール名に置き換えます。

ipsec-port を、実際の IPsec ポートに置き換えます。

pipConfig を、クラスターのフロントエンド IP 設定名に置き換えます。

nic-name を、実際のネットワークインターフェースカード (NIC) 名に置き換えます。

2. Submariner ゲートウェイメトリクス要求を転送する負荷分散の受信 NAT ルールを 1 つ作成します。

```
# create inbound nat rule
$ az network lb inbound-nat-rule create --lb-name <lb-name> \
--resource-group <res-group> \
--name <name> \
--protocol Tcp --frontend-port 8080 --backend-port 8080 \
--frontend-ip-name <frontend-ip-name>

# add your vm network interface to the created inbound nat rule
$ az network nic ip-config inbound-nat-rule add \
--lb-name <lb-name> --resource-group <res-group> \
--inbound-nat-rule <nat-name> \
--nic-name <nic-name> --ip-config-name <pipConfig>
```

lb-name を、ロードバランサー名に置き換えます。

res-group を、リソースグループ名に置き換えます。

nat-name を、ロードバランシングの受信 NAT ルール名に置き換えます。

pipConfig を、クラスターのフロントエンド IP 設定名に置き換えます。

nic-name を、実際のネットワークインターフェースカード (NIC) 名に置き換えます。

3. Azure でネットワークセキュリティーグループ {NSG) セキュリティールールを作成し、Submariner の IPsec IKE (デフォルトでは 500/UDP) および NAT トラバーサルポート (デフォルトでは 4500/UDP) を開きます。

```
$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Inbound --access Allow \
--protocol Udp --destination-port-ranges <ipsec-port>

$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Outbound --access Allow \
--protocol Udp --destination-port-ranges <ipsec-port>
```

res-group を、リソースグループ名に置き換えます。

nsg-name を、実際の NSG 名に置き換えます。

priority を、ルールの優先度に置き換えます。

name を、実際のルール名に置き換えます。

ipsec-port を、実際の IPsec ポートに置き換えます。

4. NSG ルールを作成し、4800/UDP ポートを開き、ワーカーノードおよびマスターノードから Submariner Gateway ノードに Pod トラフィックをカプセル化します。

```
$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Inbound --access Allow \
--protocol Udp --destination-port-ranges 4800 \

$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Outbound --access Allow \
--protocol Udp --destination-port-ranges 4800
```

res-group を、リソースグループ名に置き換えます。

nsg-name を、実際の NSG 名に置き換えます。

priority を、ルールの優先度に置き換えます。

name を、実際のルール名に置き換えます。

5. NSG ルールを作成して 8080/TCP ポートを開き、Submariner ゲートウェイノードからメトリクスサービスをエクスポートします。

```
$ az network nsg rule create --resource-group <res-group> \
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Inbound --access Allow \
--protocol Tcp --destination-port-ranges 8080 \

$ az network nsg rule create --resource-group <res-group> \
```

```
--nsg-name <nsg-name> --priority <priority> \
--name <name> --direction Outbound --access Allow \
--protocol Udp --destination-port-ranges 8080
```

res-group を、リソースグループ名に置き換えます。

nsg-name を、実際の NSG 名に置き換えます。

priority を、ルールの優先度に置き換えます。

name を、実際のルール名に置き換えます。

13.2.4. IBM Cloud で Submariner をデプロイする準備

IBM Cloud には、従来のクラスターと、仮想プライベートクラウド (VPC) での 2 世代のコンピュートインフラストラクチャー (VPC) の 2 種類の Red Hat OpenShift Kubernetes Service (ROKS) があります。従来のクラスターの IPsec ポートを設定できないため、従来の ROKS クラスターでは Submariner を実行できません。

VPC で、Submariner を使用するように ROKS クラスターを設定するには、以下のリンクの手順を実行します。

1. クラスターを作成する前に、Pod およびサービスのサブネットを指定します。これにより、他のクラスターと CIDR が重複しないようにします。既存のクラスターを使用している場合は、クラスター間で Pod およびサービス CIDR が重複していないことを確認します。詳細は、「VPC サブネット」を参照してください。
2. パブリックゲートウェイを、クラスターで使用されるサブネットに割り当てます。この手順は、「パブリック・ゲートウェイ」を参照してください。
3. セキュリティーグループ の手順を実行して、クラスターのデフォルトのセキュリティーグループに受信ルールを作成します。ファイアウォールが、ゲートウェイノードの UDP/4500 ポートおよび UDP/500 ポートでの受信トラフィックおよび送信トラフィックを許可し、他のすべてのノードについては受信および送信の UDP/4800 を許可するようにしてください。
4. クラスター内で、パブリックゲートウェイを持つノードに **submariner.io/gateway=true** とラベルを付けます。
5. クラスターに IPPools を作成して Calico CNI を設定するには、「Calico」を参照してください。

13.2.5. Red Hat OpenShift Dedicated で Submariner をデプロイする準備

Red Hat OpenShift Dedicated は、AWS および Google Cloud Platform によってプロビジョニングされたクラスターをサポートします。

13.2.5.1. Red Hat OpenShift Dedicated で AWS に Submariner をデプロイする準備

Red Hat OpenShift Dedicated に AWS クラスターを設定するには、以下の手順を実行します。

1. Red Hat OpenShift Hosted SRE サポートチーム対して サポートチケット を作成し、**cluster-admin** グループに Red Hat OpenShift Dedicated クラスターへのアクセスを許可します。**dedicated-admin** のデフォルトアクセスには、**MachineSet** の作成に必要なパーミッションがありません。

2. グループが作成されたら、Red Hat OpenShift Dedicated ドキュメントの「Granting the cluster-admin role to users」の手順を実行して作成した **cluster-admin** グループにユーザー名を追加します。

3. 「Amazon Web Services で Submariner をデプロイする準備」に記載されている前提条件を完了します。

4. ユーザーの **osdCcsAdmin** の認証情報を設定し、それをサービスアカウントとして使用することができます。

13.2.5.2. Red Hat OpenShift Dedicated で Google Cloud Platform に Submariner をデプロイする準備

Red Hat OpenShift Dedicated で Google Cloud Platform クラスターを設定するには、以下の手順を実行します。

1. 「Google Cloud Platform で Submariner をデプロイする準備」の前提条件を完了してください。

2. デプロイメントの管理に使用できる **osd-ccs-admin** という名前のサービスアカウントを設定します。

13.2.6. VMware vSphere またはベアメタルで Submariner をデプロイする準備

Submariner をデプロイするように VMware vSphere およびベアメタルクラスターを準備するには、以下の手順を実行します。

1. 1 つ以上のクラスター上のゲートウェイノードに指定される公開ルーティング可能な IP アドレスを設定します。

2. IP セキュリティーのポートが開いていることを確認します。デフォルトのポートは 4500/UDP および 500/UDP です。ファイアウォールによりデフォルトのポートがブロックされる場合は、4501/UDP や 501/UDP などの利用可能なカスタムポートのペアを設定します。

13.3. SUBMARINER のデプロイ

submariner-addon コンポーネントはテクノロジープレビュー 機能です。

Submariner をデプロイするには、以下の手順を実行します。

1. クラスター管理者のパーミッションでハブクラスターにログインします。

2. 適用可能な準備が完了していることを確認します。要件については、「Submariner」を参照してください。

3. ManagedClusterSets の手順に従って、ハブクラスターに **ManagedClusterSet** を作成します。**submariner-addon** は、**submariner-clusterset-<clusterset-name>-broker** という名前の namespace を作成し、Submariner Broker をこれにデプロイします。namespace 名の <clusterset-name> は、ManagedClusterSet の名前になります。**ManagedClusterSet** のエントリーは以下のような内容になります。

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ManagedClusterSet
metadata:
  name: <ManagedClusterSet-name>
```

ManagedClusterSet-name は、作成する **ManagedClusterSet** の名前に置き換えます。

4. 以下のコマンドを入力して、マネージドクラスター間の通信を提供する Submariner を有効にします。

```
oc label managedclusters <managedcluster-name> "cluster.open-cluster-management.io/submariner-agent=true" --overwrite
```

managedcluster-name は、Submariner で使用するマネージドクラスターの名前に置き換えます。

5. 以下のコマンドを入力して、マネージドクラスターを **ManagedClusterSet** に追加します。

```
oc label managedclusters <managedcluster-name> "cluster.open-cluster-management.io/clusterset=<ManagedClusterSet-name>" --overwrite
```

managedcluster-name は、**ManagedClusterSet** に追加するマネージドクラスターの名前に置き換えます。**ManagedClusterSet-name** は、マネージドクラスターを追加する **ManagedClusterSet** の名前に置き換えます。

6. **ManagedClusterSet** に追加するすべてのマネージドクラスターでこの手順を繰り返します。

13.3.1. Submariner の AWS OpenShift Container Platform クラスターへのデプロイ

Red Hat Advanced Cluster Management for Kubernetes バージョン 2.2 は、Amazon Web Services にデプロイされた管理対象の OpenShift Container Platform クラスターへの Submariner の自動デプロイメントをサポートします。

Submariner をデプロイするには、以下の手順を実行します。

1. クラスター管理者のパーミッションでハブクラスターにログインします。
   - Red Hat Advanced Cluster Management で作成したクラスターの場合、手順 2 に進みます。
   - Red Hat Advanced Cluster Management にインポートされたクラスターの場合、ステップ 3 に進みます。
2. Red Hat Advanced Cluster Management for Kubernetes で作成された AWS クラスターの場合は、以下の **SubmarinerConfig** コンテンツで **submarinerconfig.yaml** という名前のファイルを作成します。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <your-cluster-namespace>
spec:
  credentialsSecret:
    name: <your-cluster-name>-aws-creds
  subscriptionConfig:
    channel: alpha
    startingCSV: submariner.v0.8.1
```

   - **your-cluster-namespace** はクラスターの namespace に置き換えます。

- **your-cluster-name は、クラスター名に置き換えます。aws-cluster-namespace aws-cloud-credentials のファイルは Hive クラスター namespace に保存されます。**
  ステップ 4 に進みます。

3. **OpenShift Container Platform で作成され、Red Hat Advanced Cluster Management にインポートされている AWS クラスターの場合は、aws-cloud-credentials を作成します。**

   a. **以下のシークレットをマネージドクラスターの namespace に適用します。**

```
apiVersion: v1
kind: Secret
metadata:
  name: aws-cloud-credentials
  namespace: <your-cluster-namespace>
type: Opaque
data:
  aws_access_key_id: <your-aws_access_key_id>
  aws_secret_access_key: <your-aws_secret_access_key>
```

   b. **以下の SubmarinerConfig コンテンツで submarinerconfig.yaml という名前のファイルを作成します。**

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: subconfig
  namespace: <your-cluster-namespace>
spec:
  credentialsSecret:
    name: <aws-cloud-credentials>
  subscriptionConfig:
    channel: alpha
    startingCSV: submariner.v0.8.1
```

   aws-cloud-credential を、直前の手順で作成した AWS 認証情報に置き換えます。

4. **以下のコマンドを入力して、ハブクラスターのマネージドクラスターに SubmarinerConfig リソースを適用します。**

```
oc apply -f submarinerconfig.yaml
```

5. **ハブクラスターに ManagedClusterSet を作成します。**

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ManagedClusterSet
metadata:
  name: my-clusterset
```

6. **以下のコマンドを入力して、マネージドクラスターを ManagedClusterSet に追加します。**

```
oc label managedclusters <managedcluster-name> "cluster.open-cluster-management.io/clusterset=<ManagedClusterSet-name>" --overwrite
```

7. 以下のコマンドを実行して、Red Hat Advanced Cluster Management が Submariner:app-name を自動的にデプロイできるようにします。

```
oc label managedclusters <managedcluster-name> "cluster.open-cluster-management.io/submariner-agent=true" --overwrite
```

13.4. SUBMARINER のサービス検出の有効化

submariner-addon コンポーネントは テクノロジープレビュー 機能です。

Submariner がマネージドクラスターと同じ環境にデプロイされると、**ManagedClusterSet** のクラスター全体の Pod とサービスとの間でセキュアな IP ルーティング用にルートが設定されます。**ManagedClusterSet** の他のクラスターにサービスを表示し、検出できるようにするには、**ServiceExport** オブジェクトを作成する必要があります。**ServiceExport** オブジェクトでサービスをエクスポートすると、**<service>.<namespace>.svc.clusterset.local** 形式でサービスにアクセスできます。複数のクラスターが同じ名前で、同じ namespace からサービスをエクスポートすると、他のクラスターは、その複数のクラスターを 1 つの論理サービスとして認識されます。

この例では、**default** の namespace で**nginx** サービスを使用しますが、Kubernetes の **ClusterIP** サービスまたはヘッドレスサービスを検出できます。

1. 以下のコマンドを入力して、**ManagedClusterSet** のマネージドクラスターに **nginx** サービスのインスタンスを適用します。

```
oc -n default create deployment nginx --image=nginxinc/nginx-unprivileged:stable-alpine
oc -n default expose deployment nginx --port=8080
```

2. YAML ファイルに以下の内容の **ServiceExport** エントリーを作成して、サービスをエクスポートします。

```
apiVersion: multicluster.x-k8s.io/v1alpha1
kind: ServiceExport
metadata:
  name: <service-name>
  namespace: <service-namespace>
```

service-name を、エクスポートするサービスの名前に置き換えます。この例では、**nginx** になります。**service-namespace** を、サービスが置かれた namespace の名前に置き換えます。この例では、**default** になります。

3. 別のマネージドクラスターから以下のコマンドを実行して、**nginx** サービスにアクセスできることを確認します。

```
oc -n default run --generator=run-pod/v1 tmp-shell --rm -i --tty --image quay.io/submariner/nettest -- /bin/bash curl nginx.default.svc.clusterset.local:8080
```

これで、**nginx** サービス検出が Submariner に対して設定されました。

第14章 クラスターのアップグレード

Red Hat Advanced Cluster Management for Kubernetes で管理する Red Hat OpenShift Container Platform クラスターを作成したら、Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、マネージドクラスターが使用するバージョンチャネルで利用可能な最新のマイナーバージョンに、これらのクラスターをアップグレードできます。

オンライン環境では、Red Hat Advanced Cluster Management コンソールでアップグレードが必要なクラスターごとに送信される通知を使用して、更新が自動識別されます。

オフライン環境のクラスターをアップグレードするプロセスでは、必要なリリースイメージを設定してミラーリングする手順が追加で必要になります。この手順では、Red Hat OpenShift Update Service の Operator を使用してアップグレードを特定します。オフライン環境を使用している場合の必要な手順については、「非接続クラスターのアップグレード」を参照してください。

注記: メジャーバージョンへのアップグレードには、そのバージョンへのアップグレードの前提条件をすべて満たしていることを確認する必要があります。コンソールでクラスターをアップグレードする前に、マネージドクラスターのバージョンチャネルを更新する必要があります。マネージドクラスターのバージョンチャネルを更新後に、Red Hat Advanced Cluster Management for Kubernetes コンソールに、アップグレードに利用可能な最新版が表示されます。

重要: Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、Red Hat OpenShift Dedicated で Red Hat OpenShift Kubernetes Service マネージドクラスターまたは OpenShift Container Platform マネージドクラスターをアップグレードできません。

このアップグレードの手法は、ステータスが Ready の OpenShift Container Platform のマネージドクラスタークラスターでだけ使用できます。

オンライン環境でクラスターをアップグレードするには、以下の手順を実行します。

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。アップグレードが利用可能な場合には、Distribution version の列に表示されます。
2. アップグレードするクラスターを選択します。注記: クラスターは Ready の状態であること、かつ、コンソールを使用して Red Hat OpenShift Container Platform クラスターをアップグレードできる必要があります。
3. Upgrade を選択します。
4. 各クラスターの新しいバージョンを選択します。
5. Upgrade を選択します。

クラスターのアップグレードに失敗すると、Operator は通常アップグレードを数回再試行し、停止し、コンポーネントに問題があるステータスを報告します。場合によっては、アップグレードプロセスは、プロセスの完了を繰り返し試行します。アップグレードに失敗した後にクラスターを以前のバージョンにロールバックすることはサポートされていません。クラスターのアップグレードに失敗した場合は、Red Hat サポートにお問い合わせください。

14.1. 非接続クラスターのアップグレード

Red Hat OpenShift Update Service を Red Hat Advanced Cluster Management for Kubernetes で使用すると、非接続環境でクラスターをアップグレードできます。

セキュリティー上の理由で、クラスターがインターネットに直接接続できない場合があります。このような場合は、アップグレードが利用可能なタイミングや、これらのアップグレードの処理方法を把握するのが困難になります。OpenShift Update Service を設定すると便利です。

OpenShift Update Service は、個別の Operator および オペランドで、非接続環境で利用可能なマネージドクラスターを監視して、クラスターのアップグレードで利用できるようにします。OpenShift Update Service の設定後に、以下のアクションを実行できます。

1. オフラインのクラスター向けにいつアップグレードが利用できるかを監視します。
2. グラフデータファイルを使用してアップグレード用にどの更新がローカルサイトにミラーリングされているかを特定します。
3. Red Hat Advanced Cluster Management コンソールを使用して、クラスターのアップグレードが利用可能であることを通知します。

14.1.1. 前提条件

OpenShift Update Service を使用して非接続クラスターをアップグレードするには、以下の前提条件を満たす必要があります。

- Red Hat OpenShift Container Platform 4.5 以降に、制限付きの OLM を設定して Red Hat Advanced Cluster Management ハブクラスターをデプロイしておく。制限付きの OLM の設定方法については、「ネットワークが制限された環境での Operator Lifecycle Manager の使用」を参照してください。
  ヒント: 制限付きの OLM の設定時に、カタログソースイメージをメモします。
- Red Hat Advanced Cluster Management ハブクラスターが管理する OpenShift Container Platform クラスター。
- クラスターイメージをミラーリング可能なローカルレジストリーにアクセスするための認証情報。このリポジトリーの作成方法については、「ネットワークが制限された環境でのインストール用のミラーレジストリーの作成」を参照してください。
  注記: アップグレードするクラスターの現行バージョンのイメージは、ミラーリングされたイメージの 1 つとして常に利用可能でなければなりません。アップグレードに失敗すると、クラスターはアップグレード試行時のクラスターのバージョンに戻ります。

14.1.2. 非接続ミラーレジストリーの準備

ローカルのミラーリングレジストリーに、アップグレード前の現行のイメージと、アップグレード後のイメージの療法をミラーリングする必要があります。イメージをミラーリングするには以下の手順を実行します。

1. 以下の例のような内容を含むスクリプトファイルを作成します。

```
UPSTREAM_REGISTRY=quay.io
PRODUCT_REPO=openshift-release-dev
RELEASE_NAME=ocp-release
OCP_RELEASE=4.5.2-x86_64
LOCAL_REGISTRY=$(hostname):5000
LOCAL_SECRET_JSON=/path/to/pull/secret

oc adm -a ${LOCAL_SECRET_JSON} release mirror \
--
from=${UPSTREAM_REGISTRY}/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELE
```

```
ASE} \
--to=${LOCAL_REGISTRY}/ocp4 \
--to-release-image=${LOCAL_REGISTRY}/ocp4/release:${OCP_RELEASE}
```

path-to-pull-secret は、OpenShift Container Platform のプルシークレットへのパスに置き換えます。

2. スクリプトを実行して、イメージのミラーリング、設定の構成、リリースイメージとリリースコンテンツの分離を行います。

ヒント: **ImageContentSourcePolicy** の作成時に、このスクリプトの最後の行にある出力を使用できます。

14.1.3. OpenShift Update Service の Operator のデプロイ

OpenShift Container Platform 環境で OpenShift Update Service の Operator をデプロイするには、以下の手順を実行します。

1. ハブクラスターで、OpenShift Container Platform Operator のハブにアクセスします。
2. **Red Hat OpenShift Update Service Operator** を選択して Operator をデプロイします。必要に応じてデフォルト値を更新します。Operator をデプロイすると、**openshift-cincinnati** という名前の新規プロジェクトが作成されます。
3. Operator のインストールが完了するまで待ちます。
   ヒント: OpenShift Container Platform コマンドラインで**oc get pods** コマンドを入力して、インストールのステータスを確認できます。Operator の状態が **running** であることを確認します。

14.1.4. グラフデータの init コンテナーの構築

OpenShift Update Service はグラフデータ情報を使用して、利用可能なアップグレードを判別します。オンライン環境では、OpenShift Update Service は Cincinnati グラフデータの GitHub リポジトリー から直接利用可能なアップグレードがないか、グラフデータ情報をプルします。非接続環境を設定しているので、**init container** を使用してローカルリポジトリーでグラフデータを利用できるようにする必要があります。以下の手順を実行して、グラフデータの **init container** を作成します。

1. 以下のコマンドを入力して、グラフデータ Git リポジトリーのクローンを作成します。

```
git clone https://github.com/openshift/cincinnati-graph-data
```

2. グラフデータの **init** の情報が含まれるファイルを作成します。このサンプル Dockerfile は、**cincinnati-operator** GitHub リポジトリーにあります。ファイルの内容は以下の例のようになります。

```
FROM registry.access.redhat.com/ubi8/ubi:8.1

RUN curl -L -o cincinnati-graph-data.tar.gz https://github.com/openshift/cincinnati-graph-data/archive/master.tar.gz

RUN mkdir -p /var/lib/cincinnati/graph-data/

CMD exec /bin/bash -c "tar xvzf cincinnati-graph-data.tar.gz -C /var/lib/cincinnati/graph-data/ --strip-components=1"
```

この例では、以下のように設定されています。

- **FROM** 値は、OpenShift Update Service がイメージを検索する先の外部レジストリーに置き換えます。
- **RUN** コマンドはディレクトリーを作成し、アップグレードファイルをパッケージ化します。
- **CMD** コマンドは、パッケージファイルをローカルリポジトリーにコピーして、ファイルを展開してアップグレードします。

3. 以下のコマンドを実行して、グラフデータの **init container** をビルドします。

```
podman build -f <path_to_Dockerfile> -t
${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-container:latest
podman push ${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-
container:latest --authfile=/path/to/pull_secret.json
```

path_to_Dockerfile は、直前の手順で作成したファイルへのパスに置き換えます。

${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-container は、ローカルグラフデータ init container へのパスに置き換えます。

/path/to/pull_secret は、プルシークレットへのパスに置き換えます。

注記: **podman** がインストールされていない場合には、コマンドの **podman** を **docker** に置き換えることもできます。

14.1.5. ミラーリングされたレジストリーの証明書の設定

セキュアな外部コンテナーレジストリーを使用してミラーリングされた OpenShift Container Platform リリースイメージを保存する場合には、アップグレードグラフをビルドするために OpenShift Update Service からこのレジストリーへのアクセス権が必要です。OpenShift Update Service Pod と連携するように CA 証明書を設定するには、以下の手順を実行します。

1. **image.config.openshift.io** にある OpenShift Container Platform 外部レジストリー API を検索します。これは、外部レジストリーの CA 証明書の保存先です。
詳細は、OpenShift Container Platform ドキュメントの「イメージレジストリーアクセス用の追加のトラストストアの設定」を参照してください。
2. **openshift-config** namespace に ConfigMap を作成します。
3. CA 証明書をキーの **cincinnati-registry** に追加します。OpenShift Update Service はこの設定を使用して、証明書を特定します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: trusted-ca
data:
  cincinnati-registry: |
    -----BEGIN CERTIFICATE-----
    ...
    -----END CERTIFICATE-----
```

4. **image.config.openshift.io** API の **cluster** リソースを編集して、**additionalTrustedCA** フィールドを作成した ConfigMap 名に設定します。

```
oc patch image.config.openshift.io cluster -p '{"spec":{"additionalTrustedCA":
{"name":"trusted-ca"}}}' --type merge
```

trusted-ca は、新しい ConfigMap へのパスに置き換えます。

OpenShift Update Service Operator は、変更がないか、**image.config.openshift.io** API と、**openshift-config** namespace に作成した ConfigMap を監視し、CA 証明書が変更された場合にはデプロイメントを再起動します。

14.1.6. OpenShift Update Service インスタンスのデプロイ

ハブクラスターへの OpenShift Update Service インスタンスのデプロイが完了したら、このインスタンスは、クラスターのアップグレードのイメージをミラーリングして非接続マネージドクラスターに提供する場所に配置されます。インスタンスをデプロイするには、以下の手順を実行します。

1. デフォルトの Operator の namespace (**openshift-cincinnati**) を使用しない場合には、お使いの OpenShift Update Service インスタンスの namespace を作成します。
   a. OpenShift Container Platform ハブクラスターコンソールのナビゲーションメニューで、Administration >Namespaces を選択します。
   b. Create Namespaceを選択します。
   c. namespace 名と、namespace のその他の情報を追加します。
   d. Create を選択して namespace を作成します。
2. OpenShift Container Platform コンソールの Installed Operators セクションで、Red Hat OpenShift Update Service Operator を選択します。
3. メニューから Create Instance を選択します。
4. OpenShift Update Service インスタンスからコンテンツを貼り付けます。YAML ファイルは以下のマニフェストのようになります。

```
apiVersion: cincinnati.openshift.io/v1beta1
kind: Cincinnati
metadata:
  name: openshift-update-service-instance
  namespace: openshift-cincinnati
spec:
  registry: <registry_host_name>:<port>
  replicas: 1
  repository: ${LOCAL_REGISTRY}/ocp4/release
  graphDataImage: '<host_name>:<port>/cincinnati-graph-data-container'
```

spec.registry の値は、イメージの非接続環境にあるローカルレジストリーへのパスに置き換えます。

spec.graphDataImage の値は、グラフデータ init container へのパスに置き換えます。ヒント: これは、**podman push** コマンドを使用して、グラフデータ init container をプッシュする時に使用した値と同じです。

5. Create を選択してインスタンスを作成します。

6. ハブクラスター CLI で **oc get pods** コマンドを入力し、インスタンス作成のステータスを表示します。時間がかかる場合がありますが、コマンド結果でインスタンスと Operator が実行中である旨が表示されたらプロセスは完了です。

14.1.7. デフォルトレジストリーを上書きするためのポリシーのデプロイ (任意)

注記: 本セクションの手順は、ミラーレジストリーにリリースをミラーリングした場合にのみ該当します。

OpenShift Container Platform にはイメージレジストリーのデフォルト値があり、この値でアップグレードパッケージの検索先を指定します。非接続環境では、リリースイメージをミラーリングするローカルイメージレジストリーへのパスに値を置き換えるポリシーを作成してください。

これらの手順では、ポリシーの名前に ImageContentSourcePolicy を指定します。ポリシーを作成するには、以下の手順を実行します。

1. **ハブクラスターの OpenShift Container Platform 環境にログインします。**

2. **OpenShift Container Platform ナビゲーションから Administration >Custom Resource Definitions を選択します。**

3. **Instances タブを選択します。**

4. **コンテンツが表示されるように非接続 OLM を設定する時に作成した ImageContentSourcePolicy の名前を選択します。**

5. **YAML タブを選択して、YAML 形式でコンテンツを表示します。**

6. **ImageContentSourcePolicy の内容全体をコピーします。**

7. **Red Hat Advanced Cluster Management コンソールで、 Govern risk >Create policy を選択します。**

8. **YAML スイッチを On に設定して、ポリシーの YAML バージョンを表示します。**

9. **YAML コードのコンテンツをすべて削除します。**

10. **以下の YAML コンテンツをウィンドウに貼り付け、カスタムポリシーを作成します。**

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
```

```
        name: policy-pod-sample-nginx-pod
      spec:
        object-templates:
          - complianceType: musthave
            objectDefinition:
              apiVersion: v1
              kind: Pod
              metadata:
                name: sample-nginx-pod
                namespace: default
              status:
                phase: Running
        remediationAction: inform
        severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      []  # selects all clusters if not specified
```

11. **テンプレートの objectDefinition セクション内のコンテンツは、ImageContentSourcePolicy の設定を追加する以下の内容に置き換えます。**

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: ImageContentSourcePolicy
spec:
  repositoryDigestMirrors:
  - mirrors:
    - <path-to-local-mirror>
    source: registry.redhat.io
```

- **path-to-local-mirror は、ローカルミラーリポジトリーへのパスに置き換えます。**
- **ヒント: oc adm release mirror コマンドを入力すると、ローカルミラーへのパスが分かります。**

12. **Enforce if supported のボックスを選択します。**
13. **Create を選択してポリシーを作成します。**

14.1.8. 非接続カタログソースをデプロイするためのポリシーのデプロイ

マネージドクラスターに Catalogsource ポリシーをプッシュして、接続環境がある場所から非接続のローカルレジストリーにデフォルトの場所を変更します。

1. **Red Hat Advanced Cluster Management コンソールで Automate infrastructure >Clusters を選択します。**
2. **クラスター一覧でポリシーを受信するマネージドクラスターを検索します。**
3. **マネージドクラスターの 名前 ラベルの値をメモします。ラベルの形式は name=managed-cluster-name です。この値は、ポリシーのプッシュ時に使用します。**
4. **Red Hat Advanced Cluster Management コンソールメニューで、 Govern risk >Create policy を選択します。**
5. **YAML スイッチを On に設定して、ポリシーの YAML バージョンを表示します。**
6. **YAML コードのコンテンツをすべて削除します。**
7. **以下の YAML コンテンツをウィンドウに貼り付け、カスタムポリシーを作成します。**
8. **以下の YAML コンテンツをウィンドウに貼り付け、カスタムポリシーを作成します。**

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
        spec:
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: v1
                kind: Pod
```

```
            metadata:
              name: sample-nginx-pod
              namespace: default
            status:
              phase: Running
        remediationAction: inform
        severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      []  # selects all clusters if not specified
```

9. **以下の内容をポリシーに追加します。**

```
apiVersion: config.openshift.io/vi
kind: OperatorHub
metadata:
 name: cluster
spec:
 disableAllDefaultSources: true
```

10. **以下の内容を追加します。**

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: my-operator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
```

```
  image: <registry_host_name>:<port>/olm/redhat-operators:v1
  displayName: My Operator Catalog
  publisher: grpc
```

spec.image の値は、ローカルの制約付きのカタログソースイメージへのパスに置き換えます。

11. Red Hat Advanced Cluster Management コンソールのナビゲーションで、Automate infrastructure >Clusters を選択して、マネージドクラスターのステータスを確認します。ポリシーが適用されると、クラスターのステータスは Ready になります。

14.1.9. マネージドクラスターのパラメーターを変更するためのポリシーのデプロイ

ClusterVersion ポリシーをマネージドクラスターにプッシュし、アップグレード取得先のデフォルトの場所を変更します。

1. マネージドクラスターから、以下のコマンドを入力して ClusterVersion アップストリームパラメーターがデフォルトの OpenShift Update Service オペランドであることを確認します。

```
oc get clusterversion -o yaml
```

返される内容は以下のようになります。

```
apiVersion: v1
items:
- apiVersion: config.openshift.io/v1
  kind: ClusterVersion
[..]
  spec:
    channel: stable-4.4
    upstream: https://api.openshift.com/api/upgrades_info/v1/graph
```

2. ハブクラスターから、**oc get routes** というコマンドを入力して OpenShift Update Service オペランドへのルート URL を特定します。
ヒント: 今後の手順で使用できるようにこの値をメモします。

3. ハブクラスターの Red Hat Advanced Cluster Management コンソールメニューで、Govern risk >Create a policyを選択します。

4. **YAML** スイッチを On に設定して、ポリシーの YAML バージョンを表示します。

5. **YAML** コードのコンテンツをすべて削除します。

6. 以下の **YAML** コンテンツをウィンドウに貼り付け、カスタムポリシーを作成します。

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
```

```
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
        spec:
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: v1
                kind: Pod
                metadata:
                  name: sample-nginx-pod
                  namespace: default
                status:
                  phase: Running
          remediationAction: inform
          severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      []  # selects all clusters if not specified
```

7. **policy セクションの policy.spec に以下の内容を追加します。**

```
apiVersion: config.openshift.io/v1
  kind: ClusterVersion
  metadata:
    name: version
```

```
  spec:
    channel: stable-4.4
    upstream: https://example-cincinnati-policy-engine-uri/api/upgrades_info/v1/graph
```

spec.upstream の値は、ハブクラスター OpenShift Update Service オペランドへのパスに置き換えます。

ヒント: 以下の手順を実行すると、オペランドへのパスを確認できます。

a. **ハブクラスターで oc get routes -A コマンドを実行します。**

b. **cincinnati へのルートを見つけます。+ オペランドへのパスは、HOST/PORT フィールドの値です。**

8. **マネージドクラスター CLI で、ClusterVersion のアップストリームパラメーターがローカルハブクラスター OpenShift Update Service URL に更新されていることを確認します。これには以下のコマンドを入力します。**

```
oc get clusterversion -o yaml
```

結果は、以下の内容のようになります。

```
apiVersion: v1
items:
- apiVersion: config.openshift.io/v1
  kind: ClusterVersion
[..]
  spec:
    channel: stable-4.4
    upstream: https://<hub-cincinnati-uri>/api/upgrades_info/v1/graph
```

14.1.10. 利用可能なアップグレードの表示

以下の手順を実行して、マネージドクラスターで利用可能なアップグレード一覧を確認します。

1. **Red Hat Advanced Cluster Management コンソールにログインします。**
2. **ナビゲーションメニューで Automate infrastructure >Clusters を選択します。**
3. **状態が Ready のクラスターを選択します。**
4. **Actions メニューから Upgrade cluster を選択します。**
5. **オプションのアップグレードパスが利用可能であることを確認します。**
   注記: 現行バージョンがローカルのイメージリポジトリーにミラーリングされていない場合には、利用可能なアップグレードバージョンは表示されません。

14.1.11. クラスターのアップグレード

非接続レジストリーの設定後に、Red Hat Advanced Cluster Management および OpenShift Update Service は非接続レジストリーを使用して、アップグレードが利用可能かどうかを判断します。利用可能なアップグレードが表示されない場合は、クラスターの現行のリリースイメージと、1つ後のイメージがローカルリポジトリーにミラーリングされていることを確認します。クラスターの現行バージョンのリリースイメージが利用できない場合、アップグレードは利用できません。

以下の手順を実行してアップグレードします。

1. Red Hat Advanced Cluster Management コンソールで Automate infrastructure >Clusters を選択します。
2. そのクラスターの内、利用可能なアップグレードがあるかどうかを判断するクラスターを特定します。
3. 利用可能なアップグレードがある場合には、クラスターの Distribution version コラムで、アップグレードが利用可能であることが表示されます。
4. クラスターの Options メニュー、Upgrade cluster の順に選択します。
5. アップグレードのターゲットバージョン、Upgrade の順に選択します。

マネージドクラスターは、選択したバージョンに更新されます。

クラスターのアップグレードに失敗すると、Operator は通常アップグレードを数回再試行し、停止し、コンポーネントに問題があるステータスを報告します。場合によっては、アップグレードプロセスは、プロセスの完了を繰り返し試行します。アップグレードに失敗した後にクラスターを以前のバージョンにロールバックすることはサポートされていません。クラスターのアップグレードに失敗した場合は、Red Hat サポートにお問い合わせください。

第15章 マネージメントからのクラスターの削除

Red Hat Advanced Cluster Management for Kubernetes で作成したマネージメントから、OpenShift Container Platform クラスターを削除すると、このクラスターをデタッチ するか、破棄 できます。

クラスターをデタッチするとマネージメントから削除されますが、完全には削除されません。管理する場合には、もう一度インポートし直すことができます。このオプションは、クラスターが Ready 状態にある場合にだけ利用できます。

クラスターを破棄すると、マネージメントから削除され、クラスターのコンポーネントが削除されます。これは永続的であるため、クラスターを再度インポートおよび管理することはできません。

15.1. コンソールを使用したクラスターの削除

1. ナビゲーションメニューから Automate infrastructure >Clusters に移動します。
2. 管理から削除するクラスターの Option メニューを選択します。
3. Destroy cluster または Detach cluster を選択します。
   ヒント: 複数のクラスターをデタッチまたは破棄するには、デタッチまたは破棄するクラスターのチェックボックスを選択します。次に、Detach または Destroy を選択します。
4. クラスターの削除後に残りのリソースの削除 を続行します。

注記: **local-cluster** という名前のハブクラスターをデタッチしようとする場合には、デフォルトの **disableHubSelfManagement** 設定が **false** である点に注意してください。この設定が原因で、ハブクラスターはデタッチされると、自身を再インポートして管理し、**MultiClusterHub** コントローラーが調整されます。ハブクラスターがデタッチプロセスを完了して再インポートするのに時間がかかる場合があります。プロセスが終了するのを待たずにハブクラスターを再インポートする場合には、以下のコマンドを実行して **multiclusterhub-operator** Pod を再起動して、再インポートの時間を短縮できます。

```
oc delete po -n open-cluster-management `oc get pod -n open-cluster-management | grep multiclusterhub-operator| cut -d' ' -f1`
```

「ネットワーク接続時のオンラインインストール」で説明されているように、**disableHubSelfManagement** の値を **true** に指定して、自動的にインポートされないように、ハブクラスターの値を変更できます。

15.2. コマンドラインを使用したクラスターの削除

ハブクラスターのコマンドラインを使用してマネージドクラスターをデタッチするには、以下のコマンドを実行します。

```
oc delete managedcluster $CLUSTER_NAME
```

注記: **local-cluster** という名前のハブクラスターをデタッチしようとする場合には、デフォルトの **disableHubSelfManagement** 設定が **false** である点に注意してください。この設定が原因で、ハブクラスターはデタッチされると、自身を再インポートして管理し、**MultiClusterHub** コントローラーが調整されます。ハブクラスターがデタッチプロセスを完了して再インポートするのに時間がかかる場合があります。プロセスが終了するのを待たずにハブクラスターを再インポートする場合には、以下のコマンドを実行して **multiclusterhub-operator** Pod を再起動して、再インポートの時間を短縮できます。

```
oc delete po -n open-cluster-management `oc get pod -n open-cluster-management | grep multiclusterhub-operator| cut -d' ' -f1`
```

「ネットワーク接続時のオンラインインストール」で説明されているように、disableHubSelfManagement の値を true に指定して、自動的にインポートされないように、ハブクラスターの値を変更できます。

クラスターの削除後に残りのリソースの削除 を続行します。

15.3. クラスター削除後の残りのリソースの削除

削除したマネージドクラスターにリソースが残っている場合は、残りのすべてのコンポーネントを削除するための追加の手順が必要になります。これらの追加手順が必要な場合には、以下の例が含まれます。

- マネージドクラスターは、完全に作成される前にデタッチされ、klusterlet などのコンポーネントはマネージドクラスターに残ります。
- マネージドクラスターをデタッチする前に、クラスターを管理していたハブが失われたり、破棄されたりし、ハブからマネージドクラスターをデタッチする方法はありません。
- マネージドクラスターは、デタッチ時にオンライン状態ではありませんでした。

マネージドクラスターをデタッチするには、以下の手順を実行します。

1. oc コマンドラインインターフェースが設定されていることを確認してください。
2. また、マネージドクラスターに KUBECONFIG が設定されていることを確認してください。oc get ns | grep open-cluster-management-agent を実行すると、2 つの namespace が表示されるはずです。

```
open-cluster-management-agent         Active   10m
open-cluster-management-agent-addon   Active   10m
```

3. cleanup-managed-cluster スクリプトを deploy Git リポジトリーからダウンロード します。
4. 以下のコマンドを入力して cleanup-managed-cluster.sh スクリプトを実行します。

```
./cleanup-managed-cluster.sh
```

5. 以下のコマンドを実行して、namespace が両方削除されていることを確認します。

```
oc get ns | grep open-cluster-management-agent
```