



# Red Hat Advanced Cluster Management for Kubernetes 2.3

## クラスター

クラウドプロバイダー間でクラスターを作成、インポート、および管理する方法については、詳細をご覧ください。



## Red Hat Advanced Cluster Management for Kubernetes 2.3 クラスター

---

クラウドプロバイダー間でクラスターを作成、インポート、および管理する方法については、詳細をご覧ください。

Enter your first name here. Enter your surname here.

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

Enter your email address here.

## 法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Clusters.ent file |.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 概要

クラウドプロバイダー間でクラスターを作成、インポート、および管理する方法については、詳細をご覧ください。

## 目次

第1章 クラスターの管理 .....	6
第2章 クラスターライフサイクルのアーキテクチャー .....	7
第3章 サポート対象のクラウド .....	10
3.1. サポート対象のハブクラスタープロバイダー .....	10
3.2. サポート対象のマネージドクラスタープロバイダー .....	10
3.3. KUBECTL の設定 .....	11
第4章 クラスターのサイズ変更 (テクノロジープレビュー) .....	12
4.1. RED HAT ADVANCED CLUSTER MANAGEMENT で作成したマネージドクラスターのスケールリング .....	12
4.1.1. 自動スケールリング .....	12
4.1.1.1. 自動スケールリングの有効化 .....	12
4.1.1.2. 自動スケールリングの無効化 .....	13
4.1.2. クラスターの手動スケールリング .....	14
4.1.2.1. コンソールを使用した手作業でのクラスターのスケールリング .....	14
4.1.2.2. コマンドラインを使用した手作業でのクラスターのスケールリング .....	14
4.2. インポートされたマネージドクラスターのスケールリング .....	15
4.2.1. OpenShift Container Platform クラスター .....	15
4.2.2. Amazon Elastic Kubernetes Services .....	15
4.2.3. Google Kubernetes Engine .....	16
4.2.4. Microsoft Azure Kubernetes Service .....	16
4.2.5. VMware vSphere .....	16
4.2.6. Red Hat OpenStack Platform .....	16
4.2.7. ベアメタルクラスター .....	16
4.2.8. IBM Kubernetes Service .....	16
第5章 リリースイメージ .....	17
5.1. 利用可能なリリースイメージの同期 .....	18
5.1.1. 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 .....	19
5.1.2. 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 .....	20
第6章 ベアメタルアセットの作成および変更 .....	22
6.1. 前提条件 .....	22
6.2. コンソールを使用したベアメタルアセットの作成 .....	22
6.3. CLI を使用したベアメタルアセットの作成 .....	23
6.3.1. 前提条件 .....	23
6.3.2. ベアメタルノードの作成 .....	23
6.4. コンソールを使用したベアメタルアセットの一括インポート .....	25
6.4.1. 前提条件 .....	25
6.4.2. アセットのインポート .....	25
6.5. ベアメタルアセットの変更 .....	25
6.6. ベアメタルアセットの削除 .....	25
6.7. REST API を使用したベアメタルアセットの作成 .....	26
6.7.1. 前提条件 .....	26
6.7.2. ベアメタルノードの作成 .....	26
第7章 クラスターの作成 .....	28
7.1. クラスター作成時の追加のマニフェストの設定 .....	28
7.2. AMAZON WEB SERVICES でのクラスターの作成 .....	29
7.2.1. 前提条件 .....	30
7.2.2. コンソールを使用したクラスターの作成 .....	30
7.2.3. クラスターへのアクセス .....	32

7.3. MICROSOFT AZURE でのクラスターの作成	33
7.3.1. 前提条件	33
7.3.2. コンソールを使用したクラスターの作成	33
7.3.3. クラスターへのアクセス	35
7.4. GOOGLE CLOUD PLATFORM でのクラスターの作成	36
7.4.1. 前提条件	36
7.4.2. コンソールを使用したクラスターの作成	36
7.4.3. クラスターへのアクセス	38
7.5. VMWARE VSPHERE でのクラスターの作成	39
7.5.1. 前提条件	39
7.5.2. コンソールを使用したクラスターの作成	39
7.5.3. クラスターへのアクセス	41
7.6. RED HAT OPENSTACK PLATFORM でのクラスターの作成	42
7.6.1. 前提条件	42
7.6.2. コンソールを使用したクラスターの作成	42
7.6.3. クラスターへのアクセス	45
7.7. ベアメタルでのクラスターの作成	45
7.7.1. 前提条件	45
7.7.2. コンソールを使用したクラスターの作成	46
7.7.3. クラスターへのアクセス	49
7.8. 作成されたクラスターの休止 (テクノロジープレビュー)	50
7.8.1. コンソールを使用したクラスターの休止	50
7.8.2. CLI を使用したクラスターの休止	50
7.8.3. コンソールを使用して休止中のクラスターの通常操作を再開する手順	51
7.8.4. CLI を使用して休止中のクラスターの通常操作を再開する手順	51
<b>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</b> .....	<b>53</b>
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	53
8.1.1. 前提条件	53
8.1.2. クラスターのインポート	54
8.1.3. インポートされたクラスターの削除	56
8.2. CLI を使用したマネージドクラスターのインポート	57
8.2.1. 前提条件	57
8.2.2. サポートされているアーキテクチャー	57
8.2.3. インポートの準備	58
8.2.4. クラスターのインポート	58
8.2.5. klusterlet のインポート	59
8.2.6. CLI でのインポートされたクラスターの削除	60
8.3. クラスターの KLUSTERLET アドオン設定の変更	60
8.3.1. klusterlet アドオン設定の説明	61
8.3.2. ハブクラスターのコンソールを使用した変更	61
8.3.3. ハブクラスターのコマンドラインを使用した変更	62
<b>第9章 プロキシ環境でのクラスターの作成</b> .....	<b>63</b>
9.1. 既存クラスターでのクラスター全体のプロキシの有効化	64
<b>第10章 特定のクラスター管理ロールの設定</b> .....	<b>66</b>
<b>第11章 クラスターラベルの管理</b> .....	<b>68</b>
<b>第12章 マネージドクラスターで実行する ANSIBLE TOWER タスクの設定</b> .....	<b>69</b>
12.1. 前提条件	69
12.2. コンソールでを使用したクラスターでの実行用の ANSIBLEJOB テンプレート設定	69
12.3. ANSIBLEJOB テンプレートの作成	69

12.4. ラベルを使用したマネージドクラスターでの実行用の ANSIBLEJOB テンプレート設定	71
12.5. ANSIBLE ジョブのステータスの表示	73
<b>第13章 MANAGEDCLUSTERSETS の管理 (テクノロジープレビュー) .....</b>	<b>74</b>
13.1. MANAGEDCLUSTERSET の作成	74
13.1.1. コンソールを使用した ManagedClusterSet の作成	74
13.1.2. コマンドラインを使用した ManagedClusterSet の作成	75
13.2. クラスターの MANAGEDCLUSTERSET への追加	75
13.2.1. コンソールを使用したクラスターの ManagedClusterSet への追加	75
13.2.2. コマンドラインを使用した ManagedClusterSet へのクラスターの追加	75
13.3. MANAGEDCLUSTERSET からのマネージドクラスターの削除	77
13.3.1. コンソールを使用した ManagedClusterSet からのマネージドクラスターの削除	77
13.3.2. コマンドラインを使用した ManagedClusterSet からのクラスターの削除	77
13.4. MANAGEDCLUSTERSETBINDING リソースの作成	78
13.4.1. コンソールを使用した ManagedClusterSetBinding の作成	78
13.4.2. コマンドラインを使用した ManagedClusterSetBinding の作成	78
13.5. MANAGEDCLUSTERSET に対するユーザーまたはグループのロールベースのアクセス制御パーミッションの割り当て	79
13.6. PLACEMENT での MANAGEDCLUSTERSETS の使用	80
13.6.1. 配置の概要	80
13.6.2. 配置の例	81
13.6.3. 配置のデシジョン	82
<b>第14章 クラスタープールの管理 (テクノロジープレビュー) .....</b>	<b>84</b>
14.1. クラスタープールの作成	84
14.1.1. 前提条件	84
14.1.2. クラスタープールの作成	85
14.2. クラスタープールからのクラスターの要求	86
14.2.1. 前提条件	86
14.2.2. クラスタープールからのクラスターの要求	86
14.3. クラスタープールのスケーリング	86
14.4. クラスタープールリリースイメージの更新	87
14.5. クラスタープールの破棄	87
<b>第15章 CLUSTERCLAIMS .....</b>	<b>89</b>
15.1. 既存の CLUSTERCLAIM の表示	91
15.2. カスタム CLUSTERCLAIMS の作成	91
<b>第16章 DISCOVERY サービスの概要 (テクノロジープレビュー) .....</b>	<b>92</b>
16.1. コンソールでの検出の設定 (テクノロジープレビュー)	92
16.1.1. 前提条件	92
16.1.2. 検出の設定	92
16.2. CLI を使用した検出の有効化 (テクノロジープレビュー)	93
16.2.1. 前提条件	93
16.2.2. 検出の設定とプロセス	93
16.2.3. 検出されたクラスターの表示	93
16.2.3.1. DiscoveredClusters	94
16.3. 検出したクラスターの表示 (テクノロジープレビュー)	94
16.4. 検出クラスターのインポート (テクノロジープレビュー)	95
16.4.1. 前提条件	95
16.4.2. 検出クラスターのインポート	95
<b>第17章 クラスターのアップグレード .....</b>	<b>96</b>
17.1. チャンネルの選択	96

17.2. 非接続クラスターのアップグレード	97
17.2.1. 前提条件	98
17.2.2. 非接続ミラーレジストリーの準備	98
17.2.3. OpenShift Update Service の Operator のデプロイ	99
17.2.4. グラフデータの init コンテナの構築	99
17.2.5. ミラーリングされたレジストリーの証明書の設定	100
17.2.6. OpenShift Update Service インスタンスのデプロイ	101
17.2.7. 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)	101
17.2.8. 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	103
17.2.9. 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	105
17.2.10. 利用可能なアップグレードの表示	108
17.2.11. チャンネルの選択	108
17.2.12. クラスターのアップグレード	108
<b>第18章 マネージメントからのクラスターの削除</b> .....	<b>110</b>
18.1. コンソールを使用したクラスターの削除	110
18.2. コマンドラインを使用したクラスターの削除	110
18.3. クラスター削除後の残りのリソースの削除	111
18.4. クラスターの削除後の ETCD データベースのデフラグ	111
18.4.1. 前提条件	112
18.4.2. 手順	112



## 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](#)
- [リリースイメージ](#)
- [ベアメタルアセットの作成および変更](#)
- [認証情報の管理の概要](#)
- [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](#)
- [プロキシ環境でのクラスターの作成](#)
- [特定のクラスター管理ロールの設定](#)
- [クラスターラベルの管理](#)
- [ManagedClusterSets の管理 \(テクノロジープレビュー\)](#)
- [Placement での ManagedClusterSets の使用](#)
- [クラスタープールの管理 \(テクノロジープレビュー\)](#)
- [マネージドクラスターで実行する Ansible Tower タスクの設定 \(テクノロジープレビュー\)](#)
- [クラスタープールからのクラスターの要求](#)
- [Discovery の概要 \(テクノロジープレビュー\)](#)
- [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](#)

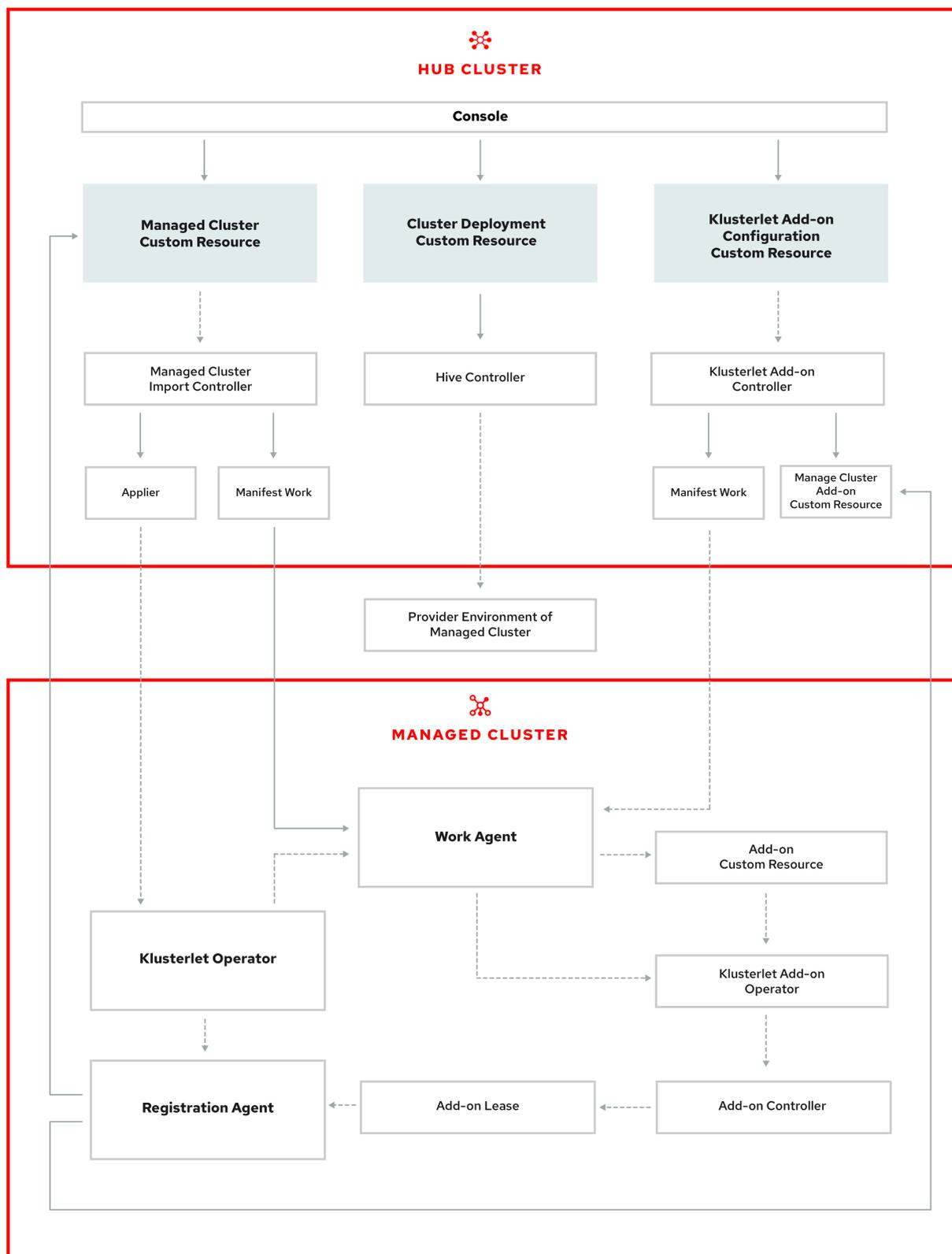
## 第2章 クラスターライフサイクルのアーキテクチャー

クラスターには、ハブクラスターとマネージドクラスターという2つの主なタイプがあります。

ハブクラスターは、Red Hat Advanced Cluster Management for Kubernetes でインストールされたメインとなるクラスターのことです。ハブクラスターを使用して他の Kubernetes クラスターを作成し、管理し、監視することができます。

マネージドクラスターは、ハブクラスターが管理する Kubernetes クラスターです。Red Hat Advanced Cluster Management ハブクラスターを使用してクラスターを作成することもできますが、ハブクラスターで管理する既存のクラスターをインポートすることもできます。

以下の図は、Red Hat Advanced Cluster Management for クラスター管理でインストールされるコンポーネントを示しています。



クラスターライフサイクル管理のアーキテクチャーのコンポーネントには、以下の項目が含まれます。

ハブクラスターのコンポーネント:

- **コンソール:** Red Hat Advanced Cluster Management マネージドクラスターのクラスターライフサイクルを管理する Web ベースのインターフェースを提供します。

- Hive Controller: Red Hat Advanced Cluster Management で作成するクラスターをプロビジョニングします。Hive コントローラーは、Red Hat Advanced Cluster Management で作成されたマネージドクラスターをデタッチおよび破棄します。
- マネージドクラスターのインポートコントローラー: `klusterlet Operator` をマネージドクラスターにデプロイします。
- `klusterlet` アドオンコントローラー: `klusterlet アドオン Operator` をマネージドクラスターにデプロイします。

マネージドクラスター上のコンポーネント:

- `klusterlet Operator`: マネージドクラスターに登録およびワークコントローラーをデプロイします。
- 登録エージェント: ハブクラスターを使用してマネージドクラスターに登録します。
- ワークエージェント: マニフェストはマネージドクラスターで機能します。
- `klusterlet アドオン Operator`: アドオンコンポーネントをデプロイします。

## 第3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。サポートされるクラウドの詳細は、「[Red Hat Advanced Cluster Management 2.3 サポートマトリックス](#)」を参照してください。

- [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。

### 3.1. 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 4.6.1 以降です。

- [Amazon Web Services 上の Red Hat OpenShift Container Platform](#) を参照してください。
- [Red Hat OpenShift Container Platform on Microsoft Azure](#) を参照してください。
- [Red Hat OpenShift Container Platform on Google Cloud Platform](#) を参照してください。
- [Red Hat OpenShift Dedicated \(OSD\)](#) (OpenShift Container Platform バージョン 4.7 以降) 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 4.7 以降) を参照してください。
- [Azure for Red Hat OpenShift \(ARO\)](#) (Red Hat OpenShift Container Platform バージョン 4.7 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 OpenStack](#) (OpenStack バージョン 16.1 以降) を参照してください。
- [OpenShift on Amazon Web Services \(ROSA\)](#) を参照してください。
- [テクノロジープレビュー: IBM Power Systems 上の OpenShift Container Platform](#) を参照してください。

### 3.2. 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、4.6.1 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。

- [Amazon Web Services 上の Red Hat OpenShift Container Platform](#) を参照してください。
- [Red Hat OpenShift Container Platform on Microsoft Azure](#) を参照してください。

- [Red Hat OpenShift Container Platform on Google Cloud Platform](#) を参照してください。
- [Red Hat OpenShift Dedicated \(OSD\)](#) (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 IBM Cloud \(ROKS\)](#) (Kubernetes 1.17 以降) を参照してください。
- [Azure Red Hat OpenShift \(ARO\)](#) 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 OpenStack](#) (OpenStack バージョン 16.1 以降) を参照してください。
- [OpenShift on Amazon Web Services \(ROSA\)](#) を参照してください。
- [About Red Hat OpenShift Kubernetes Engine](#) 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。
- [Google Kubernetes Engine](#) (Kubernetes 1.17 以降) 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。
- [Azure Kubernetes Service](#) (Kubernetes 1.19.6 以降) 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 Power Systems 上の OpenShift Container Platform](#) を参照してください。

### 3.3. 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4章 クラスターのサイズ変更 (テクノロジープレビュー)

仮想マシンのサイズやノード数などのマネージドクラスターの仕様をカスタマイズできます。

Red Hat Advanced Cluster Management for Kubernetes が管理するクラスターの多くは、Red Hat Advanced Cluster Management コンソールまたはコマンドライン、**MachinePool** リソースを使用してスケーリングできます。**MachinePool** リソースは、ハブクラスター上の Kubernetes リソースで、**MachineSet** リソースをマネージドクラスターでグループ化します。**MachinePool** リソースは、ゾーンの設定、インスタンスタイプ、ルートストレージなど、マシンリソースのセットを均一に設定します。また、マネージドクラスターで、必要なノード数を手動で設定したり、ノードの自動スケーリングを設定したりするのに役立ちます。

### 4.1. RED HAT ADVANCED CLUSTER MANAGEMENT で作成したマネージドクラスターのスケーリング

**MachinePool** リソースを使用して Red Hat Advanced Cluster Management で作成されたクラスターのスケーリングは、テクノロジープレビュー機能であり、Red Hat Advanced Cluster Management で作成されたベアメタルクラスターではサポートされません。

#### 4.1.1. 自動スケーリング

自動スケーリングを設定すると、トラフィックが少ない場合にリソースをスケールダウンし、多くのリソースが必要な場合に十分にリソースを確保できるようにスケールアップするなど、必要に応じてクラスターに柔軟性を持たせることができます。

##### 4.1.1.1. 自動スケーリングの有効化

- Red Hat Advanced Cluster Management コンソールを使用して、**MachinePool** リソースで自動スケーリングを有効化するには、以下の手順を実行します。
  1. Red Hat Advanced Cluster Management ナビゲーションで **Infrastructure > Clusters** に移動します。
  2. ターゲットクラスターの名前をクリックし、クラスターの詳細を表示します。
  3. **Machine pools** タブを選択して、マシンプール情報を表示します。
  4. マシンプールページのターゲットマシンプールの **Options** メニューから **Enable autoscale** を選択します。
  5. マシンセットレプリカの最小数および最大数を選択します。マシンセットレプリカは、クラスターのノードに直接マップします。
  6. **Scale** を選択して変更を送信します。  
変更内容をコンソールに反映するのに数分かかる場合があります。**Machine pools** タブの通知がある場合は、**View machines** をクリックしてスケーリング操作のステータスを表示できます。
- コマンドラインを使用して **MachinePool** リソースで自動スケーリングを有効にするには、以下の手順を実行します。
  1. 以下のコマンドを実行して、マシンプールの一覧を表示します。

```
oc get machinepools -n <managed-cluster-namespace>
```

**managed-cluster-namespace** は、ターゲットのマネージドクラスターの namespace に置き換えます。

2. 以下のコマンドを入力してマシンプールの YAML ファイルを編集します。

```
oc edit machinepool <name-of-MachinePool-resource> -n <namespace-of-managed-cluster>
```

**name-of-MachinePool-resource** は、**MachinePool** リソースの名前に置き換えます。

**namespace-of-managed-cluster** は、マネージドクラスターの namespace 名に置き換えます。

3. YAML ファイルから **spec.replicas** フィールドを削除します。
4. **spec.autoscaling.minReplicas** 設定および **spec.autoscaling.maxReplicas** フィールドをリソース YAML に追加します。
5. レプリカの最小数を **minReplicas** 設定に追加します。
6. レプリカの最大数を **maxReplicas** 設定に追加します。
7. ファイルを保存して変更を送信します。

マシンプールの自動スケーリングが有効になりました。

#### 4.1.1.2. 自動スケーリングの無効化

コンソールまたはコマンドラインを使用して自動スケーリングを無効にできます。

- Red Hat Advanced Cluster Management コンソールを使用して自動スケーリングを無効にするには、以下の手順を実行します。
  1. Red Hat Advanced Cluster Management ナビゲーションで **Infrastructure > Clusters** に移動します。
  2. ターゲットクラスターの名前をクリックし、クラスターの詳細を表示します。
  3. **Machine pools** タブを選択して、マシンプール情報を表示します。
  4. マシンプールページから、ターゲットマシンプールの **Options** メニューから **autoscale** を無効にします。
  5. 必要なマシンセットのレプリカ数を選択します。マシンセットのレプリカは、クラスター上のノードを直接マップします。
  6. **Scale** を選択して変更を送信します。コンソールに表示されるまでに数分かかる場合があります。
  7. **Machine pools** タブの通知で **View machine** をクリックし、スケーリングのステータスを表示します。
- コマンドラインを使用して自動スケーリングを無効にするには、以下の手順を実行します。
  1. 以下のコマンドを実行して、マシンプールの一覧を表示します。

```
oc get machinepools -n <managed-cluster-namespace>
```

**managed-cluster-namespace** は、ターゲットのマネージドクラスタの namespace に置き換えます。

2. 以下のコマンドを入力してマシンプールの YAML ファイルを編集します。

```
oc edit machinepool <name-of-MachinePool-resource> -n <namespace-of-managed-cluster>
```

**name-of-MachinePool-resource** は、**MachinePool** リソースの名前に置き換えます。

**namespace-of-managed-cluster** は、マネージドクラスタの namespace 名に置き換えます。

3. YAML ファイルから **spec.autoscaling** フィールドを削除します。
4. **spec.replicas** フィールドをリソース YAML に追加します。
5. **replicas** の設定にレプリカ数を追加します。
6. ファイルを保存して変更を送信します。

自動スケーリングが無効になりました。

## 4.1.2. クラスタの手動スケーリング

クラスタの自動スケーリングを有効にしない場合には、Red Hat Advanced Cluster Management コンソールまたはコマンドラインで、クラスタが管理するレプリカの静的数を変更できます。これにより、必要に応じてサイズを増減できます。

### 4.1.2.1. コンソールを使用した手作業でのクラスタのスケーリング

Red Hat Advanced Cluster Management コンソールを使用して **MachinePool** リソースをスケーリングするには、以下の手順を実行します。

1. Red Hat Advanced Cluster Management ナビゲーションで **Infrastructure** > **Clusters** に移動します。
2. ターゲットクラスタの名前をクリックし、クラスタの詳細を表示します。
3. **Machine pools** タブを選択して、マシンプール情報を表示します。  
**注記:** **Autoscale** フィールドの値が **Enabled** になっている場合に、「[自動スケーリングの無効化](#)」の手順を実行して自動スケーリング機能を無効にするようにしてください。
4. マシンプールの **Options** メニューから、**Scale machine pool** を選択します。
5. マシンプールをスケーリングするようにマシンセットのレプリカ数を調整します。
6. **Scale** を選択して変更を実装します。

### 4.1.2.2. コマンドラインを使用した手作業でのクラスタのスケーリング

コマンドラインを使用して **MachinePool** リソースをスケーリングするには、以下の手順を実行します。

1. 以下のコマンドを実行して、マシンプールの一覧を表示します。

```
oc get machinepools -n <managed-cluster-namespace>
```

**managed-cluster-namespace** は、ターゲットのマネージドクラスターの namespace に置き換えます。

2. 以下のコマンドを入力してマシンプールの YAML ファイルを編集します。

```
oc edit machinepool <name-of-MachinePool-resource> -n <namespace-of-managed-cluster>
```

**name-of-MachinePool-resource** は、**MachinePool** リソースの名前に置き換えます。

**namespace-of-managed-cluster** は、マネージドクラスターの namespace 名に置き換えます。

3. YAML の **spec.replicas** 設定は、レプリカの数に更新します。
4. ファイルを保存して変更を送信します。

クラスターが新しいサイズ設定を使用するようになりました。

## 4.2. インポートされたマネージドクラスターのスケーリング

インポートしたマネージドクラスターには、Red Hat Advanced Cluster Management で作成したクラスターと同じリソースがありません。そのため、クラスターのスケーリングの手順が異なります。プロバイダーのドキュメントには、インポートしたクラスターのスケーリング方法についての情報が含まれます。

利用可能なプロバイダーごとの推奨設定は以下の一覧を参照してください。ただし、詳細な情報については、ドキュメントも参照してください。

- [OpenShift Container Platform クラスター](#)
- [Amazon Elastic Kubernetes Services](#)
- [Google Kubernetes Engine](#)
- [Microsoft Azure Kubernetes Service](#)
- [VMware vSphere](#)
- [Red Hat OpenStack Platform](#)
- [ベアメタルクラスター](#)
- [IBM Kubernetes Service](#)

### 4.2.1. OpenShift Container Platform クラスター

お使いのバージョンに該当する OpenShift Container Platform ドキュメントの「[クラスターのスケーリングに関する推奨プラクティス](#)」および「[MachineSet の手動によるスケーリング](#)」を参照してください。

### 4.2.2. Amazon Elastic Kubernetes Services

インポートした Amazon EKS クラスターにあるノード数を変更する場合には、「[Cluster autoscaler](#)」でクラスターのスケーリングに関する情報を参照してください。

### 4.2.3. Google Kubernetes Engine

インポートした Google Kubernetes Engine クラスターにあるノード数を変更する場合には、「[Resizing a cluster](#)」でクラスターのスケーリングに関する情報を参照してください。

### 4.2.4. Microsoft Azure Kubernetes Service

インポートした Azure Kubernetes Services クラスターにあるノード数を変更する場合には、「[Scaling a cluster](#)」でクラスターのスケーリングに関する情報を参照してください。

### 4.2.5. VMware vSphere

インポートした VMware vSphere クラスターにあるノード数を変更する場合には、「[Edit cluster settings](#)」でクラスターのスケーリングに関する情報を参照してください。

### 4.2.6. Red Hat OpenStack Platform

インポートした Red Hat OpenStack Platform クラスターにあるノード数を変更する場合には、「[Auto scaling for instances](#)」でクラスターのスケーリングに関する情報を参照してください。

### 4.2.7. ベアメタルクラスター

インポートしたベアメタルクラスターにあるノード数を変更する場合には、「[Expanding the cluster](#)」でクラスターのスケーリングに関する情報を参照してください。

注記: ベアメタルクラスターは、ハブクラスターが OpenShift Container Platform バージョン 4.6 以降である場合にのみサポートされます。

### 4.2.8. IBM Kubernetes Service

インポートした IBM Kubernetes Service クラスターにあるノード数を変更する場合には、「[Adding worker nodes and zones to clusters](#)」でクラスターのスケーリングに関する情報を参照してください。

## 第5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 チャン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:4.6.23-x86_64`
- `quay.io/openshift-release-dev/ocp-release:4.7.4-x86_64`
- `quay.io/openshift-release-dev/ocp-release:4.8.0-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。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 チャンネルイメージが含まれます。リポジトリには、**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。

リポジトリには、**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 チャンネル) 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 チャン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 **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 **acm-hive-openshift-releases** の **README.md** に従って、フォークした GitHub リポジトリから **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、バージョン 2.3 の [acm-hive-openshift-releases GitHub repository](#) のファイルで参照されます。

## 5.1. 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、バージョン 2.3 の [acm-hive-openshift-releases GitHub リポジトリ](#) 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5.1 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. バージョン 2.3 用の [acm-hive-openshift-releases GitHub リポジトリ](#) 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、これらのチャン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
```

### 5.1.1. 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. バージョン 2.3 用の [acm-hive-openshift-releases GitHub リポジトリ](#) をフォークします。
3. **./subscribe/channel.yaml** ファイルを更新して、**stolostron** ではなく、フォークしたリポジトリの 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へのパスに置き換えます。

- 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リーから取得できます。
- フォークしたリポジリーに変更をコミットし、マージします。
- acm-hive-openshift-releases** リポジ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。

- 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。

- 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、利用可能なものを表示し、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。

- 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。

### 5.1.2. 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。

- オンラインシステムを使用している場合には、バージョン 2.3 でクラスターイメージセットを含む [acm-hive-openshift-releases GitHub リポジリー](#) 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. クラスターイメージセットを使用して、マネージドクラスターと非接続リポジトリー間のマッピングを追加します。 `ImageContentSourcePolicy` オブジェクトを使用してマッピングを完了する方法についての詳細は、「[イメージレジストリーリポジトリーのミラーリングの設定](#)」を参照してください。
4. `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。
5. 残りの OpenShift Container Platform リリースイメージの `clusterImageSet` YAML ファイルを、イメージの保存先の正しいオフラインリポジトリーを参照するように変更します。更新は次の例のようになります。

```
apiVersion: hive.openshift.io/v1
kind: ClusterImageSet
metadata:
  name: img4.4.0-rc.6-x86-64
spec:
  releaseImage: IMAGE_REGISTRY_IPADDRESS_or_DNSNAME/REPO_PATH/ocp-
  release:4.4.0-rc.6-x86_64
```

YAML ファイルで参照されているオフラインイメージレジストリーにイメージがロードされていることを確認します。

6. 各 YAML ファイルに以下のコマンドを入力して、各 `clusterImageSets` を作成します。

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

`clusterImageSet_FILE` を、クラスターイメージセットファイルの名前に置き換えます。以下は例になります。

```
oc create -f img4.9.9-x86_64.yaml
```

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。

7. 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。イメージ URL が存在しない場合は、イメージ URL を追加すると新しい `clusterImageSets` が作成されます。
8. 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6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、OpenShift Container Platform クラスターで実行されるように設定する仮想サーバーまたは物理サーバーです。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できます。ベアメタルアセットはマネージドクラスターにデプロイできます。

ハブクラスターのインベントリーコントローラーは、ベアメタルのアセットのインベントリーレコードを保持する **BareMetalAsset** というカスタムリソース定義 (CRD) を定義します。マネージドクラスターをプロビジョニングする場合には、インベントリーコントローラーは、マネージドクラスター内にある対応する **BareMetalHost** リソースと、**BareMetalAsset** インベントリーレコードを調整します。

Red Hat Advanced Cluster Management は **BareMetalAsset** CR を使用して、設定管理データベース (CMDB) または同様のシステムで入力したレコードに基づいてクラスターハードウェアをプロビジョニングします。外部ツールまたは自動化は CMDB をポーリングし、Red Hat Advanced Cluster Management API を使用して、マネージドクラスターでの後続のデプロイメントに備え、対応する **BareMetalAsset** と **Secret** リソースをハブクラスターに作成します。

以下の手順を使用して、Red Hat Advanced Cluster Management が管理するクラスターのベアメタルアセットを作成して管理します。

- [前提条件](#)
- [コンソールを使用したベアメタルアセットの作成](#)
- [CLI を使用したベアメタルアセットの作成](#)
- [コンソールを使用したベアメタルアセットの一括インポート](#)
- [ベアメタルアセットの変更](#)
- [ベアメタルアセットの削除](#)
- [REST API を使用したベアメタルアセットの作成](#)

### 6.1. 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 4.6 以降に、Red Hat Advanced Cluster Management ハブクラスターをデプロイしておく。
- Red Hat Advanced Cluster Management 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の認証情報には、管理者が提供するアセットの項目 (ユーザー名、パスワード、Baseboard Management Controller (BMC) アドレスの起動 NIC MAC アドレス) が含まれます。

### 6.2. 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 `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. 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。

## 6.3. CLI を使用したベアメタルアセットの作成

BareMetalAsset CR を使用して、クラスター内の特定の namespace のベアメタルアセットを作成します。各 BareMetalAsset には、同じ namespace に対応の Secret があり、そこには BMC (Baseboard Management Controller) 認証情報とシークレット名が含まれます。

### 6.3.1. 前提条件

- ハブクラスターに Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをインストールする。
- Red Hat OpenShift CLI (oc) をインストールする。
- `cluster-admin` 権限を持つユーザーとしてログインすること。

### 6.3.2. ベアメタルノードの作成

1. 環境にベアメタルアセットをインストールしてプロビジョニングします。
2. BMC の電源をオンにし、ハードウェアの IPMI または Redfish BMC アドレスおよび MAC アドレスを書き留めます。

- 以下の BareMetalAsset および Secret CR を作成し、ファイルを baremetalasset-cr.yaml として保存します。

```

apiVersion: inventory.open-cluster-management.io/v1alpha1
kind: BareMetalAsset
metadata:
  name: <baremetalasset-machine>
  namespace: <baremetalasset-namespace>
spec:
  bmc:
    address: ipmi://<out_of_band_ip>:<port>
    credentialsName: baremetalasset-machine-secret
    bootMACAddress: "00:1B:44:11:3A:B7"
    hardwareProfile: "hardwareProfile"
    role: "<role>"
    clusterName: "<cluster name>"
---
apiVersion: v1
kind: Secret
metadata:
  name: baremetalasset-machine-secret
type: Opaque
data:
  username: <username>
  password: <password>

```

- baremetalasset-machine は、ベアメタルアセットが置かれているマシンの名前に置き換えます。作成時に、マネージドクラスターの BareMetalHost は、ハブクラスター上の対応する BareMetalAsset と同じ名前を取得します。BareMetalHost 名は常に対応する BareMetalAsset 名と一致している必要があります。
  - baremetalasset-namespace は、ベアメタルアセットが作成されるクラスター namespace に置き換えます。
  - out\_of\_band\_ip および port は、ベアメタルアセットのアドレスおよびポートに置き換えます。Redfish アドレス設定には、redfish://<out-of-band-ip>/redfish/v1/Systems/1 のアドレス形式を使用します。
  - ロール は、worker か、master に置き換えるか、またはマシンのロールの種類に応じてそのままにします。ロール 設定を使用して、クラスター内の固有のマシンロールタイプに、ベアメタルアセットを一致させます。指定のマシンロールタイプの BareMetalAsset リソースは、別のロールを満たすためには使用しないでください。ロールの値は、キーが inventory.open-cluster-management.io/role のラベル値として使用されます。これにより、クラスター管理アプリケーションまたはユーザーは、特定のロール向けに用意されたインベントリーについてクエリーできます。
  - cluster\_name は、クラスターの名前に置き換えます。この名前は、クラスター管理アプリケーションまたはユーザーが、特定のクラスターに関連付けられたインベントリーのクエリーに使用します。クラスターデプロイメントに追加せずにベアメタルアセットを作成するには、この値を空欄のままにします。
  - username は、シークレットのユーザー名に置き換えます。
  - password は、シークレットのパスワードに置き換えます。
- 以下のコマンドを実行して BareMetalAsset CR を作成します。

■

```
$ oc create -f baremetalasset-cr.yaml
```

5. BareMetalAsset が正常に作成されていることを確認します。

```
$ oc get baremetalassets -A
```

出力例

NAMESPACE	NAME	AGE
ocp-example-bm	baremetalasset-machine	2m
ocp-example-bm	csv-f24-h27-000-r630-master-1-1	4d21h

## 6.4. コンソールを使用したベアメタルアセットの一括インポート

CSV形式の一覧を使用して、Red Hat Advanced Cluster Management for Kubernetes コンソールでベアメタルアセットを一括インポートできます。

### 6.4.1. 前提条件

- 1つ以上のスポーククラスターを管理するハブクラスターに Red Hat Advanced Cluster Management をインストールする。
- OpenShift Container Platform CLI (oc) をインストールします。
- cluster-admin 権限を持つユーザーとしてログインすること。

### 6.4.2. アセットのインポート

ベアメタルアセットをインポートするには、以下の手順を実行します。

1. Red Hat Advanced Cluster Management コンソールのナビゲーションメニューで Cluster management > Bare metal assets を選択します。
2. Import assets を選択し、ベアメタルアセットのデータを含む CSV ファイルをインポートします。CSV ファイルには、以下のヘッダーコラムが必要です。

```
hostName, hostNamespace, bmcAddress, macAddress, role (optional), username, password
```

## 6.5. 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、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。

## 6.6. 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。

コンソールで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、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。

## 6.7. REST API を使用したベアメタルアセットの作成

OpenShift Container Platform REST API を使用して、Red Hat Advanced Cluster Management クラスターで使用するベアメタルアセットを管理できます。これは、お使いの環境でベアメタルアセットを管理するために別の CMDB アプリケーションまたはデータベースがある場合に役立ちます。

### 6.7.1. 前提条件

- ハブクラスターに Red Hat Advanced Cluster Management for Kubernetes のハブクラスターをインストールする。
- OpenShift Container Platform CLI (oc) をインストールします。
- cluster-admin 権限を持つユーザーとしてログインすること。

### 6.7.2. ベアメタルノードの作成

REST API を使用してベアメタルアセットを作成するには、以下を実行します。

1. ハブクラスターのログイントークンを取得して、コマンドラインでクラスターにログインします。以下は例になります。

```
$ oc login --token=<login_token> --server=https://<hub_cluster_api_url>:6443
```

2. 以下の curl コマンドを、クラスターに追加するベアメタルアセットの詳細を使用して変更し、コマンドを実行します。

```
$ curl --location --request POST '<hub_cluster_api_url>:6443/apis/inventory.open-cluster-management.io/v1alpha1/namespaces/<bare_metal_asset_namespace>/baremetalassets?fieldManager=kubectl-create' \
--header 'Authorization: Bearer <login_token>' \
--header 'Content-Type: application/json' \
--data-raw '{
  "apiVersion": "inventory.open-cluster-management.io/v1alpha1",
  "kind": "BareMetalAsset",
  "metadata": {
    "name": "<baremetalasset_name>",
    "namespace": "<bare_metal_asset_namespace>"
  },
  "spec": {
    "bmc": {
```

```

    "address": "ipmi://<ipmi_address>",
    "credentialsName": "<credentials-secret>"
  },
  "bootMACAddress": "<boot_mac_address>",
  "clusterName": "<cluster_name>",
  "hardwareProfile": "hardwareProfile",
  "role": "worker"
}
}'

```

- **baremetalasset-name** は、ベアメタルアセットの名前に置き換えます。作成時に、マネージドクラスターの **BareMetalHost** は、ハブクラスター上の対応する **BareMetalAsset** と同じ名前を取得します。 **BareMetalHost** 名は常に対応する **BareMetalAsset** 名と一致している必要があります。
- **baremetalasset-namespace** は、ベアメタルアセットが作成されるクラスター namespace に置き換えます。
- **out\_of\_band\_ip** および **port** は、ベアメタルアセットのアドレスおよびポートに置き換えます。 Redfish アドレス設定には、 `redfish://<out-of-band-ip>/redfish/v1/Systems/1` のアドレス形式を使用します。
- **ロール** は、 **worker** か、 **master** に置き換えるか、またはマシンのロールの種類に応じてそのままにします。ロール設定を使用して、クラスター内の固有のマシンロールタイプに、ベアメタルアセットを一致させます。指定のマシンロールタイプの **BareMetalAsset** リソースは、別のロールを満たすためには使用しないでください。ロールの値は、キーが `inventory.open-cluster-management.io/role` のラベル値として使用されます。これにより、クラスター管理アプリケーションまたはユーザーは、特定のロール向けに用意されたインベントリについてクエリーできます。
- **cluster\_name** は、クラスターの名前に置き換えます。この名前は、クラスター管理アプリケーションまたはユーザーが、特定のクラスターに関連付けられたインベントリのクエリーに使用します。クラスターデプロイメントに追加せずにベアメタルアセットを作成するには、この値を空欄のままにします。  
注記: 以前の `curl` コマンドでは、API サーバーが HTTPS 経由で提供され、安全にアクセスされることを前提としています。開発またはテスト環境では、 `--insecure` パラメーターを指定できます。

ヒント: `--v=9` を `oc` コマンドに追加して、結果となるアクションの出力を未加工で表示できます。これは、`oc` コマンドの REST API ルートの認定に役立ちます。

## 第7章 クラスタの作成

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を作成する方法を説明します。

- [クラスタ作成時の追加のマニフェストの設定](#)
- [Amazon Web Services でのクラスタの作成](#)
- [Google Cloud Platform でのクラスタの作成](#)
- [Microsoft Azure でのクラスタの作成](#)
- [VMware vSphere でのクラスタの作成](#)
- [Red Hat OpenStack Platform でのクラスタの作成](#)
- [ベアメタルでのクラスタの作成](#)

### 7.1. クラスタ作成時の追加のマニフェストの設定

追加の Kubernetes リソースマニフェストは、クラスタ作成のインストールプロセス中に設定できません。これは、ネットワークの設定やロードバランサーの設定など、シナリオの追加マニフェストを設定する必要がある場合に役立ちます。

クラスタを作成する前に、追加のリソースマニフェストが含まれる **ConfigMap** を指定する **ClusterDeployment** リソースへの参照を追加する必要があります。

注記： **ClusterDeployment** リソースと **ConfigMap** は同じ namespace にある必要があります。以下の例で、どのような内容かを紹介しています。

- リソースマニフェストを含む **ConfigMap**  
**ConfigMap** リソースが別のマニフェストが含まれる **ConfigMap**。リソースマニフェストの **ConfigMap** には、`data.<resource_name>\.yaml` パターンに追加されたリソース設定が指定されたキーを複数含めることができます。

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: <my-baremetal-cluster-install-manifests>
  namespace: <mynamespace>
data:
  99_metal3-config.yaml: |
    kind: ConfigMap
    apiVersion: v1
    metadata:
      name: metal3-config
      namespace: openshift-machine-api
    data:
      http_port: "6180"
      provisioning_interface: "enp1s0"
      provisioning_ip: "172.00.0.3/24"
      dhcp_range: "172.00.0.10,172.00.0.100"
      deploy_kernel_url: "http://172.00.0.3:6180/images/ironic-python-agent.kernel"
      deploy_ramdisk_url: "http://172.00.0.3:6180/images/ironic-python-agent.initramfs"
      ironic_endpoint: "http://172.00.0.3:6385/v1/"
```

```

ironic_inspector_endpoint: "http://172.00.0.3:5150/v1/"
cache_url: "http://192.168.111.1/images"
rhcos_image_url: "https://releases-art-
rhcos.svc.ci.openshift.org/art/storage/releases/rhcos-
4.3/43.81.201911192044.0/x86_64/rhcos-43.81.201911192044.0-
openstack.x86_64.qcow2.gz"

```

- リソースマニフェスト ConfigMap が参照される ClusterDeployment リソースマニフェスト ConfigMap は spec.provisioning.manifestsConfigMapRef で参照されます。

```

apiVersion: hive.openshift.io/v1
kind: ClusterDeployment
metadata:
  name: <my-baremetal-cluster>
  namespace: <mynamespace>
  annotations:
    hive.openshift.io/try-install-once: "true"
spec:
  baseDomain: test.example.com
  clusterName: <my-baremetal-cluster>
  controlPlaneConfig:
    servingCertificates: {}
  platform:
    baremetal:
      libvirtSSHPrivateKeySecretRef:
        name: provisioning-host-ssh-private-key
  provisioning:
    installConfigSecretRef:
      name: <my-baremetal-cluster-install-config>
    sshPrivateKeySecretRef:
      name: <my-baremetal-hosts-ssh-private-key>
    manifestsConfigMapRef:
      name: <my-baremetal-cluster-install-manifests>
    imageSetRef:
      name: <my-clusterimageset>
    sshKnownHosts:
      - "10.1.8.90 ecdsa-sha2-nistp256
AAAAE2VjZHNhLXvVVVKUYVkuYvkuYgkuyTCYTytfkufTYAAAAIbmlzdHAyNTYAAABB
BKWjJRzeUVuZs4yxSy4eu45xiANFIllwE3e1aPzGD58x/NX7Yf+S8eFKq4RrsfSaK2hVJyJ
jvVlhUsU9z2sBJP8="
    pullSecretRef:
      name: <my-baremetal-cluster-pull-secret>

```

## 7.2. 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。

- [前提条件](#)
- [コンソールを使用したクラスターの作成](#)
- [クラスターへのアクセス](#)

## 7.2.1. 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 認証情報。詳細は、「[Amazon Web Services の認証情報の作成](#)」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。

## 7.2.2. コンソールを使用したクラスターの作成

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 Infrastructure > Clusters に移動します。
2. Clusters ページで、Create cluster をクリックします。  
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。
3. インフラストラクチャプロバイダー用の Kubernetes ディストリビューションおよび Amazon Web Services を選択します。残りの手順は、選択した内容により異なります。
4. 一覧で利用可能な認証情報から、お使いのインフラストラクチャプロバイダーの認証情報を選択します。設定されていない場合や、新たに設定する場合には、Add credential を参照してください。認証情報の作成に関する詳細は、「[Amazon Web Services の認証情報の作成](#)」を参照してください。
5. クラスターの詳細を追加します。
  - a. 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。
  - b. クラスターセットを既存のクラスターセットに追加する場合は、クラスターセットを 1 つ指定します。クラスターの作成時に cluster-admin 権限がない場合に、クラスターを作成する clusterset-admin パーミッションがあるクラスターセットを選択する必要があります。

す。クラスターセットを選択しない場合には、クラスターの作成には失敗します。選択するクラスターセットがない場合には、クラスター管理者に連絡して、クラスターセットへの `clusterset-admin` パーミッションを受け取ってください。

- c. AWS アカウントに設定した Base DNS domain 情報を指定します。選択した認証情報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。
  - d. 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。
  - e. 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。
6. マスタープールの詳細を入力します。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の管理を共有します。オプションの情報には以下のフィールドが含まれます。
- a. マスタープールを実行する ゾーン を1つ以上指定します。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。ゾー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、ゾーンの距離が離れると、より分散されません。
  - b. マスターノードの インスタンスタイプ を指定します。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。デフォルト値は `mx5.xlarge - 4 vCPU, 16 GiB RAM - General Purpose` です。
  - c. Root ストレージの割り当てを入力します。デフォルト値は、ルートストレージ 100 GiB です。
7. ワーカープールに関する情報を入力します。ワーカープールにワーカーノードを作成し、クラスターのコンテナ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。オプションの情報には以下のフィールドが含まれます。
- a. ワーカープールの名前を追加します。
  - b. ワーカープールを実行する ゾーン を指定します。より分散されている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。ゾー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、ゾーンの距離が離れると、より分散されません。
  - c. ワーカープールの インスタンスタイプ を指定します。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。デフォルト値は `mx5.xlarge - 4 vCPU, 16 GiB RAM - General Purpose` です。
  - d. ワーカープールの Node count を入力します。ワーカープールを定義する場合にこの設定は必須です。
  - e. Root ストレージの割り当てを入力します。デフォルト値は、ルートストレージ 100 GiB です。ワーカープールを定義する場合にこの設定は必須です。  
`Add worker pool` をクリックして追加のワーカープールを定義できます。

8. クラスターのネットワーク情報を入力します。この情報は必須です。IPv6 を使用するには、複数のネットワークが必要です。
  - a. ネットワークタイプを選択します。デフォルト値は OpenShiftSDN です。IPv6 を使用するには、OVNKubernetes の設定は必須です。
  - b. Cluster network CIDR を選択します。これは、Pod IP アドレスに使用できる IP アドレスの数およびリストです。この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 10.128.0.0/14 です。
  - c. ネットワークホストの接頭辞を指定します。これにより、各ノードにサブネットプレフィックスの長さを設定します。デフォルト値は 23 です。
  - d. サービスネットワーク CIDR でサービスの IP アドレスのブロックを指定します。この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 172.30.0.0/16 です。
  - e. マシン CIDR で OpenShift Container Platform ホストで使用される IP アドレスのブロックを指定します。この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。  
Add network をクリックして、追加のネットワークを追加できます。IPv6 アドレスを使用している場合は、複数のネットワークが必要です。
9. クラスターのインストールまたはアップグレードの前後に実行する Ansible Automation template (任意) を追加します。  
Add automation template をクリックしてテンプレートを作成できます。
10. 情報を確認し、必要に応じてカスタマイズします。
  - a. YAML スライダーを On にクリックし、パネルに install-config.yaml ファイルの内容を表示します。
  - b. YAML ファイルをカスタム設定で編集します。
11. Create を選択してクラスターを作成します。  
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。

### 7.2.3. 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 ナビゲーションメニューで 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。

## 7.3. 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 クラスターをデプロイできます。

- [前提条件](#)
- [コンソールを使用したクラスターの作成](#)
- [クラスターへのアクセス](#)

### 7.3.1. 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 認証情報。詳細は、「[Microsoft Azure の認証情報の作成](#)」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。「[Microsoft Azure Portal](#)」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。

### 7.3.2. コンソールを使用したクラスターの作成

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 Infrastructure > Clusters に移動します。
2. Clusters ページで、Create cluster をクリックします。  
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。
3. インフラストラクチャプロバイダーに Kubernetes ディストリビューションおよび Microsoft Azure を選択します。残りの手順は、選択した内容により異なります。
4. 一覧で利用可能な認証情報から、お使いのインフラストラクチャプロバイダーの認証情報を選択します。設定されていない場合や、新たに設定する場合には、Add credential を参照して

ください。認証情報の作成に関する詳細は、「[Microsoft Azure の認証情報の作成](#)」を参照してください。

5. クラスターの詳細を追加します。
  - a. 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。
  - b. クラスターセットを既存のクラスターセットに追加する場合は、クラスターセットを1つ指定します。クラスターの作成時に `cluster-admin` 権限がない場合に、クラスターを作成する `clusterset-admin` パーミッションがあるクラスターセットを選択する必要があります。クラスターセットを選択しない場合には、クラスターの作成には失敗します。選択するクラスターセットがない場合には、クラスター管理者に連絡して、クラスターセットへの `clusterset-admin` パーミッションを受け取ってください。
  - c. Azure アカウントに設定した Base DNS domain 情報を指定します。選択した認証情報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、[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。
  - d. 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。
  - e. 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。
6. マスタープールの詳細を入力します。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の管理を共有します。オプションの情報には以下のフィールドが含まれます。
  - a. マスタープールを実行する リージョン を指定します。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。ゾー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、ゾーンの距離が離れると、より分散されます。
  - b. マスターノードの インスタンスタイプ を指定します。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。デフォルト値は、Standard\_D4s\_v3 - 4 vCPU, 16 GiB RAM - General Purpose です。
  - c. Root ストレージの割り当てを入力します。デフォルト値は、ルートストレージ 128 GiB です。
7. ワーカープールに関する情報を入力します。ワーカープールにワーカーノードを作成し、クラスターのコンテナ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。オプションの情報には以下のフィールドが含まれます。
  - a. ワーカープールの名前を追加します。
  - b. ワーカープールを実行する ゾーン を指定します。より分散されている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。ゾー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、ゾーンの距離が離れると、より分散されます。

- c. ワーカープールのインスタンスタイプを指定します。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。デフォルト値は Standard\_D2s\_v3 - 2 vCPU, 8 GiB - General Purpose です。
  - d. ワーカープールの Node count を入力します。ワーカープールを定義する場合にこの設定は必須です。
  - e. Root ストレージの割り当てを入力します。デフォルト値は、ルートストレージ 128 GiB です。ワーカープールを定義する場合にこの設定は必須です。  
Add worker pool をクリックして追加のワーカープールを定義できます。
8. クラスターのネットワーク情報を入力します。この情報は必須です。IPv6 を使用するには、複数のネットワークが必要です。
- a. ネットワークタイプを選択します。デフォルト値は OpenShiftSDN です。IPv6 を使用するには、OVNKubernetes の設定は必須です。
  - b. Cluster network CIDR を選択します。これは、Pod IP アドレスに使用できる IP アドレスの数およびリストです。この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 10.128.0.0/14 です。
  - c. ネットワークホストの接頭辞を指定します。これにより、各ノードにサブネットプレフィックスの長さを設定します。デフォルト値は 23 です。
  - d. サービスネットワーク CIDR でサービスの IP アドレスのブロックを指定します。この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 172.30.0.0/16 です。
  - e. マシン CIDR で OpenShift Container Platform ホストで使用される IP アドレスのブロックを指定します。この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。  
Add network をクリックして、追加のネットワークを追加できます。IPv6 アドレスを使用している場合は、複数のネットワークが必要です。
9. クラスターのインストールまたはアップグレードの前後に実行する Ansible Automation template (任意) を追加します。  
Add automation template をクリックしてテンプレートを作成できます。
10. 情報を確認し、必要に応じてカスタマイズします。
- a. YAML スライダーを On にクリックし、パネルに install-config.yaml ファイルの内容を表示します。
  - b. YAML ファイルをカスタム設定で編集します。
11. Create を選択してクラスターを作成します。  
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。

### 7.3.3. 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 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。

## 7.4. 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。

- [前提条件](#)
- [コンソールを使用したクラスターの作成](#)
- [クラスターへのアクセス](#)

### 7.4.1. 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 認証情報。詳細は、「[Google Cloud Platform の認証情報の作成](#)」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。

### 7.4.2. コンソールを使用したクラスターの作成

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 Infrastructure > Clusters に移動します。
2. Clusters ページで、Create Cluster タブを選択します。  
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。

3. インフラストラクチャプロバイダーに、Kubernetes ディストリビューションおよび Google Cloud を選択します。残りの手順は、選択した内容により異なります。
4. 一覧で利用可能な認証情報からお使いの認証情報を選択します。設定されていない場合や、新たに設定する場合には、Add credential を参照してください。認証情報の作成に関する詳細は、「[Google Cloud Platform の認証情報の作成](#)」を参照してください。
5. クラスターの詳細を追加します。
  - a. 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。
  - b. クラスターセットを既存のクラスターセットに追加する場合は、クラスターセットを1つ指定します。クラスターの作成時に cluster-admin 権限がない場合に、クラスターを作成する clusterset-admin パーミッションがあるクラスターセットを選択する必要があります。クラスターセットを選択しない場合には、クラスターの作成には失敗します。選択するクラスターセットがない場合には、クラスター管理者に連絡して、クラスターセットへの clusterset-admin パーミッションを受け取ってください。
  - c. GCP アカウントに設定した Base DNS domain 情報を指定します。選択した認証情報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。この名前はクラスターのホスト名で使用されます。詳細は、[Setting up a custom domain](#)を参照してください。
  - d. 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。
  - e. 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。
6. マスタープールの詳細を入力します。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の管理を共有します。オプションの情報には以下のフィールドが含まれます。
  - a. マスタープールを実行する リージョンを指定します。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。
  - b. マスターノードのインスタンスタイプを指定します。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。デフォルト値は、n1-standard-1 - n1-standard-11 vCPU - General Purpose です。
7. ワーカープールに関する情報を入力します。ワーカープールにワーカーノードを作成し、クラスターのコンテナ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。オプションの情報には以下のフィールドが含まれます。
  - a. ワーカープールの名前を追加します。

- b. ワーカープールのインスタンスタイプを指定します。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。デフォルト値は、n1-standard-4 - 4 vCPU 15 GiB RAM - General Purpose です。
  - c. ワーカープールの Node count を入力します。ワーカープールを定義する場合にこの設定は必須です。  
Add worker pool をクリックして追加のワーカープールを定義できます。
8. クラスターのネットワーク情報を入力します。この情報は必須です。IPv6 を使用するには、複数のネットワークが必要です。
- a. ネットワークタイプを選択します。デフォルト値は OpenShiftSDN です。IPv6 を使用するには、OVNKubernetes の設定は必須です。
  - b. Cluster network CIDR を選択します。これは、Pod IP アドレスに使用できる IP アドレスの数およびリストです。この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 10.128.0.0/14 です。
  - c. ネットワークホストの接頭辞を指定します。これにより、各ノードにサブネットプレフィックスの長さを設定します。デフォルト値は 23 です。
  - d. サービスネットワーク CIDR でサービスの IP アドレスのブロックを指定します。この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 172.30.0.0/16 です。
  - e. マシン CIDR で OpenShift Container Platform ホストで使用される IP アドレスのブロックを指定します。この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。  
Add network をクリックして、追加のネットワークを追加できます。IPv6 アドレスを使用している場合は、複数のネットワークが必要です。
9. クラスターのインストールまたはアップグレードの前後に実行する Ansible Automation template (任意) を追加します。  
Add automation template をクリックしてテンプレートを作成できます。
10. 情報を確認し、必要に応じてカスタマイズします。
- a. YAML スライダーを On にクリックし、パネルに install-config.yaml ファイルの内容を表示します。
  - b. YAML ファイルをカスタム設定で編集します。
11. Create を選択してクラスターを作成します。  
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。

### 7.4.3. 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 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、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。

## 7.5. 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 クラスターをデプロイできます。

- [前提条件](#)
- [コンソールを使用したクラスターの作成](#)
- [クラスターへのアクセス](#)

### 7.5.1. 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 4.6 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 認証情報。詳細は、「[VMware vSphere での認証情報の作成](#)」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。
- 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。

### 7.5.2. コンソールを使用したクラスターの作成

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 Infrastructure > Clusters に移動します。
2. Clusters ページで、Create cluster をクリックします。  
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。

3. インフラストラクチャプロバイダーに Kubernetes ディストリビューションおよび VMware vSphere を選択します。残りの手順は、選択した内容により異なります。
4. 一覧で利用可能な認証情報から、お使いのインフラストラクチャプロバイダーの認証情報を選択します。設定されていない場合や、新たに設定する場合には、Add credential を参照してください。認証情報の作成に関する詳細は、「[VMware vSphere の認証情報の作成](#)」を参照してください。
5. クラスターの詳細を追加します。
  - a. クラスターの名前を入力します。値は、認証情報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。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。
  - b. クラスターセットを既存のクラスターセットに追加する場合は、クラスターセットを1つ指定します。クラスターの作成時に cluster-admin 権限がない場合に、クラスターを作成する clusterset-admin パーミッションがあるクラスターセットを選択する必要があります。クラスターセットを選択しない場合には、クラスターの作成には失敗します。選択するクラスターセットがない場合には、クラスター管理者に連絡して、クラスターセットへの clusterset-admin パーミッションを受け取ってください。
  - c. VMware vSphere アカウントに設定した Base DNS domain 情報を指定します。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。選択した認証情報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。
  - d. 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。  
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。
  - e. 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。
6. マスタープールの詳細を入力します。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の管理を共有します。必要な情報には以下のフィールドが含まれます。
  - a. クラスターに割り当てる ソケットごとにコアを1つ以上指定します。
  - b. マスターノードに割り当てる CPU の数を指定します。
  - c. 割り当てる メモリーサイズを MB 単位で指定します。
  - d. マスターノード用に作成する ディスクサイズを追加します。
7. ワーカープールに関する情報を入力します。ワーカープールにワーカーノードを作成し、クラスターのコンテナ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。情報には以下のフィールドが含まれます。

- a. ワーカープールの名前を追加します。
  - b. クラスターに割り当てるソケットごとにコアを1つ以上指定します。
  - c. 割り当てるCPUの数を指定します。
  - d. 割り当てるメモリーサイズをMB単位で指定します。
  - e. 作成するディスクのサイズ (GiB 単位) を追加します。
  - f. ノード数を追加し、クラスター内のワーカーノードの数を指定します。  
Add worker pool をクリックして追加のワーカープールを定義できます。
8. クラスターネットワークオプションを設定します。この情報は必須です。IPv6 を使用するには、複数のネットワークが必要です。
- 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。
- Add network をクリックして、追加のネットワークを追加できます。IPv6 アドレスを使用している場合は、複数のネットワークが必要です。
9. クラスターのインストールまたはアップグレードの前後に実行する Ansible Automation template (任意) を追加します。  
Add automation template をクリックしてテンプレートを作成できます。
10. 情報を確認し、必要に応じてカスタマイズします。
- a. YAML スライダーを On にクリックし、パネルに `install-config.yaml` ファイルの内容を表示します。
  - b. YAML ファイルをカスタム設定で編集します。
11. Create を選択してクラスターを作成します。  
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。

### 7.5.3. 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. Red Hat Advanced Cluster Management ナビゲーションメニューで 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。クラスターへのログイン時にこの値を使用します。
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。

## 7.6. RED HAT OPENSTACK PLATFORM 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、Red Hat OpenStack Platform で Red Hat OpenShift Container Platform クラスターをデプロイできます。

- [前提条件](#)
- [コンソールを使用したクラスターの作成](#)
- [クラスターへのアクセス](#)

### 7.6.1. 前提条件

Red Hat OpenStack Platform 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 4.6 以降に Red Hat Advanced Cluster Management ハブクラスターをデプロイしておく。
- Red Hat OpenStack Platform 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。
- Red Hat OpenStack Platform の認証情報。詳細は、「[Red Hat OpenStack Platform の認証情報の作成](#)」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。
- デプロイする Red Hat OpenStack Platform インスタンスについての以下の情報。
  - マスターおよびワーカーインスタンスのフレーバー名。(例: m1.xlarge)。
  - Floating IP アドレスを提供する外部ネットワークのネットワーク名。
  - 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>`。API の Floating IP アドレスを参照する必要があります。
    - `*.apps.<cluster_name>.<base_domain>`。Ingress の Floating IP アドレスを参照する必要があります。

### 7.6.2. コンソールを使用したクラスターの作成

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 Infrastructure > Clusters に移動します。
2. Clusters ページで、Create cluster をクリックします。  
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。
3. インフラストラクチャプロバイダーに Kubernetes ディストリビューションおよび Red Hat OpenStack を選択します。残りの手順は、選択した内容により異なります。
4. 一覧で利用可能な認証情報から、お使いのインフラストラクチャプロバイダーの認証情報を選択します。設定されていない場合や、新たに設定する場合には、Add credential を参照してください。認証情報の作成に関する詳細は、「[Red Hat OpenStack Platform の認証情報の作成](#)」を参照してください。

5. クラスターの詳細を追加します。

- a. 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。名前には 15 文字以上指定できません。  
注記:: 値は、認証情報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。

- b. クラスターセットを既存のクラスターセットに追加する場合は、クラスターセットを 1 つ指定します。クラスターの作成時に cluster-admin 権限がない場合に、クラスターを作成する clusterset-admin パーミッションがあるクラスターセットを選択する必要があります。クラスターセットを選択しない場合には、クラスターの作成には失敗します。選択するクラスターセットがない場合には、クラスター管理者に連絡して、クラスターセットへの clusterset-admin パーミッションを受け取ってください。
  - c. Red Hat OpenStack Platform アカウントに設定した Base DNS domain 情報を指定します。選択した認証情報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。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。詳細は、Red Hat OpenStack Platform ドキュメントの「[ドメインの管理](#)」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。
  - d. 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。  
注記: OpenShift Container Platform バージョン 4.6.x 以降のリリースイメージのみがサポートされます。
  - e. 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。
6. マスターノードの詳細を入力します。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の管理を共有します。オプションの情報には以下のフィールドが含まれます。

- a. マスターノードのインスタンスタイプを指定します。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。デフォルト値は m1.xlarge です。
7. ワーカープールに関する情報を入力します。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。オプションの情報には以下のフィールドが含まれます。
    - a. ワーカープールの名前を追加します。
    - b. ワーカープールのインスタンスタイプを指定します。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。デフォルト値は m1.xlarge です。
    - c. ワーカープールの Node count を入力します。ワーカープールを定義する場合にこの設定は必須です。  
Add worker pool をクリックして追加のワーカープールを定義できます。
  8. クラスターのネットワーク情報を入力します。IPv4 ネットワーク用に1つ以上のネットワークの値を指定する必要があります。IPv6 ネットワークの場合は、複数のネットワークを定義する必要があります。
    - a. External network name に Red Hat OpenStack Platform の外部ネットワーク名を追加します。
    - b. API Floating IP アドレスを追加します。既存の Floating IP アドレスは、OpenShift Container Platform API の外部ネットワーク用です。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。
    - c. Ingress Floating IP アドレスを追加します。既存の Floating IP アドレスは、Ingress ポートの外部ネットワーク上にあります。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。既存の Floating IP アドレスは、Ingress ポートの外部ネットワーク上にあります。
    - d. プライベートネットワークの名前解決をサポートする External DNS IP addresses を追加します。
    - e. デプロイする Pod ネットワークプロバイダープラグインを指定する ネットワークタイプ を指定します。使用できる値は OVNKubernetes または OpenShiftSDN です。デフォルト値は OpenShiftSDN です。IPv6 を使用するには、OVNKubernetes の設定は必須です。
    - f. Cluster network CIDR を選択します。クラスターで使用するために予約される IP アドレスのグループ。クラスターに十分な数値を指定し、その範囲に他のクラスターの IP アドレスを追加しないようにします。デフォルト値は 10.128.0.0/14 です。
    - g. Network host prefix を指定して、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を設定します。デフォルト値は 23 です。
    - h. サービスネットワーク CIDR でサービスの IP アドレスのブロックを指定します。この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 172.30.0.0/16 です。
    - i. マシン CIDR で OpenShift Container Platform ホストで使用される IP アドレスのブロックを指定します。この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。  
Add network をクリックして、追加のネットワークを追加できます。IPv6 アドレスを使用している場合は、複数のネットワークが必要です。

9. クラスターのインストールまたはアップグレードの前後に実行する Ansible Automation template (任意) を追加します。  
Add automation template をクリックしてテンプレートを作成できます。
10. 情報を確認し、必要に応じてカスタマイズします。
  - a. YAML スライダーを On にクリックし、パネルに install-config.yaml ファイルの内容を表示します。
  - b. YAML ファイルをカスタム設定で編集します。
11. Create を選択してクラスターを作成します。  
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。

### 7.6.3. 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. Red Hat Advanced Cluster Management ナビゲーションメニューで 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。クラスターへのログイン時にこの値を使用します。
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。

## 7.7. 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。

- [前提条件](#)
- [コンソールを使用したクラスターの作成](#)
- [クラスターへのアクセス](#)

### 7.7.1. 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 4.6 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、[プロビジョナーホストの準備](#) を参照してください。
- デプロイされた Red Hat Advanced Cluster Management for Kubernetes ハブクラスターが、プロビジョニングネットワークにルーティングできる。
- 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](#) について参照してください。
- 構成済みのベアメタルクレデンシャル。詳細は [ベアメタルのクレデンシャルの作成](#) 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。

### 7.7.2. コンソールを使用したクラスターの作成

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 Infrastructure > Clusters に移動します。
2. Clusters ページで、Create cluster をクリックします。  
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。
3. インフラストラクチャプロバイダーに Kubernetes ディストリビューションおよび Bare Metal を選択します。残りの手順は、選択した内容により異なります。
4. 一覧で利用可能な認証情報からお使いの認証情報を選択します。設定されていない場合や、新たに設定する場合には、Add credential を参照してください。認証情報の作成に関する詳細は、「[ベアメタルの認証情報の作成](#)」を参照してください。
5. クラスターの詳細を追加します。
  - a. 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。

- b. クラスターセットを既存のクラスターセットに追加する場合は、クラスターセットを1つ指定します。クラスターの作成時に `cluster-admin` 権限がない場合に、クラスターを作成する `clusterset-admin` パーミッションがあるクラスターセットを選択する必要があります。クラスターセットを選択しない場合には、クラスターの作成には失敗します。選択するクラスターセットがない場合には、クラスター管理者に連絡して、クラスターセットへの `clusterset-admin` パーミッションを受け取ってください。
  - c. ベアメタルプロバイダーアカウントに設定した Base DNS domain 情報を指定します。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。選択した認証情報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が、この設定はクラスターの作成後には変更できません。詳細は、OpenShift Container Platform ドキュメントの「[ベアメタルへのインストール](#)」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。
  - d. 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。
  - e. 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。
6. 認証情報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ベアメタルアセットを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 `Import assets` を選択して作成またはインポートを行うことができます。ベアメタルアセットの詳細は、[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。 `Disable certificate verification` を選択して要件を無視することができます。
  7. クラスターネットワークオプションを設定します。以下の表では、ネットワークオプションとその説明をまとめています。

パラメーター	説明	必須またはオプション
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

パラメーター	説明	必須またはオプション
API VIP	内部 API 通信に使用する仮想 IP <b>papi.&lt;cluster_name&gt;.&lt;Base DNS domain&gt;</b> 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
Ingress VIP	Ingress トラフィックに使用する仮想 IP。 <b>*.apps.&lt;cluster_name&gt;.&lt;Base DNS domain&gt;</b> 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。	任意
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 <b>OpenShiftSDN</b> です。	必須
クラスタのネットワーク 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ネットワークをサポートします。複数のクラスタ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

パラメーター	説明	必須またはオプション
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
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

Add network をクリックして、追加のネットワークを追加できます。IPv6 アドレスを使用している場合は、複数のネットワークが必要です。

8. クラスターのインストールまたはアップグレードの前後に実行する Ansible Automation template (任意) を追加します。  
Add automation template をクリックしてテンプレートを作成できます。
9. 情報を確認し、必要に応じてカスタマイズします。
  - a. YAML スライダーを On にクリックし、パネルに install-config.yaml ファイルの内容を表示します。
  - b. YAML ファイルをカスタム設定で編集します。
10. Create を選択してクラスターを作成します。  
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。

### 7.7.3. 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 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。

## 7.8. 作成されたクラスターの休止 (テクノロジープレビュー)

Red Hat Advanced Cluster Management for Kubernetes を使用して作成されたクラスターを休止し、リソースを節約できます。休止状態のクラスターに必要なリソースは、実行中のものより少なくなるので、クラスターを休止状態にしたり、休止状態を解除したりすることで、プロバイダーのコストを削減できる可能性があります。この機能は、以下の環境の Red Hat Advanced Cluster Management で作成したクラスターにのみ該当します。

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform

### 7.8.1. コンソールを使用したクラスターの休止

Red Hat Advanced Cluster Management コンソールを使用して、Red Hat Advanced Cluster Management で作成したクラスターを休止するには、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 Infrastructure > Clusters に移動します。Manage clusters タブが選択されていることを確認します。
2. 休止するクラスターを特定します。
3. そのクラスターの Options メニューから Hibernate cluster を選択します。注記: Hibernate cluster オプションが利用できない場合には、クラスターを休止状態にすることはできません。これは、クラスターがインポートされており、Red Hat Advanced Cluster Management で作成されていない場合に、発生する可能性があります。
4. 確認ダイアログボックスの Hibernate を選択して、クラスターを休止します。

Clusters ページのクラスターのステータスは、プロセスが完了すると Hibernating になります。

ヒント: Clusters ページで休止するするクラスターを選択し、Actions > Hibernate cluster を選択して、複数のクラスターを休止できます。

選択したクラスターが休止状態になりました。

### 7.8.2. CLI を使用したクラスターの休止

CLI を使用して、Red Hat Advanced Cluster Management が作成したクラスターを休止するには、以下の手順を実行します。

1. 以下のコマンドを入力して、休止するクラスターの設定を編集します。

```
oc edit clusterdeployment <name-of-cluster> -n <namespace-of-cluster>
```

**name-of-cluster** は、休止するクラスター名に置き換えます。

**namespace-of-cluster** は、休止するクラスターの namespace に置き換えます。

2. **spec.powerState** の値は **Hibernating** に変更します。
3. 以下のコマンドを実行して、クラスターのステータスを表示します。

```
oc get clusterdeployment <name-of-cluster> -n <namespace-of-cluster> -o yaml
```

**name-of-cluster** は、休止するクラスター名に置き換えます。

**namespace-of-cluster** は、休止するクラスターの namespace に置き換えます。

クラスターを休止するプロセスが完了すると、クラスターのタイプの値は **type=Hibernating** になります。

選択したクラスターが休止状態になりました。

### 7.8.3. コンソールを使用して休止中のクラスターの通常操作を再開する手順

Red Hat Advanced Cluster Management コンソールを使用して、休止中のクラスターの通常操作を再開するには、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 Infrastructure > Clusters に移動します。Manage clusters タブが選択されていることを確認します。
2. 休止状態で、再開させるクラスターを見つけます。
3. クラスターの Options メニューから Resume cluster を選択します。
4. 確認ダイアログボックスの Resume を選択して、クラスターの機能を再開します。

プロセスを完了すると、Clusters ページのクラスターのステータスは **Ready** になります。

ヒント: Clusters ページで、再開するクラスターを選択し、Actions > Resume cluster の順に選択して、複数のクラスターを再開できます。

選択したクラスターで通常の操作が再開されました。

### 7.8.4. CLI を使用して休止中のクラスターの通常操作を再開する手順

CLI を使用して、休止中のクラスターの通常操作を再開するには、以下の手順を実行します。

1. 以下のコマンドを入力してクラスターの設定を編集します。

```
oc edit clusterdeployment <name-of-cluster> -n <namespace-of-cluster>
```

**name-of-cluster** は、休止するクラスター名に置き換えます。

**namespace-of-cluster** は、休止するクラスターの namespace に置き換えます。

2. **spec.powerState** の値を **Running** に変更します。
3. 以下のコマンドを実行して、クラスターのステータスを表示します。

```
oc get clusterdeployment <name-of-cluster> -n <namespace-of-cluster> -o yaml
```

**name-of-cluster** は、休止するクラスター名に置き換えます。

**namespace-of-cluster** は、休止するクラスターの namespace に置き換えます。

クラスターの再開プロセスが完了すると、クラスターのタイプの値は **type=Running** になります。

選択したクラスターで通常の操作が再開されました。

## 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 4.6 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 Container Platform によって作成されていないクラスターをインポートする場合は、`multiclusterhub.spec.imagePullSecret` を定義する必要があります。このシークレットは、Red Hat Advanced Cluster Management for Kubernetes のインストール時に作成されている場合もあります。シークレットの定義の詳細は [OperatorHub からのインストール](#) を参照してください。

- Red Hat OpenShift Dedicated 環境にインポートする場合：
  - 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から Infrastructure > Clusters を選択します。
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. 以下のオプションからインポートするクラスター特定に使用するインポートモードを選択します。
  - import コマンドを手動で実行する: 指定した情報に基づいてコピーして実行できるインポートコマンドを生成します。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。
    - a. 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。
    - b. 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。

- c. Red Hat OpenShift Dedicated 環境のみ対象: 以下の手順を実行します。
- i. 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。
  - ii. OpenShift Container Platform カタログで `klusterlet Operator` を検索します。
  - iii. 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。
  - iv. 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。
    - A. `import` コマンドを生成します。
      - I. Red Hat Advanced Cluster Management コンソールで、`Infrastructure > Clusters` を選択します。
      - II. `Add a cluster > Import an existing cluster` を選択します。
      - III. クラスター情報を追加し、`Save import and generate code` を選択します。
    - B. `import` コマンドをコピーします。
    - C. `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。
    - D. 以下のコマンドを実行して、新しいファイルにコンテンツを挿入します。

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

- ii. 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。
  - e. 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。
  - 既存のクラスターのサーバー URL および API トークンを入力する: インポートするクラスターのサーバー URL および API トークンを指定します。
  - kubeconfig: インポートしているクラスターの kubeconfig ファイルの内容をコピーし、貼り付けます。
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。

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](#)
- [クラスターのインポート](#)
- [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 4.6 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 4.6 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。
- Red Hat OpenShift Container Platform によって作成されていないクラスターをインポートする場合は、`multiclusterhub.spec.imagePullSecret` を定義する必要があります。このシークレットは、Red Hat Advanced Cluster Management for Kubernetes のインストール時に作成されている場合もあります。シークレットの定義の詳細は [OperatorHub からのインストール](#) を参照してください。

### 8.2.2. サポートされているアーキテクチャー

- Linux (x86\_64, s390x, ppc64le)
- 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}
```

「[Importing the cluster](#)」に進みます。

### 8.2.4. クラスターのインポート

以下のいずれかの手順を実行してクラスターをインポートします。

- 自動インポートを使用するには、インポートするクラスターの kubeconfig またはサーバーおよびトークンのペアのいずれかを含むシークレットを作成する必要があります。
  1. 以下のテンプレートのような内容を含む YAML ファイルを作成します。

```
apiVersion: v1
kind: Secret
metadata:
  name: auto-import-secret
  namespace: <cluster_name>
stringData:
  autoImportRetry: 5
  # If you are using the kubeconfig file, add the following value for the kubeconfig
  # file
  # that has the current context set to the cluster to import:
  kubeconfig: |- <kubeconfig_file>
  # If you are using the token/server pair, add the following two values:
  token: <Token to access the cluster>
  server: <cluster_api_url>
type: Opaque
```

2. ファイルを `auto-import-secret.yaml` として保存します。
3. 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 auto-import-secret.yaml
```

- 手動によるインポートプロセスを使用するには、以下の手順を行います。
  1. 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>
```

```
labels:
  cloud: auto-detect
  vendor: auto-detect
spec:
  hubAcceptsClient: true
```

**cloud** および **vendor** の値を **auto-detect** する場合、Red Hat Advanced Cluster Management はインポートしているクラスターからクラウドおよびベンダータイプを自動的に検出します。オプションで、**auto-detect** の値をクラスターのクラウドおよびベンダーの値に置き換えることができます。以下の例を参照してください。

```
cloud: Amazon
vendor: OpenShift
```

2. ファイルは **managed-cluster.yaml** として保存します。
3. 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
```

[klusterlet のインポート](#) を進めます。

### 8.2.5. klusterlet のインポート

以下の手順を実行して、klusterlet アドオン設定ファイルを作成および適用できます。

1. 以下の例のような YAML ファイルを作成します。

```
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: 2.3.0
```

2. ファイルは **klusterlet-addon-config.yaml** として保存します。
3. 以下のコマンドを実行して YAML を適用します。

■

```
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。

クラスターがインポートされました。

### 8.2.6. CLI でのインポートされたクラスターの削除

クラスターを削除するには、以下のコマンドを実行します。

```
oc delete managedcluster <cluster_name>
```

`cluster_name` は、クラスターの名前に置き換えます。

これでクラスターが削除されます。

## 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、`KlusterletAddonConfig` の設定を変更します。

`KlusterletAddonConfig` 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。以下で `KlusterletAddonConfig` を参照してください。

```
apiVersion: agent.open-cluster-management.io/v1
kind: KlusterletAddonConfig
metadata:
  name: <cluster-name>
  namespace: <cluster-name>
spec:
  clusterName: <cluster-name>
  clusterNamespace: <cluster-name>
  clusterLabels:
    cloud: auto-detect
    vendor: auto-detect
  applicationManager:
    enabled: true
  certPolicyController:
    enabled: true
  iamPolicyController:
    enabled: true
  policyController:
    enabled: true
  searchCollector:
    enabled: false
  version: 2.3.0
```

### 8.3.1. klusterlet アドオン設定の説明

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。

表8.1 klusterlet アドオン設定の表一覧

設定名	値	説明
applicationmanager	<b>true</b> または <b>false</b>	このコントローラーは、マネージドクラスターでアプリケーションのサブスクリプションライフサイクルを管理します。
certPolicyController	<b>true</b> または <b>false</b>	このコントローラーは、マネージドクラスターで証明書ベースのポリシーを有効にします。
iamPolicyController	<b>true</b> または <b>false</b>	このコントローラーは、マネージドクラスターで IAM ベースのポリシーライフサイクルを有効にします。
policyController	<b>true</b> または <b>false</b>	このコントローラーは、マネージドクラスターの他の全ポリシールールを有効にします。
searchCollector	<b>true</b> または <b>false</b>	このコントローラーを使用して、リソースインデックスデータをハブクラスターに定期的に戻します。

### 8.3.2. 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 コンソールにログインします。
2. ハブクラスターコンソールのヘッダーメニューから Search アイコンを選択します。
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。

### 8.3.3. 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. ハブクラスターにログインします。
2. 以下のコマンドを入力してリソースを編集します。

```
kubectrl 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 プロキシ環境でのクラスタの作成

ハブクラスタがプロキシサーバー経由で接続されている場合は、Red Hat OpenShift Container Platform クラスタを作成できます。

クラスタの作成を成功させるには、以下のいずれかの状況が true である必要があります。

- Red Hat Advanced Cluster Management for Kubernetes には、作成しているマネージドクラスタを使用したプライベートネットワーク接続がありますが、Red Hat Advanced Cluster Management およびマネージドクラスタは、プロキシを使用してインターネットにアクセスします。
- マネージドクラスタはインフラストラクチャプロバイダーにあります。ファイアウォールポートを使用することでマネージドクラスタからハブクラスタへの通信を有効にします。

プロキシで設定されたクラスタを作成するには、以下の手順を実行します。

1. 以下の情報を `install-config.yaml` ファイルに追加して、ハブクラスタに `cluster-wide-proxy` 設定を指定します。

```
apiVersion: v1
kind: Proxy
baseDomain: <domain>
proxy:
  httpProxy: http://<username>:<password>@<proxy.example.com>:<port>
  httpsProxy: https://<username>:<password>@<proxy.example.com>:<port>
  noProxy: <wildcard-of-domain>,<provisioning-network/CIDR>,<BMC-address-range/CIDR>
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  <trusted-ca-cert>
  -----END CERTIFICATE-----
```

`username` は、プロキシサーバーのユーザー名に置き換えます。

`password` は、プロキシサーバーへのアクセス時に使用するパスワードに置き換えます。

`proxy.example.com` は、プロキシサーバーのパスに置き換えます。

`port` は、プロキシサーバーとの通信ポートに置き換えます。

`wildcard-of-domain` は、プロキシをバイパスするドメインのエントリに置き換えます。

`provisioning-network/CIDR` は、プロビジョニングネットワークの IP アドレスと割り当てられた IP アドレスの数 (CIDR 表記) に置き換えます。

`BMC-address-range/CIDR` は、BMC アドレスおよびアドレス数 (CIDR 表記) に置き換えます。

`trusted-ca-cert` は、`openshift-config` namespace の ConfigMap からの `user-ca-bundle` の内容に置き換えます。`additionalTrustBundle` フィールドは、プロキシのアイデンティティ証明書が Red Hat Enterprise Linux Core OS 信頼バンドルからの認証局によって署名されない限り必要になります。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、MITM CA 証明書を指定する必要があります。

以前の値を追加すると、設定はクラスターに適用されます。

2. クラスターの作成手順を実行してクラスターをプロビジョニングします。「[クラスターの作成](#)」を参照してプロバイダーを選択します。

## 9.1. 既存クラスターでのクラスター全体のプロキシの有効化

クラスター全体のプロキシ機能を有効にする場合には、マネージドクラスターのコンポーネントを認識して、プロキシの背後で機能できるようにする必要があります。Operator Pod を環境変数 (HTTP\_PROXY, HTTPS\_PROXY, NO\_PROXY) で更新して、Application Manager などのコンポーネントを有効にします。

これらの環境変数は、Pod が HTTP および HTTPS トラフィックをプロキシサーバーにルーティングする場所を指定します。Operator Pod はこれらの環境変数で自動的に更新されるので、スタンドアロンサブスクリプションはクラスター全体のプロキシを有効にした後に機能します。

既存のマネージドクラスターでは、サブスクリプションコントローラーは `klusterlet` アドオン内で実行されます。環境変数は、Pod の作成時に自動的に追加されません。デプロイメントにパッチを適用し、デプロイメントの各コンテナに 3 つの環境変数を追加して、環境変数を既存のクラスターに手動で追加する必要があります。

以下の手順を実行して、`appmgr` の `klusterlet add-on` に環境変数を追加します。

1. 以下のコマンドを入力して HTTP プロキシを作成します。

```
kubectl patch deployment <deployment-name> -p '{"spec":{"template":{"spec":{"containers":[{"env":[{"name":"HTTP_PROXY","value":"<http-proxy>"}],"name":"<container-name>"}]}}}}'
```

`deployment-name` は、デプロイメント名に置き換えます。

`http-proxy` は、http 要求に使用するプロキシサーバーのアドレスに置き換えます。(例: <http://192.168.1.1:3128>)。

`container-name` は、コンテナの名前に置き換えます。

2. 以下のコマンドを入力して HTTPS プロキシを作成します。

```
kubectl patch deployment <deployment-name> -p '{"spec":{"template":{"spec":{"containers":[{"env":[{"name":"HTTPS_PROXY","value":"<https-proxy>"}],"name":"<container-name>"}]}}}}'
```

`deployment-name` は、デプロイメント名に置き換えます。

`https-proxy` は https 要求に使用するプロキシサーバーのアドレスに置き換えます。(例: <http://192.168.1.0:3128>)。

`container-name` は、コンテナの名前に置き換えます。

3. 以下のコマンドを実行して、プロキシを通過させないものを指定します。

```
kubectl patch deployment <deployment-name> -p '{"spec":{"template":{"spec":{"containers":[{"env":[{"name":"NO_PROXY","value":"<no-proxy>"}],"name":"<container-name>"}]}}}}'
```

`deployment-name` は、デプロイメント名に置き換えます。

`no-proxy` は、トラフィックがプロキシ経由でルーティングされない IP、ホスト名、およびドメインのコンマ区切りリストに置き換えます。(例: `.cluster.local,svc,10.128.0.0/14,example.com`)。

`container-name` は、コンテナの名前に置き換えます。

ヒント: 複数のコンテナでは、各コンテナで、以下の例のようなファイルを作成して実行できます。

```
DEPLOYMENT_NAME="klusterlet-addon-appmgr"
CONTAINER_NAME="subscription-controller"
HTTP_PROXY="http://192.168.1.1:3128"
HTTPS_PROXY="http://192.168.1.0:3128"
NO_PROXY=".cluster.local,svc,10.128.0.0/14, example.com"

kubectl patch deployment $DEPLOYMENT_NAME -p '{"spec":{"template":{"spec":{"containers":[{"env":[{"name":"HTTP_PROXY","value":"$HTTP_PROXY"}],"name":"$CONTAINER_NAME"}]}}}}'

kubectl patch deployment $DEPLOYMENT_NAME -p '{"spec":{"template":{"spec":{"containers":[{"env":[{"name":"HTTPS_PROXY","value":"$HTTPS_PROXY"}],"name":"$CONTAINER_NAME"}]}}}}'

kubectl patch deployment $DEPLOYMENT_NAME -p '{"spec":{"template":{"spec":{"containers":[{"env":[{"name":"NO_PROXY","value":"$NO_PROXY"}],"name":"$CONTAINER_NAME"}]}}}}'
```

注記: マネージドクラスターで登録エージェントおよびワークエージェントは、プロキシを通過できない mTLS 接続を確立してハブクラスターで `apiserver` と通信するため、プロキシ設定をサポートしません。

重要: グローバルプロキシ設定は、アラート転送には影響しません。クラスター全体のプロキシを使用して Red Hat Advanced Cluster Management ハブクラスターに [アラート転送を設定するには、「送信アラート」](#) を参照してください。

## 第10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
  resourceName:
  - <managed-cluster-name>
  verbs:
  - get
  - list
  - watch
  - update
  - delete
  - deletecollection
  - patch
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  verbs:
  - create
- apiGroups:
  - ""
  resources:
  - namespaces
  resourceName:
  - <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11章 クラスターラベルの管理

クラスターにラベルを追加してグループリソースを選択します。詳細は、「[Labels and Selectors](#)」を参照してください。

クラスターの新規ラベルの追加、既存ラベルの削除、既存ラベルの編集が可能です。ラベルを管理するには、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 `infrastructure > Clusters` をクリックします。
2. Clusters テーブルでクラスターを探します。
3. Options をクリックし、メニューのオプションを表示します。
4. Edit labels を選択します。
5. Edit labels ダイアログボックスで、ラベルを入力します。入力内容は `Key=Value` のようになります。
6. Enter を押してコンマを追加するか、スペースを追加して別のラベルを追加します。
7. 最後の項目を入力したら、Save をクリックします。Save をクリックしてからでないと、ラベルは保存されません。
8. 完了したら、ラベルの編集ペインを閉じます。
  - 既存のラベルを削除するには、一覧から削除するラベルの Remove アイコンをクリックします。
  - 既存のラベルを更新する場合は、同じキーに別の値を使用して、新規ラベルを追加すると、新しい値にキーを再割り当てすることができます。たとえば、`Key=Value` を変更するには、`Key=NewValue` を入力して `Key` の値を更新します。

ヒント: クラスターの詳細ページからクラスターラベルを編集することもできます。ナビゲーションメニューから `Infrastructure > Clusters` をクリックします。Clusters ページで、クラスターの名前をクリックしてクラスターの詳細ページにアクセスします。Labels セクションの Edit アイコンをクリックします。Edit labels ダイアログボックスが表示されます。

## 第12章 マネージドクラスターで実行する ANSIBLE TOWER タスクの設定

Red Hat Advanced Cluster Management は、Ansible Tower 自動化と統合し、クラスターの作成またはアップグレード前後に、prehook および posthook AnsibleJob インスタンスを作成できるようにします。クラスター破棄の prehook および posthook ジョブの設定やクラスターのスケールアクションはサポートされません。

必要なアクセス権限: クラスターの管理者

- [前提条件](#)
- [コンソールを使用したクラスターでの実行用の AnsibleJob テンプレート設定](#)
- [AnsibleJob テンプレートの作成](#)
- [ラベルを使用したマネージドクラスターでの実行用の AnsibleJob テンプレート設定](#)
- [Ansible ジョブのステータスの表示](#)

### 12.1. 前提条件

Red Hat Advanced Cluster Management クラスターで Ansible テンプレートを実行するには、以下の前提条件を満たす必要があります。

- OpenShift Container Platform 4.6 以降
- Ansible Tower バージョン 3.7.3 以降がインストールされていること。Ansible Tower の最新のサポートバージョンをインストールすることがベストプラクティスです。詳細は、[Red Hat AnsibleTower ドキュメント](#) を参照してください。
- Ansible Automation Platform Resource Operator をインストールして、Ansible ジョブを Git サブスクリプションのライフサイクルに接続しておくこと。AnsibleJob を使用した Ansible Tower ジョブの実行時に最善の結果を得るには、実行時に Ansible Tower ジョブテンプレートが幕等でなければなりません。Ansible Automation Platform Resource Operator は、Red Hat OpenShift Container Platform OperatorHub ページから検索できます。

Ansible Tower 自動化のインストールおよび設定に関する詳細は、「[Ansible タスクの設定 \(テクノロジープレビュー\)](#)」を参照してください。

### 12.2. コンソールでを使用したクラスターでの実行用の ANSIBLEJOB テンプレート設定

クラスターの作成時にクラスターに使用する Ansible ジョブテンプレートを指定する必要があります。クラスターの作成時にテンプレートを指定するには、Automation の手順でクラスターに適用する Ansible テンプレートを選択します。Ansible テンプレートがない場合は、Add automation template をクリックして作成します。

### 12.3. ANSIBLEJOB テンプレートの作成

クラスターのインストールまたはアップグレードで Ansible ジョブを開始するには、ジョブ実行のタイミングを指定する Ansible ジョブテンプレートを作成する必要があります。これらは、クラスターのインストールまたはアップグレード前後に実行するように設定できます。

テンプレートの作成時に Ansible テンプレートの実行に関する詳細を指定するには、以下の手順を実行します。

1. Red Hat Advanced Cluster Management ナビゲーションで Infrastructure > Automation に移動します。既存の Ansible テンプレートが表示されます。
2. 状況に適したパスを選択します。
  - 新規テンプレートを作成する場合には、Create Ansible template をクリックして手順 3 に進みます。
  - 既存のテンプレートを変更する場合は、変更するテンプレートの Options メニューの Edit template をクリックして、手順 5 に進みます。
3. テンプレートの名前を入力します。小文字、英数字、ハイフン (-) が含まれる一意の名前を指定する必要があります。
4. 新規テンプレートの認証情報を選択します。Ansible 認証情報を Ansible テンプレートにリンクするには、以下の手順を実行します。
  - a. Red Hat Advanced Cluster Management ナビゲーションで Automation を選択します。認証情報にリンクされていないテンプレートの一覧内のテンプレートには、テンプレートを既存の認証情報にリンクするために使用できるリンク「認証情報へのリンク」アイコンが含まれています。
  - b. Link to credential を選択して、既存の認証情報を検索し、リンクします。
  - c. Ansible Automation Platform credential フィールドにあるメニューから、利用可能な認証情報を選択します。テンプレートと同じ namespace の認証情報のみが表示されます。
  - d. 選択できる認証情報がない場合や、既存の認証情報を使用しない場合は、リンクするテンプレートの Options メニューから Edit template を選択します。
  - e. Add credential をクリックして、テンプレートの認証情報を作成します。
  - f. 認証情報を作成する場合は、「[Ansible Automation Platform の認証情報の作成](#)」の手順を行います。
  - g. テンプレートと同じ namespace に認証情報を作成したら、テンプレートの編集時に Ansible Automation Platform credential フィールドで認証情報を選択します。
  - h. Save をクリックして、認証情報とテンプレート間のリンクをコミットします。
5. クラスターをインストールする前に Ansible ジョブを開始する場合は、Pre-install Ansible job templates セクションで Add an Ansible job template を選択します。
6. prehook および posthook Ansible ジョブを選択して、クラスターのインストールまたはアップグレードに追加します。

注記: Ansible job template name は、Ansible Tower の Ansible ジョブの名前と一致している必要があります。
7. Ansible Tower ジョブで必要な追加変数 (ある場合) を入力します。
8. 必要に応じて、Ansible ジョブをドラッグして、順番を変更します。
9. Save をクリックして情報をコミットします。

10. Post-install Ansible job templates セクションでクラスターのインストールを完了したら、開始する Ansible ジョブテンプレートごとに手順 5-9 を繰り返します。
11. Save をクリックして情報をコミットします。
12. Next をクリックし、アップグレードの開始に使用する Ansible ジョブの指定を開始します。
13. Pre-install Ansible job templates セクションでクラスターのアップグレード前に、開始する Ansible ジョブテンプレートごとに手順 5-9 を繰り返します。
14. Save をクリックして情報をコミットします。
15. Post-install Ansible job templates セクションでクラスターのアップグレード後に、開始する Ansible ジョブテンプレートごとに手順 5-9 を繰り返します。
16. Save をクリックして情報をコミットします。
17. Next をクリックして、追加した Ansible ジョブを確認します。
18. Add を選択して、Ansible ジョブの設定情報をテンプレートに追加します。

Ansible テンプレートは、指定のアクションが起こるタイミングで、このテンプレートを指定するクラスターで実行するように設定されます。

## 12.4. ラベルを使用したマネージドクラスターでの実行用の ANSIBLEJOB テンプレート設定

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 で管理するためにラベルを使用して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ンネルのいずれかで、Ansible ジョブの定義ファイルを作成します。Git チャン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。

- Red Hat Advanced Cluster Management ハブクラスターにログインします。
- 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ンネルを参照する 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がベストプラクティスです。

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。

- 「[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。

- Git チャンネルの 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。

## 12.5. ANSIBLE ジョブのステータスの表示

実行中の Ansible ジョブのステータスを表示して、起動し、正常に実行されていることを確認できます。実行中の Ansible ジョブの現在のステータスを表示するには、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 Infrastructure > Clusters に移動して、Clusters ページにアクセスします。
2. クラスターの名前を選択して、その詳細を表示します。
3. クラスター情報で Ansible ジョブの最後の実行ステータスを表示します。エントリーには、以下のステータスの1つが表示されます。
  - インストール prehook または posthook ジョブが失敗すると、クラスターのステータスは **Failed** と表示されます。
  - アップグレード prehook または posthook ジョブが失敗すると、アップグレードに失敗した Distribution フィールドに警告が表示されます。  
ヒント: クラスターの prehook または posthook が失敗した場合は、Clusters ページからアップグレードを再試行できます。

## 第13章 MANAGEDCLUSTERSETS の管理 (テクノロジープレビュー)

**ManagedClusterSet** は、マネージドクラスタのグループです。マネージドクラスタセットでは、グループ内の全マネージドクラスタ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。

- **ManagedClusterSet の作成**
  - [コンソールを使用した ManagedClusterSet の作成](#)
  - [コマンドラインを使用した ManagedClusterSet の作成](#)
- **クラスタの ManagedClusterSet への追加**
  - [コンソールを使用したクラスタの ManagedClusterSet への追加](#)
  - [コマンドラインを使用した ManagedClusterSet へのクラスタの追加](#)
- **ManagedClusterSet からのマネージドクラスタの削除**
  - [コンソールを使用した ManagedClusterSet からのマネージドクラスタの削除](#)
  - [コマンドラインを使用した ManagedClusterSet からのクラスタの削除](#)
- **ManagedClusterSetBinding リソースの作成**
  - [コンソールを使用した ManagedClusterSetBinding の作成](#)
  - [コマンドラインを使用した ManagedClusterSetBinding の作成](#)
- **ManagedClusterSet に対するユーザーまたはグループのロールベースのアクセス制御パーミッションの割り当て**

### 13.1. MANAGEDCLUSTERSET の作成

マネージドクラスタセットにマネージドクラスタをグループ化して、マネージドクラスタでのユーザーのアクセス権を制限できます。

必要なアクセス権限:: クラスタの管理者

**ManagedClusterSet** は、クラスタ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で管理者権限が必要です。マネージドクラスタは、複数の **ManagedClusterSet** に追加できません。Red Hat Advanced Cluster Management for Kubernetes コンソールまたはコマンドラインインターフェースから、マネージドクラスタセットを作成できます。

#### 13.1.1. コンソールを使用した ManagedClusterSet の作成

Red Hat Advanced Cluster Management コンソールを使用して、マネージドクラスタセットを作成するには、以下の手順を実行します。

1. メインコンソールナビゲーションで、Infrastructure > Clusters を選択します。
2. Cluster sets タブを選択します。

3. Create cluster setを選択します。
4. 作成するクラスターセットの名前を入力します。
5. Create を選択して、クラスターセットリソースを作成します。

### 13.1.2. コマンドラインを使用した ManagedClusterSet の作成

コマンドラインを使用して yaml ファイルに以下のマネージドクラスターセットの定義を追加し、マネージドクラスターセットを作成します。

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

`clusterset1` はマネージドクラスターセットの名前に置き換えます。

## 13.2. 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。コンソールまたはコマンドラインのいずれかを使用して、マネージドクラスターをマネージドクラスターセットに追加できます。

### 13.2.1. コンソールを使用したクラスターの ManagedClusterSet への追加

Red Hat Advanced Cluster Management コンソールを使用してマネージドクラスターセットにクラスターを追加するには、以下の手順を実行します。

1. マネージドクラスターセットを作成している場合は、Manage resource assignments を選択して、Manage resource assignments ページに直接移動します。この手順のステップ 6 に進みます。
2. クラスターがすでに存在する場合は、メインのナビゲーションで Infrastructure > Clusters を選択してクラスターページにアクセスします。
3. Cluster sets タブを選択して、利用可能なクラスターセットを表示します。
4. マネージドクラスターセットに追加するクラスターセットの名前を選択し、クラスターセットの詳細を表示します。
5. Actions > Manage resource assignments を選択します。
6. Manage resource assignments ページで、クラスターセットに追加するリソースのチェックボックスを選択します。
7. Review を選択して変更を確認します。
8. Save を選択して変更を保存します。  
注記: マネージドクラスターを別のマネージドクラスターセットに移動する場合には、マネージドクラスター両方で RBAC パーミッションの設定が必要です。

### 13.2.2. コマンドラインを使用した ManagedClusterSet へのクラスターの追加

コマンドラインを使用してクラスターをマネージドクラスターに追加するには、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、マネージドクラスターセット両方で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` のマネージドクラスターセット名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マネージドクラスターセット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マネージドクラスターセット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。

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

マネージドクラスターセットからマネージドクラスターを削除して別のマネージドクラスターセットに移動するか、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。コンソールまたはコマンドラインインターフェースを使用して、マネージドクラスターセットからマネージドクラスターを削除できます。

#### 13.3.1. コンソールを使用した ManagedClusterSet からのマネージドクラスターの削除

Red Hat Advanced Cluster Management コンソールを使用して、マネージドクラスターセットからクラスターを削除するには、以下の手順を実行します。

1. マネージドクラスターセットを作成している場合は、Manage resource assignments を選択して、Manage resource assignments ページに直接移動します。この手順のステップ 6 に進みます。
2. クラスターがすでに存在する場合は、メインのナビゲーションで Infrastructure > Clusters を選択してクラスターページにアクセスします。
3. Cluster sets タブを選択して、利用可能なクラスターセットを表示します。
4. マネージドクラスターセットから削除するクラスターセットの名前を選択し、クラスターセットの詳細を表示します。
5. Actions > Manage resource assignments を選択します。
6. Manage resource assignments ページで、クラスターセットから削除するリソースのチェックボックスを選択します。  
この手順では、クラスターセットのメンバーであるリソースを削除するか、クラスターセットのメンバーでないリソースを追加します。マネージドクラスターの詳細を表示して、リソースがすでにクラスターセットのメンバーであるかどうかを確認できます。
7. Review を選択して変更を確認し、Save を選択します。

注記: マネージドクラスターを別のマネージドクラスターセットに移動する場合には、マネージドクラスターセット両方で RBAC パーミッションの設定が必要です。

#### 13.3.2. コマンドラインを使用した ManagedClusterSet からのクラスターの削除

マネージドクラスターセット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、マネージドクラスターセットでマネージドクラスターのリストを表示します。

```
oc get managedclusters -l cluster.open-cluster-management.io/clusterSet=<clusterset1>
```

`clusterset1` はマネージドクラスターセット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マネージドクラスターセットに移動する場合には、マネージドクラスターセット両方で RBAC パーミッションの設定が必要です。

## 13.4. MANAGEDCLUSTERSETBINDING リソースの作成

**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、バインドされたマネージドクラスターセットリソースに含まれるマネージドクラスターだけです。

namespace へのアクセスパーミッションは、その namespace にバインドされるマネージドクラスターセットに自動的に適用されます。マネージドクラスターセットがバインドされる namespace へのアクセス権限がある場合は、その namespace にバインドされているマネージドクラスターセットにアクセス権限が自動的に付与されます。ただし、マネージドクラスターセットへのアクセス権限のみがある場合は、対象の namespace にある他のマネージドクラスターセットにアクセスする権限は自動的に割り当てられません。マネージドクラスターセットが表示されない場合は、表示に必要なパーミッションがない可能性があります。

コンソールまたはコマンドラインを使用してマネージドクラスターセットバインディングを作成できます。

### 13.4.1. コンソールを使用した ManagedClusterSetBinding の作成

Red Hat Advanced Cluster Management コンソールを使用して、マネージドクラスターセットからクラスターを削除するには、以下の手順を実行します。

1. メインのナビゲーションで Infrastructure > Clusters を選択して、クラスターページにアクセスします。
2. Cluster sets タブを選択して、利用可能なクラスターセットを表示します。
3. バインディングを作成するクラスターセットの名前を選択し、クラスターセットの詳細を表示します。
4. Actions > Edit namespace bindings を選択します。
5. Edit namespace bindings ページで、ドロップダウンメニューからクラスターセットをバインドする namespace を選択します。このクラスターセットにバインディングされている既存の namespace は選択してあります。
6. Save を選択して変更を送信します。

### 13.4.2. コマンドラインを使用した ManagedClusterSetBinding の作成

コマンドラインを使用してマネージドクラスターセットバインディングを作成するには、以下の手順を実行します。

1. yaml ファイルに **ManagedClusterSetBinding** リソースを作成します。クラスターセットバインディングの作成時には、クラスターセットバインディング名と、バインドするマネージドクラスターセット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
```

2. ターゲットのマネージドクラスターセット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"]
```

## 13.5. MANAGEDCLUSTERSET に対するユーザーまたはグループのロールベースのアクセス制御パーミッションの割り当て

ハブクラスターに設定したアイデンティティプロバイダーが提供するクラスターセットに、ユーザーまたはグループを割り当てることができます。

必要なアクセス権限: クラスターの管理者

ManagedClusterSet API が提供する RBAC パーミッションにはレベルが2つあります。

- クラスターセット admin
  - マネージドクラスターセットに割り当てられた、すべてのクラスターおよびクラスタープールリソースに対する完全なアクセス権限。
  - クラスターの作成、クラスターのインポート、クラスタープールの作成権限。パーミッションは、マネージドクラスターセットの作成時に設定したマネージドクラスターに割り当てする必要があります。
- クラスターセット view
  - マネージドクラスターセットに割り当てられた、すべてのクラスターおよびクラスタープールリソースに対する読み取り専用の権限。
  - クラスターの作成、クラスターのインポート、またはクラスタープールの作成を実行する権限なし。

Red Hat Advanced Cluster Management コンソールからマネージドクラスターセットにユーザーまたはグループを割り当てするには、以下の手順を実行します。

1. コンソールのメインナビゲーションメニューで、Infrastructure > Clusters を選択します。
2. Cluster sets タブを選択します。
3. ターゲットクラスターセットを選択します。

4. Access management タブを選択します。
5. Add user or group を選択します。
6. アクセス権を割り当てるユーザーまたはグループを検索して選択します。
7. Cluster set admin または Cluster set view ロールを選択して、選択したユーザーまたはグループに付与します。ロールのパーミッションについての詳細は、「[ロールの概要](#)」を参照してください。
8. Add を選択して変更を送信します。

テーブルにユーザーまたはグループが表示されます。全マネージドクラスターセットリソースのパーミッションの割り当てがユーザーまたはグループに伝播されるまでに数秒かかる場合があります。

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。

配置の詳細については、「[Placement での ManagedClusterSets の使用](#)」を参照してください。

## 13.6. PLACEMENT での MANAGEDCLUSTERSETS の使用

Placement リソースは、namespace レベルのリソースで、placement namespace にバインドされる ManagedClusterSets から ManagedClusters セットを選択するルールを定義します。

必要なアクセス権限: クラスター管理者またはクラスターセット管理者。

### 13.6.1. 配置の概要

マネージドクラスターによる配置の仕組みについては、以下を参照してください。

- Kubernetes クラスターは、cluster スコープの ManagedClusters としてハブクラスターに登録されます。
- managedcluster は、クラスタースコープの ManagedClusterSets に編成されます。
- ManagedClusterSets はワークロード namespace にバインドされます。
- namespace スコープの Placements は、ManagedClusterSets の一部を指定して、ManagedClusters 候補の作業セットを選択します。
- Placements は、ラベルと要求セレクターを使用して作業セットから選択します。  
重要: Placement では、placement namespace に ManagedClusterSet がバインドされていない場合には ManagedCluster は選択されません。

Placement 仕様には以下のフィールドが含まれます。

- ClusterSets は、ManagedClusters の選択元の ManagedClusterSets を表します。
  - 指定されていない場合には、Placement namespace にバインドされる ManagedClusterSets から ManagedClusters が選択されます。
  - 指定されている場合は、ManagedClusters がこのセットの交差部分から選択され、ManagedClusterSets は Placement namespace にバインドされます。
- NumberOfClusters は、配置要件を満たす ManagedClusters の中から選択する数を表します。

指定されていない場合、配置要件を満たすすべての **ManagedClusters** が選択されます。

- 述語は、ラベルおよび要求セクターで **ManagedClusters** を選択する述語のスライスを表します。述語は ORed です。

### 13.6.2. 配置の例

namespace に **ManagedClusterSetBinding** を作成して、その namespace に最低でも1つ **ManagedClusterSet** をバインドする必要があります。注記: `managedclustersets/bind` の仮想サブリソースの **CREATE** に対してロールベースのアクセスが必要です。以下の例を参照してください。

- **labelSelector** で **ManagedClusters** を選択します。以下の例では、**labelSelector** はラベル `vendor: OpenShift` のクラスターだけに一致します。

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: Placement
metadata:
  name: placement1
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
      labelSelector:
        matchLabels:
          vendor: OpenShift
```

- **claimSelector** で **ManagedClusters** を選択します。以下の例では、**claimSelector** は `region.open-cluster-management.io` が `us-west-1` のクラスターだけに一致します。

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: Placement
metadata:
  name: placement2
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
      claimSelector:
        matchExpressions:
          - key: region.open-cluster-management.io
            operator: In
            values:
              - us-west-1
```

- **clusterSets** から **ManagedClusters** を選択します。以下の例では、**claimSelector** は `clusterSets: clusterset1 clusterset2` だけに一致します。

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: Placement
metadata:
  name: placement3
  namespace: ns1
spec:
  clusterSets:
    - clusterset1
```

```

- clusterset2
predicates:
- requiredClusterSelector:
  claimSelector:
  matchExpressions:
  - key: region.open-cluster-management.io
    operator: In
    values:
  - us-west-1

```

- 任意の数の ManagedClusters を選択します。以下の例は、numberOfClusters が 3 の場合です。

```

apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: Placement
metadata:
  name: placement4
  namespace: ns1
spec:
  numberOfClusters: 3
  predicates:
  - requiredClusterSelector:
    labelSelector:
    matchLabels:
    vendor: OpenShift
    claimSelector:
    matchExpressions:
    - key: region.open-cluster-management.io
      operator: In
      values:
    - us-west-1

```

### 13.6.3. 配置のデシジョン

ラベルが `cluster.open-cluster-management.io/placement={placement name}` の PlacementDecisions が 1 つまたは複数作成され、Placement で選択された ManagedClusters を表します。

ManagedCluster が選択され、PlacementDecision に追加されると、この Placement を使用するコンポーネントは、ManagedCluster でワークロードを適用する可能性があります。ManagedCluster が選択されなくなり、PlacementDecisions から削除されると、この ManagedCluster に適用されるワークロードも随時削除する必要があります。

以下の PlacementDecision の例を参照してください。

```

apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: PlacementDecision
metadata:
  labels:
  cluster.open-cluster-management.io/placement: placement1
  name: placement1-kbc7q
  namespace: ns1
ownerReferences:
- apiVersion: cluster.open-cluster-management.io/v1alpha1
  blockOwnerDeletion: true

```

```
controller: true
kind: Placement
name: placement1
uid: 05441cf6-2543-4ecc-8389-1079b42fe63e
```

```
status:
```

```
decisions:
```

- clusterName: cluster1  
reason: "
- clusterName: cluster2  
reason: "
- clusterName: cluster3  
reason: "

## 第14章 クラスタープールの管理 (テクノロジープレビュー)

クラスタープールは、Red Hat OpenShift Container Platform クラスターにオンデマンドで、スケーリングする場合に、迅速かつコスト効果を高く保ちながら、アクセスできるようにします。クラスタープールは、Amazon Web Services、Google Cloud Platform または Microsoft Azure で設定可能な数多くの OpenShift Container Platform クラスターを維持します。このプールは、開発、継続統合および実稼働のシナリオにおいてクラスター環境を提供したり、置き換えたりする場合に特に便利です。

ClusterClaim リソースは、クラスタープールからクラスターをチェックアウトするために使用されます。クラスターがクラスタープールからチェックアウトされると、チェックされたクラスターは再開されて、Ready 状態に切り替わります。クラスタープールは、クラスタープールに休止状態のクラスターを自動的に作成して要求されたクラスターと置き換えます。

注記: クラスタープールから要求されたクラスターが不要になり、破棄されると、リソースは削除されます。クラスターはクラスタープールに戻りません。

- [クラスタープールの作成](#)
- [前提条件](#)
- [クラスタープールの作成](#)
- [クラスタープールからのクラスターの要求](#)
- [クラスタープールのスケーリング](#)
- [クラスタープールリリースイメージの更新](#)
- [クラスタープールの破棄](#)

### 14.1. クラスタープールの作成

クラスタープールを作成する手順は、クラスターの作成手順と似ています。クラスタープールのクラスターは、すぐ使用するために作成されるわけではありません。

必要なアクセス権限:: 管理者

#### 14.1.1. 前提条件

クラスタープールを作成する前に、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。
- プロバイダー環境で 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
- AWS、GCP、または Microsoft Azure プロバイダーの認証情報。詳細は、「[認証情報の管理](#)」を参照してください。
- お使いのプロバイダー環境で設定済みのドメイン。ドメインの設定方法は、プロバイダーのドキュメントを参照してください。
- プロバイダーのログイン認証情報。

- [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。

### 14.1.2. クラスタープールの作成

クラスタープールを作成するには、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 `infrastructure > Clusters` をクリックします。
2. `Cluster pools` タブを選択して、アクセス可能なクラスタープールを表示します。
3. `Create cluster pool` を選択します。
4. クラスタープールのインフラストラクチャプロバイダーを選択します。
5. クラスタープールに使用するインフラストラクチャーの認証情報を選択します。一覧にない場合は、`Add credential` を選択して作成します。
6. `Next` を選択して、クラスタープールの詳細に進みます。
7. クラスタープールの名前を入力します。
8. クラスタープールを作成する `namespace` を入力します。一覧から既存の `namespace` を選択するか、作成する新規 `namespace` の名前を入力します。クラスタープールは、クラスターと同じ `namespace` に配置する必要はありません。  
クラスターセットでクラスタープールを作成すると、クラスタープールの追加先の `namespace` の `clusterset admin` パーミッションがある全ユーザーに、`namespace admin` パーミッションが適用されます。同様に、`namespace view` パーミッションは `clusterset view` パーミッションのあるユーザーに適用されます。
9. オプション: クラスタープールの RBAC ロールを使用して、既存クラスターセットのロール割り当てを共有する場合は、クラスターセット名を選択します。  
注記: `cluster admin` のパーミッションがない場合は、クラスターセットを選択する必要があります。この状況でクラスターセットの名前が含まれない場合には、禁止エラーで、クラスターセットの作成要求が拒否されます。選択できるクラスターセットがない場合には、クラスター管理者に連絡してクラスターセットを作成し、`clusterset admin` パーミッションを付与してもらいます。
10. クラスタープールのサイズに、クラスタープールで利用可能にするクラスターの数を追加して、入力します。
11. `Next` を選択します。
12. 残りの手順で必要な情報を入力して、クラスタープールをプロビジョニングします。以下の手順は、クラスターを作成する手順と同じです。  
プロバイダーに必要な固有の情報は、以下を参照してください。
  - [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](#)
13. すべての情報を入力したら、`Create` を選択してクラスタープールを作成します。1つ以上のクラスターがプールで利用可能になると、`Claim cluster` ボタンが表示されます。

## 14.2. クラスタプールからのクラスタの要求

**ClusterClaim** リソースは、クラスタプールからクラスタをチェックアウトするために使用されます。クラスタがクラスタプールからチェックアウトされると、チェックされたクラスタは Ready 状態に切り替わります。クラスタプールは、クラスタプールに休止状態のクラスタを新たに自動作成し、要求されたクラスタと置き換えます。

注記: クラスタプールから要求されたクラスタが不要になり、破棄されると、リソースは削除されます。クラスタはクラスタプールに戻りません。

必要なアクセス権限:: 管理者

### 14.2.1. 前提条件

クラスタプールからクラスタを要求する前に、以下を利用意する必要があります。

要求可能なクラスタが含まれるクラスタプール。クラスタプールの作成方法については、「[クラスタプールの作成](#)」を参照してください。要求可能なクラスタがない場合には、Claim Cluster ボタンは表示されません。

### 14.2.2. クラスタプールからのクラスタの要求

クラスタプールからクラスタをチェックアウトするには、クラスタを再開し、マネージドクラスタの1つとして要求します。以下の手順を実行してクラスタを要求します。

1. ナビゲーションメニューから infrastructure > Clusters をクリックします。
2. Cluster pools タブを選択します。
3. クラスタを要求するクラスタプールの名前を見つけ、Claim cluster を選択します。
4. クラスタの特定に役立つクラスタの要求名を入力します。
5. Claim を選択します。

クラスタは要求されると、休止状態を終了して、**Resuming** のステータスで Managed Cluster タブに表示されます。クラスタのステータスが **Ready** になると、クラスタは使用できます。

## 14.3. クラスタプールのスケーリング

クラスタプールのサイズを変更するには、クラスタプール内の休止状態のクラスタの数を増減できます。

必要なアクセス権限:: クラスタの管理者

クラスタプールのクラスタ数を変更するには、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 infrastructure > Clusters をクリックします。
2. Cluster pools タブを選択します。
3. Cluster pools の表でクラスタプールの名前を見つけます。
4. 変更するクラスタプールの Options メニューで、Scale cluster pool を選択します。
5. プールサイズの値を変更して、プールサイズを増減します。

6. Scale を選択して変更をコミットします。

クラスタープールは、新しい値を反映するようにスケーリングされます。

## 14.4. クラスタープールリリースイメージの更新

クラスタープールのクラスターが一定期間、休止状態のままになると、クラスターの Red Hat OpenShift Container Platform リリースイメージがバックレベルになる可能性があります。このような場合は、クラスタープールにあるクラスターのリリースイメージのバージョンをアップグレードしてください。

必要なアクセス: 編集

クラスタープールにあるクラスターの OpenShift Container Platform リリースイメージを更新するには、以下の手順を実行します。

注記: この手順では、クラスタープールですでに要求されているクラスターや休止状態のクラスターは更新されません。この手順の完了後に、更新済みのリリースイメージを使用するのは、クラスタープールで作成されたクラスターと、クラスタープールに追加されたクラスターのみです。

1. ナビゲーションメニューから infrastructure > Clusters をクリックします。
2. Cluster pools タブを選択します。
3. クラスタープールの表で、更新するクラスタープールの名前を見つけます。
4. Options メニューをクリックして、利用可能なオプションを表示します。
5. Update release image を選択します。
6. このクラスタープールから今後、クラスターの作成に使用する新規リリースイメージを選択します。
7. Update を選択して、選択したリリースイメージを適用します。
8. クラスタープールのリリースイメージが更新されました。

ヒント: アクション1つで複数のクラスターのリリースイメージを更新するには、各クラスタープールのボックスを選択して Actions メニューを使用し、選択したクラスタープールのリリースイメージを更新します。

## 14.5. クラスタープールの破棄

クラスタープールを作成し、不要になった場合は、そのクラスタープールを破棄できます。クラスタープールを破棄する時に、要求されていない休止状態のクラスターはすべて破棄され、そのリソースは解放されます。

必要なアクセス権限:: クラスタの管理者

クラスタープールを破棄するには、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 infrastructure > Clusters をクリックします。
2. Cluster pools タブを選択します。
3. Cluster pools の表でクラスタープールの名前を見つけます。

4. Options メニューをクリックし、クラスタープールのオプションを表示します。
5. Destroy cluster pool を選択します。
6. クラスターの名前を入力して確定します。
7. Destroy を選択して、クラスタープールを削除します。破棄されたクラスターは、破棄のタイミングでコンソールから削除されます。クラスターがコンソールから削除されると、すべてのリソースが削除されるまで時間がかかる場合があります。  
クラスタープールを破棄する時に、そのシークレットを手動で削除する必要があります。シークレットを削除しない場合には、新規シークレットが既存のシークレットと競合するため、同じ名前および namespace でクラスタープールを作成しようとするとう失敗します。

ヒント: アクション1つで複数のクラスタープールを破棄するには、各クラスタープールのボックスを選択して Actions メニューを使用し、選択したクラスタープールを破棄します。

クラスタープールが削除されました。

## 第15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
<b>id.k8s.io</b>	true	false	アップストリームの提案で定義された ClusterID
<b>kubeversion.open-cluster-management.io</b>	true	true	Kubernetes バージョン
<b>platform.open-cluster-management.io</b>	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
<b>product.open-cluster-management.io</b>	true	false	OpenShift、Anthos、EKS、および GKE などの製品名
<b>id.openshift.io</b>	false	false	OpenShift Container Platform クラスターでのみ利用できる OpenShift Container Platform 外部 ID
<b>consoleurl.openshift.io</b>	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
<b>version.openshift.io</b>	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
```

## 15.1. 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
```

## 15.2. 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16章 DISCOVERY サービスの概要 (テクノロジープレビュー)

[OpenShift Cluster Manager](#) で利用可能な OpenShift 4 クラスターを検出できます。検出後に、クラスターをインポートして管理できます。Discovery サービスは、バックエンドおよびコンソールでの用途に Discover Operator を使用します。

OpenShift Cluster Manager 認証情報があること。認証情報を作成する必要がある場合には、[Red Hat OpenShift Cluster Manager の認証情報の作成](#) を参照してください。

必要なアクセス権限:: 管理者

- [コンソールを使用した検出機能の設定](#)
- [CLI を使用した検出機能の設定](#)
- [検出されたクラスターの表示](#)
- [検出クラスターのインポート](#)

### 16.1. コンソールでの検出の設定 (テクノロジープレビュー)

製品のコンソールを使用して検出を有効にします。

必要なアクセス権: 認証情報が作成された namespace へのアクセス権。

#### 16.1.1. 前提条件

- 認証情報が必要です。OpenShift Cluster Manager に接続するには、[Red Hat OpenShift Cluster Manager の認証情報の作成](#) を参照してください。

#### 16.1.2. 検出の設定

コンソールで検出を設定し、クラスターを検索します。個別の認証情報を使用して複数の `DiscoveryConfig` リソースを作成できます。

検出を初めて使用する場合は、以下の手順を参照してください。

1. コンソールナビゲーションで Clusters をクリックします。
2. Configure Discovery をクリックして検出設定を指定します。
3. namespace および認証情報を選択します。
  - a. 認証情報を含む namespace を選択します。
  - b. その namespace 内での認証情報を選択します。
4. フィルターを設定して、関連するクラスターを検出します。
  - a. 有効な最新のクラスターのフィルターを選択します。たとえば、7日を選択した場合は、直前の7日間にアクティブなクラスターを検索します。
  - b. Red Hat OpenShift バージョンを選択します。
5. Add をクリックします。

6. Clusters ページから、コンソールで検出されたクラスターを表示します。
7. オプション: Create discovery settings をクリックして、他のクラスターを作成します。
8. オプション: Configure discovery settings をクリックして、検出設定を変更したり、削除したりします。

## 16.2. CLI を使用した検出の有効化 (テクノロジープレビュー)

CLI を使用して検出を有効にし、Red Hat OpenShift Cluster Manager (リンク) が入手できるクラスターを見つけます。

必要なアクセス権限:: 管理者

### 16.2.1. 前提条件

- Red Hat OpenShift Cluster Manager に接続するための認証情報を作成しておく。

### 16.2.2. 検出の設定とプロセス

注記: `DiscoveryConfig` は `discovery` という名前に指定し、選択した `credential` と同じ namespace に作成する必要があります。以下の `DiscoveryConfig` のサンプルを参照してください。

```
apiVersion: discovery.open-cluster-management.io/v1alpha1
kind: DiscoveryConfig
metadata:
  name: discovery
  namespace: <NAMESPACE_NAME>
spec:
  credential: <SECRET_NAME>
  filters:
    lastActive: 7
    openshiftVersions:
      - "4.8"
      - "4.7"
      - "4.6"
      - "4.5"
```

1. `SECRET_NAME` は、以前に設定した認証情報に置き換えます。
2. `NAMESPACE_NAME` は `SECRET_NAME` の namespace に置き換えます。
3. クラスターの最後のアクティビティ (日数) からの最大時間を入力します。たとえば、`lastActive: 7` は、過去 7 日間にアクティブなクラスターのみ検出します。
4. Red Hat OpenShift クラスターのバージョンを入力して、文字列の一覧として検出します。注記: `openshiftVersions` 一覧に含まれるエントリはすべて、OpenShift のメジャーバージョンとマイナーバージョンを指定します。たとえば、`"4.7"` には OpenShift バージョン 4.7 (4.7.1、4.7.2 など) のパッチリリースすべてが含まれます。

### 16.2.3. 検出されたクラスターの表示

検出されたクラスターを表示するには、`oc get discoveredclusters -n <namespace>` を実行して、`namespace` は検出認証情報が存在する namespace に置き換えます。

### 16.2.3.1. DiscoveredClusters

オブジェクトは Discovery コントローラーにより作成されます。このような DiscoveredClusters は、DiscoveryConfig discoveredclusters.discovery.open-cluster-management.io API で指定したフィルターと認証情報を使用して OpenShift Cluster Manager で検出されたクラスタを表します。name の値はクラスタの外部 ID です。

```

apiVersion: discovery.open-cluster-management.io/v1alpha1
kind: DiscoveredCluster
metadata:
  name: fd51aafa-95a8-41f7-a992-6fb95eed3c8e
  namespace: <NAMESPACE_NAME>
spec:
  activity_timestamp: "2021-04-19T21:06:14Z"
  cloudProvider: vsphere
  console: https://console-openshift-console.apps.qe1-vmware-pkt.dev02.red-chesterfield.com
  creation_timestamp: "2021-04-19T16:29:53Z"
  credential:
    apiVersion: v1
    kind: Secret
    name: <SECRET_NAME>
    namespace: <NAMESPACE_NAME>
  display_name: qe1-vmware-pkt.dev02.red-chesterfield.com
  name: fd51aafa-95a8-41f7-a992-6fb95eed3c8e
  openshiftVersion: 4.6.22
  status: Stale

```

## 16.3. 検出したクラスタの表示 (テクノロジープレビュー)

認証情報を設定してインポートするクラスタを検出した後に、コンソールで表示できます。

- Cluster > Discovered clusters の順にクリックします。
- 以下の情報が投入された表を確認してください。
  - name は、OpenShift Cluster Manager で指定された表示名です。クラスタに表示名がない場合には、クラスタコンソール URL をもとに生成された名前が表示されます。OpenShift Cluster Manager でコンソール URL がない場合や手動で変更された場合には、クラスタの外部 ID が表示されます。
  - Namespace は、認証情報および検出クラスタを作成した namespace です。
  - type は検出されたクラスタの Red Hat OpenShift タイプです。
  - distribution version は、検出されたクラスタの Red Hat OpenShift バージョンです。
  - infrastructure provider は検出されたクラスタのクラウドプロバイダーです。
  - Last active は、検出されたクラスタが最後にアクティブであった時間です。
  - Created は検出クラスタが作成された時間です。
  - Discovered は検出クラスタが検出された時間です。
- 表の中にある情報はどれでも検索できます。たとえば、特定の namespace で Discovered clusters のみを表示するには、その namespace を検索します。

4. Import cluster をクリックすると、マネージドクラスターを作成できます。[検出クラスターのインポート](#)を参照してください。

## 16.4. 検出クラスターのインポート (テクノロジープレビュー)

クラスターの検出後に、コンソールの Discovered clusters に表示されるクラスターをインポートできます。

### 16.4.1. 前提条件

検出の設定に使用した namespace へのアクセス権が必要である。

### 16.4.2. 検出クラスターのインポート

1. 既存の Clusters ページに移動し、Discovered clusters タブをクリックします。
2. Discovered Clusters の表から、インポートするクラスターを見つけます。
3. オプションメニューから Import cluster を選択します。
4. 検出クラスターの場合には、本書を使用して手動でインポートしたり、Import cluster を自動的に選択したりできます。
5. 認証情報または Kubeconfig ファイルを使用して自動でインポートするには、コンテンツをコピーして貼り付けます。
6. Import をクリックします。

## 第17章 クラスタのアップグレード

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が使用するバージョンチャン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をアップグレードする前に、マネージドクラスタのバージョンチャンネルを更新する必要があります。マネージドクラスタのバージョンチャンネルを更新後に、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または OpenShift Container Platform マネージドクラスタ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クラスタでだけ使用できます。

オンライン環境でクラスタ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 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を選択します。注記: クラスタは 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。

### 17.1. チャンネルの選択

Red Hat Advanced Cluster Management コンソールを使用して、OpenShift Container Platform バージョン 4.6 以降でクラスタのアップグレードのチャンネルを選択できます。チャンネルを選択すると、z-stream バージョン (4.8.1 > 4.8.2 > 4.8.3 など) および y-stream バージョン (4.7 > 4.8 など) の両方で利用可能なクラスタのアップグレードに関するリマインダー通知が自動的に出されます。

クラスターのチャンネルを選択するには、以下の手順を実行します。

1. Red Hat Advanced Cluster Management ナビゲーションで Infrastructure > Clusters に移動します。
2. 変更するクラスターの名前を選択して、Cluster details ページを表示します。クラスターに別のチャンネルが利用可能な場合には、編集アイコンが Channel フィールドに表示されます。
3. 編集アイコンをクリックして、フィールドの設定を変更します。
4. New channel フィールドでチャンネルを選択します。
5. Save をクリックして変更をコミットします。

利用可能なチャンネル更新のリマインダーは、クラスターの Cluster details ページで見つけることができます。

## 17.2. 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。

- [前提条件](#)
- [非接続ミラーレジストリーの準備](#)
- [OpenShift Update Service の Operator のデプロイ](#)
- [グラフデータの init コンテナの構築](#)
- [ミラーリングされたレジストリーの証明書の設定](#)
- [OpenShift Update Service インスタンスのデプロイ](#)
- [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 \(任意\)](#)
- [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のパラメーターを変更するためのポリシーのデプロイ](#)
- [利用可能なアップグレードの表示](#)

- [チャンネルの選択](#)
- [クラスターのアップグレード](#)

### 17.2.1. 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 4.6 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は、「[非接続インストールのイメージのミラーリング](#)」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。

### 17.2.2. 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}:${OCP_RELEASE} \
--to=${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せん。

### 17.2.3. 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。

### 17.2.4. グラフデータの init コンテナの構築

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](#) 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ため、**init container** を使用してローカルリポジトリ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。

### 17.2.5. 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。

## 17.2.6. 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/v1beta2
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。

## 17.2.7. 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、Governance > 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。

### 17.2.8. 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 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、Governance > 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/v1
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、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。

### 17.2.9. 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、Governance > 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。

- 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
```

### 17.2.10. 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。

- Red Hat Advanced Cluster Management コンソールにログインします。
- ナビゲーションメニューから Infrastructure > Clusters を選択します。
- 状態が Ready のクラスターを選択します。
- Actions メニューから Upgrade cluster を選択します。
- 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。

### 17.2.11. チャンネルの選択

Red Hat Advanced Cluster Management コンソールを使用して、OpenShift Container Platform バージョン 4.6 以降でクラスターのアップグレードのチャンネルを選択できます。これらのバージョンはミラーレジストリーで利用可能である必要があります。[Selecting a channel](#) の手順を実行して、アップグレードチャンネルを指定します。

### 17.2.12. 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。

- Red Hat Advanced Cluster Management コンソールで Infrastructure > Clusters を選択します。
- 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18章 マネージメントからのクラスタの削除

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

クラスタ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、クラスタが Ready 状態にある場合にだけ利用できます。

重要:

- クラスタを破棄すると、マネージメントから削除され、クラスタのコンポーネントが削除されます。
- マネージドクラスタをデタッチすると、関連する namespace は自動的に削除されます。カスタムリソースをこの namespace に配置しないでください。
  - [コンソールを使用したクラスタの削除](#)
  - [コマンドラインを使用したクラスタの削除](#)
  - [クラスタ削除後の残りのリソースの削除](#)
  - [クラスタの削除後の etcd データベースのデフラグ](#)

### 18.1. コンソールを使用したクラスタの削除

1. ナビゲーションメニューから Infrastructure > Clusters に移動します。
2. マネージメントから削除するクラスタの横にあるオプション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。

注記: local-cluster という名前のハブクラスタ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、ハブクラスタの値を変更できます。

### 18.2. コマンドラインを使用したクラスタの削除

ハブクラスタのコマンドラインを使用してマネージドクラスタ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、ハブクラスタの値を変更できます。

### 18.3. クラスタ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に残ります。
- マネージドクラスタをデタッチする前に、クラスタ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をデタッチする方法はありません。
- マネージドクラスタは、デタッチ時にオンライン状態ではありませんでした。

これらの状況の1つがマネージドクラスタのデタッチの試行に該当する場合は、マネージドクラスタから削除できないリソースがいくつかあります。マネージドクラスタ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
```

### 18.4. クラスタの削除後の ETCD データベースのデフラグ

マネージドクラスターが多数ある場合は、ハブクラスターの etcd データベースのサイズに影響を与える可能性があります。OpenShift Container Platform 4.8 では、マネージドクラスターを削除すると、ハブクラスターの etcd データベースのサイズは自動的に縮小されません。シナリオによっては、etcd データベースは領域不足になる可能性があります。etcdserver: mvcc: database space exceeded のエラーが表示されます。このエラーを修正するには、データベース履歴を圧縮し、etcd データベースのデフラグを実行して etcd データベースのサイズを縮小します。

注記： OpenShift Container Platform バージョン 4.9 以降では、etcd Operator はディスクを自動的にデフラグし、etcd 履歴を圧縮します。手動による介入は必要ありません。以下の手順は、OpenShift Container Platform 4.8 以前のバージョン向けです。

以下の手順を実行して、ハブクラスターで etcd 履歴を圧縮し、ハブクラスターで etcd データベースをデフラグします。

#### 18.4.1. 前提条件

- OpenShift CLI (oc) をインストールすること。
- cluster-admin 権限を持つユーザーとしてログインすること。

#### 18.4.2. 手順

##### 1. etcd 履歴を圧縮します。

- 次に、etcd メンバーへのリモートシェルセッションを開きます。

```
$ oc rsh -n openshift-etcd etcd-control-plane-0.example.com etcdctl endpoint status --cluster -w table
```

- 以下のコマンドを実行して etcd 履歴を圧縮します。

```
sh-4.4#etcdctl compact $(etcdctl endpoint status --write-out="json" | egrep -o '"revision":[0-9]*' | egrep -o '[0-9]*' -m1)
```

出力例

```
$ compacted revision 158774421
```

2. [Defragmenting etcd data](#) で説明されているように、etcd データベースをデフラグし、NOSPACE アラームを消去します。