



Red Hat Advanced Cluster Management for Kubernetes 2.10

インストール

インストール

Red Hat Advanced Cluster Management for Kubernetes 2.10 インストール

インストール

法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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

接続および非接続ネットワークへのインストール、インストールの要件および推奨事項、マルチクラスターでの高度な設定、ならびにアップグレードおよびアンインストールの手順

目次

| | |
|---------------------------------|----------|
| 第1章 インストールとアップグレード | 3 |
| 1.1. パフォーマンスおよびスケーラビリティ | 3 |
| 1.2. ネットワーク接続時のオンラインインストール | 16 |
| 1.3. 切断されたネットワーク環境でのインストール | 22 |
| 1.4. MULTICLUSTERHUB 詳細設定 | 27 |
| 1.5. アップグレード | 33 |
| 1.6. 切断されたネットワーク環境でのアップグレード | 35 |
| 1.7. ポリシーを使用した非接続クラスターのアップグレード | 37 |
| 1.8. アンインストール | 47 |

第1章 インストールとアップグレード

インストールする前に、各製品に必要なハードウェアおよびシステム設定を確認してください。サポートされているバージョンの Red Hat OpenShift Container Platform を使用して、Linux にオンラインでインストールできます。

1. サポートされているバージョンの OpenShift Container Platform が必要です。たとえば、Red Hat OpenShift Service on AWS または Red Hat OpenShift Dedicated を使用できます。
2. Kubernetes Operator 用のマルチクラスターエンジンをインストールする必要があります。

FIPS の通知: **spec.ingress.sslCiphers** で独自の暗号を指定しない場合、**multiclusterhub-operator** は暗号のデフォルトリストを提供します。アップグレードして FIPS に準拠させるには、**Multiclusterhub** リソースから **ECDHE-ECDSA-CHACHA20-POLY1305** および **ECDHE-RSA-CHACHA20-POLY1305** の2つの暗号を削除します。

ハブクラスターとマネージドクラスターの要件とサポートについては、[サポートマトリクス](#) にアクセスしてください。

Red Hat Advanced Cluster Management for Kubernetes をインストールすると、マルチノードクラスターの実稼働環境が設定されます。Red Hat Advanced Cluster Management for Kubernetes は、標準または高可用性設定のいずれかでインストールできます。インストール手順の詳細は、以下のドキュメントを参照してください。

- [クラスターのサイジング](#)
- [パフォーマンスおよびスケーラビリティ](#)
- [ネットワーク接続時のオンラインインストール](#)
- [ネットワーク切断状態でのインストール](#)
- [MultiClusterHub 詳細設定](#)
- [アップグレード](#)
- [切断されたネットワーク環境でのアップグレード](#)
- [ポリシーを使用した非接続クラスターのアップグレード](#)
- [アンインストール](#)

1.1. パフォーマンスおよびスケーラビリティ

Red Hat Advanced Cluster Management for Kubernetes は、特定のスケラビリティおよびパフォーマンスデータを判断するのにテストされています。テストしたエリアは、主にクラスターのスケラビリティと検索パフォーマンスです。この情報は、環境を計画するときに使用できます。

注記: データは、テスト時のラボ環境から取得した結果をもとにしています。

Red Hat Advanced Cluster Management は、ベアメタルマシン上の3ノードハブクラスターを使用してテストされます。テスト時には、ソフトウェアコンポーネント制限の特定に十分な量のリソース容量 (CPU、メモリー、ディスク) が存在するようにします。結果は、お使いの環境、ネットワークの速度、および製品への変更により、異なる可能性があります。

- [マネージドクラスターの最大数](#)

- スケーラビリティの検索
- 可観測性のスケーラビリティ
- バックアップおよび復元のスケーラビリティ

1.1.1. マネージドクラスタの最大数

Red Hat Advanced Cluster Management が管理できるクラスタの最大数は、以下のような複数の要因により異なります。

- クラスタ内のリソース数。この数はデプロイするポリシーやアプリケーションの数などの要素により異なります。
- スケーリングに使用する Pod 数など、ハブクラスタの設定。

マネージドクラスタは、Red Hat Enterprise Linux ハイパーバイザーでホストされるシングルノードの OpenShift 仮想マシンです。仮想マシンは、テストベッド内の単一のベアメタルマシンごとに高密度のクラスタ数を実現するために使用されます。Sushy-emulator は、Redfish API を使用してアクセス可能なベアメタルクラスタを仮想マシンに持たせるために、libvirt とともに使用されます。次の Operator は、Topology Aware Lifecycle Manager、Local Storage Operator、および Red Hat OpenShift GitOps のテストインストールの一部です。次の表は、ラボ環境のスケーリング情報を示しています。

表1.1 環境スケーリングの表

| ノード | 数 | オペレーティングシステム | ハードウェア | CPU コア数 | メモリー | ディスク |
|-------------------|------|------------------------------|--------|---------|---------|---|
| ハブクラスタのコントロールプレーン | 3 | OpenShift Container Platform | ベアメタル | 112 | 512 GiB | 446 GB SSD, 2.9 TB NVMe, 2 x 1.8 TB SSD |
| マネージドクラスタ | 3500 | シングルノード OpenShift | 仮想マシン | 8 | 18 GiB | 120 GB |

1.1.2. スケーラビリティの検索

検索コンポーネントのスケーラビリティは、データストアのパフォーマンスにより異なります。クエリーの実行時間は、検索パフォーマンスを分析する際の重要な変数です。

1.1.2.1. クエリー実行時の考慮事項

クエリーを実行して結果が返されるまでの所要時間に、影響を与える事項が複数あります。環境のプランニングおよび設定時に、以下の項目を考慮してください。

- キーワードの検索は効率的ではない。
多数のクラスタを管理している場合に **RedHat** と検索すると、検索結果を受け取るのに時間がかかる場合があります。

- 最初の検索は、ユーザーロールベースのアクセス制御ルールを収集するのに時間が余計にかかるため、2 番目以降の検索よりも時間がかかる。
- 要求の完了にかかる時間は、ユーザーのアクセスが許可されている namespace とリソースの数に比例する。
注記: 検索クエリーを保存して他のユーザーと共有する場合に、返される結果は、対象のユーザーのアクセスレベルにより異なります。ロールアクセスの詳細は、OpenShift Container Platform ドキュメントの [RBAC の仕様によるパーミッションの定義および適用](#) を参照してください。
- 要求が全 namespace または全マネージドクラスターにアクセス権限のある非管理者ユーザーからの場合に、最も悪いパフォーマンスが確認された。

1.1.3. 可観測性のスケーラビリティ

可観測性サービスを有効にして使用する場合は、環境のプランニングが必要です。可観測性コンポーネントのインストール先である OpenShift Container Platform プロジェクトで、後ほど消費するリソースを確保します。使用予定の値は、可観測性コンポーネント全体での使用量合計です。

注記: データは、テスト時のラボ環境から取得した結果をもとにしています。結果は、お使いの環境、ネットワークの速度、および製品への変更により、異なる可能性があります。

1.1.3.1. 可観測性環境の例

このサンプル環境では、Amazon Web Service クラウドプラットフォームにハブクラスターとマネージドクラスターが配置されており、以下のトポロジおよび設定が指定されています。

| ノード | フレーバー | vCPU | RAM (GiB) | ディスクタイプ | ディスクサイズ (GiB) | 数 | リージョン |
|---------|------------|------|-----------|---------|---------------|---|-----------|
| マスターノード | m5.4xlarge | 16 | 64 | gp2 | 100 | 3 | sa-east-1 |
| ワーカーノード | m5.4xlarge | 16 | 64 | gp2 | 100 | 3 | sa-east-1 |

高可用性環境用に、可観測性のデプロイメントを設定します。高可用性環境の場合は、Kubernetes デプロイメントごとにインスタンスが 2 つ、StatefulSet ごとにインスタンスが 3 つ含まれます。

サンプルテストでは、さまざまな数のマネージドクラスターがメトリックのプッシュをシミュレーションし、各テストは 24 時間実行されます。以下のスループットを参照してください。

1.1.3.2. 書き込みスループット

| Pod | 間隔 (分) | 時系列 (分) |
|-----|--------|---------|
| 400 | 1 | 83000 |

1.1.3.3. CPU 使用率 (ミリコア)

テスト時の CPU の使用率は安定しています。

| サイズ | CPU の使用率 |
|-----------|----------|
| 10 x クラスタ | 400 |
| 20 x クラスタ | 800 |

1.1.3.4. RSS およびワーキングセットメモリー

RSS およびワーキングセットメモリーに関する以下の説明を参照してください。

- **メモリー使用量 RSS: `container_memory_rss`** のメトリックから取得。テスト時の安定性を維持します。
- **メモリー使用量のワーキングセット: `container_memory_working_set_bytes`** のメトリクスから取得。テストの進捗に合わせて増加します。

24 時間のテストで、以下の結果が得られました。

| サイズ | メモリー使用量 RSS | メモリー使用量のワーキングセット |
|-----------|-------------|------------------|
| 10 x クラスタ | 9.84 | 4.93 |
| 20 x クラスタ | 13.10 | 8.76 |

1.1.3.5. `thanos-receive` コンポーネントの永続ボリューム

重要: メトリックは、保持期間 (4 日) に達するまで `thanos-receive` に保管されます。他のコンポーネントでは、`thanos-receive` コンポーネントと同じボリューム数は必要ありません。

ディスクの使用量は、テストが進むに連れて増加します。データは 1 日経過後のディスク使用量であるため、最終的なディスク使用量は 4 倍にします。

以下のディスク使用量を参照してください。

| サイズ | ディスク使用量 (GiB) |
|-----------|---------------|
| 10 x クラスタ | 2 |
| 20 x クラスタ | 3 |

1.1.3.6. ネットワーク転送

テスト中、ネットワーク転送で安定性を確保します。サイズおよびネットワーク転送の値を確認します。

| サイズ | 受信ネットワーク転送 | 送信ネットワーク転送 |
|------------|----------------|---------------|
| 10 x クラスター | 1秒あたり 6.55 MB | 1秒あたり 5.80 MB |
| 20 x クラスター | 1秒あたり 13.08 MB | 1秒あたり 10.9 MB |

1.1.3.7. Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3) の合計使用量は増加します。メトリックデータは、デフォルトの保持期間 (5 日) に達するまで S3 に保存されます。以下のディスク使用量を参照してください。

| サイズ | ディスク使用量 (GiB) |
|------------|---------------|
| 10 x クラスター | 16.2 |
| 20 x クラスター | 23.8 |

1.1.4. バックアップおよび復元のスケーラビリティ

大規模な環境で実行されたテストには、バックアップおよび復元用に以下のデータが表示されます。

表1.2 マネージドクラスターバックアップの実行時間の表

| バックアップ | 所要時間 | リソース数 | バックアップメモリー |
|------------------|-------|--------------|----------------------|
| credentials | 2分5秒 | リソース 18272 件 | バックアップサイズ 55MiB |
| managed clusters | 3分22秒 | リソース 58655 件 | バックアップサイズ 38MiB |
| resources | 1分34秒 | リソース 1190 件 | バックアップサイズ 1.7MiB |
| generic/user | 2分56秒 | リソース 0 件 | バックアップサイズ 16.5KiB |

バックアップ合計時間は **10m** です。

表1.3 パッシブハブクラスターを復元するためのランタイムの表

| バックアップ | 所要時間 | リソース数 |
|------------|-------|--------------|
| redentials | 47分8秒 | リソース 18272 件 |
| resources | 3分10秒 | リソース 1190 件 |

| バックアップ | 所要時間 | リソース数 |
|---------------------|------|----------|
| generic/user backup | 0分 | リソース 0 件 |

復元合計時間は **50m18s** です。

バックアップファイルの数は、**BackupSchedule** の作成時に設定された **veleroTtl** パラメーターオプションを使用してプルーニングされます。指定された TTL (存続期間) よりも作成時刻が古いバックアップは期限切れになり、Velero によって保管場所から自動的に削除されます。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name:schedule-acm
  namespace:open-cluster-management-backup
spec:
  veleroSchedule:0 */1 ***
  veleroTtl:120h
```

1.1.5. クラスターのサイジング

Red Hat Advanced Cluster Management for Kubernetes クラスターは一意で、以下のガイドラインは一意のデプロイメントサイズを提供します。推奨事項は、サイズと目的で分類されています。

Red Hat Advanced Cluster Management は、サポートサービスのサイジングおよび配置に以下の条件が適用されます。

- クラスター全体で障害の発生する可能性のあるドメインを分離する **アベイラビリティゾーン**。一般的なクラスターは、3つ以上のアベイラビリティゾーンでほぼ同等のワーカーノード容量を備えています。
- **vCPU の予約と制限** をもとに、コンテナに割り当てるワーカーノードの vCPU 容量が確立されます。vCPU は Kubernetes のコンピュートユニットと同じです。詳細は、Kubernetes の [Meaning of CPU](#) を参照してください。
- **メモリの予約と制限** をもとに、コンテナに割り当てるワーカーノードのメモリ容量が確立されます。
- **永続データ** は製品によって管理され、Kubernetes によって使用される etcd クラスターに保存されます。

重要: OpenShift Container Platform の場合、クラスターのマスターノードを3つのアベイラビリティゾーンに分散します。

1.1.5.1. 製品環境

注記: 以下の要件は、最小要件では**ありません**。

表1.4 製品環境

| ノードのタイプ | アベイラビリティゾーン | etcd | 総予約メモリー | 予約済み CPU の合計 |
|--------------------|-------------|------|--|--|
| マスター | 3 | 3 | OpenShift Container Platform のサイジングガイドライン別 | OpenShift Container Platform のサイジングガイドライン別 |
| ワーカーまたはインフラストラクチャー | 3 | 1 | 12 GB | 6 |

OpenShift Container Platform クラスターは、Red Hat Advanced Cluster Management for Kubernetes に加え、追加のサービスを実行してクラスター機能をサポートします。詳細は、[インフラストラクチャーノードへの Red Hat Advanced Cluster Management ハブクラスターのインストール](#) を参照してください。

1.1.5.1.1. 追加サービスの OpenShift Container Platform

クラスター全体で障害の発生する可能性のあるドメインを分離する **アベイラビリティゾーン**。

表1.5 追加サービス

| Service | ノード数 | アベイラビリティゾーン | インスタンスサイズ | vCPU | メモリー | ストレージサイズ | リソース |
|---------|------|-------------|-----------|------|------|----------|------|
|---------|------|-------------|-----------|------|------|----------|------|

| Service | ノード数 | アベイラ ビリ ティ ー ゾーン | インスタ ンスサイ ズ | vCPU | メモリー | ストレ ージサイ ズ | リソース |
|---|------|------------------------------|-----------------------------|------|-------|------------------|--|
| Amazon Web Services 上の OpenShift Container Platform | 3 | 3 | m5.xlarge | 4 | 16 GB | 120 GB | <p>詳細は、OpenShift Container Platform 製品ドキュメントのカスタマイズによる AWS へのクラスターのインストールを参照してください。</p> <p>また、マシンタイプについても確認してください。</p> |
| OpenShift Container Platform on Google Cloud Platform | 3 | 3 | N1-standard-4 (0.95–6.5 GB) | 4 | 15 GB | 120 GB | <p>クォータの詳細は、Google Cloud Platform の製品ドキュメントを参照してください。</p> <p>また、マシンタイプについても確認してください。</p> |

| Service | ノード数 | アベイラビリティゾーン | インスタンスサイズ | vCPU | メモリー | ストレージサイズ | リソース |
|---|------|-------------|----------------|---------------|-------|----------|---|
| OpenShift Container Platform on Microsoft Azure | 3 | 3 | Standard_D4_v3 | 4 | 16 GB | 120 GB | 詳細は、OpenShift Container Platform ドキュメントの Azure アカウントの設定 を参照してください。 |
| VMware vSphere 上の OpenShift Container Platform | 3 | 3 | | 42 ソケットごとのコア) | 16 GB | 120 GB | 詳細は、以下の 製品ドキュメント を参照してください。 |
| IBM Z システムの OpenShift Container Platform | 3 | 3 | | 10 | 16 GB | 100 GB | <p>詳細は、OpenShift Container Platform ドキュメントの クラスタの IBM Z システムへのインストール を参照してください。</p> <p>IBM Z システムには、同時マルチスレッド (SMT) を設定する機能があり、各コアで実行できる vCPU の数を拡張します。</p> |

| Service | ノード数 | アベイラビリティゾーン | インスタンスサイズ | vCPU | メモリー | ストレージサイズ | SMT を設定している場合は、1つの物理コア |
|---------|------|-------------|-----------|------|------|----------|---|
| | | | | | | | <p>(IFL) は 2 つの論理コア (スレッド) を提供します。ハイパーバイザーは、2 つ以上の vCPU を提供できます。</p> <p>1vCPU は、同時マルチスレッド (SMT) またはハイパースレッディングが有効にされていない場合に 1 つの物理コアと同等です。これが有効にされている場合、以下の数式を使用して対応する比率を計算します: $(\text{コアごとのスレッド} \times \text{コア数}) \times \text{ソケット数} = \text{vCPU}$</p> <p>SMT の詳細は、Simultaneous multithreading を参照してください。</p> |

| Service | ノード数 | アベイラビリティ | インスタンスサイズ | vCPU | メモリー | ストレージサイズ | リソース |
|---|------|----------|-----------|------|-------|----------|---|
| IBM Power Systems 上の OpenShift Container Platform | 3 | ティールゾーン | | 16 | 16 GB | 120 GB | <p>詳細は、OpenShift Container Platform ドキュメントのクラスタの Power システムへのインストールを参照してください。</p> <p>IBM Power システムには、同時マルチスレッド (SMT) を設定する機能があり、各コアで実行できる vCPU の数を拡張します。SMT を設定した場合、その SMT レベルでは vCPU 16 個という要件を満たす方法が決まります。以下は、最も一般的な設定です。</p> <p>SMT-8 (IBM PowerVM を実行しているシステムのデフォルト)</p> |

| Service | ノード数 | アベイラビリティゾーン | インスタンスサイズ | vCPU | メモリー | ストレージサイズ | ト設定)で実行しているコア2つでは、必要とされる16個のvCPUを提供します。 SMT-4で実行しているコア4つでは、必要とされる16個のvCPUを提供します。 SMTの詳細は、 Simultaneous multithreading を参照してください。 |
|---------|------|-------------|-----------|------|------|----------|--|
| | | | | | | | |

| Service | ノード数 | アベイラ ビリ ティ ゾーン | インスタ ンスサイ ズ | vCPU | メモリー | ストレ ージサイ ズ | リソース |
|--|------|-------------------------|-------------------|------|-------|------------------|---|
| OpenShift Container Platform オンプレ ミス | 3 | | | 4 | 16 GB | 120 GB | <p>詳細は、以下の 製品ドキュメント を参照してください。</p> <p>Red Hat Advanced Cluster Management for Kubernetes ハブクラスターは、OpenShift Container Platform ベアメタルにインストールし、サポートできます。ハブクラスターは、スケジュール可能な3つのコントロールプレーンノードがあり、追加のワーカーが0の、コンパクトなベアメタルポロジで実行できます。</p> |

1.1.5.1.2. シングルノードの OpenShift Container Platform クラスターの作成および管理

要件は、[単一ノードへのインストール](#) を参照してください。各クラスターは固有であるため、次のガイドラインでは、サイズと目的によって分類されたデプロイメント要件のサンプルのみを提供します。

クラスター全体で障害の発生する可能性のあるドメインを分離する **アベイラビリティゾーン**。一般的なクラスターは、3 つ以上のアベイラビリティゾーンでほぼ同等のワーカーノード容量を備えています。高可用性はサポートされていません。

重要: OpenShift Container Platform の場合、クラスターのマスターノードを 3 つのアベイラビリティゾーンに分散します。

3500 個のシングルノード OpenShift Container Platform クラスターを作成および管理するための要件の例を参照してください。Red Hat Advanced Cluster Management を使用してシングルノード OpenShift クラスター (230 以上を同時にプロビジョニング) を作成し、ハブクラスターで SNO クラスターを管理する最小要件を示しています。

表1.6 マスター (スケジュール可能)

| ノード数 | メモリー (クラスターのピーク使用量) | メモリー (シングルノードの最小値から最大値) | CPU クラスター | CPU シングルノード |
|------|---------------------|-------------------------|-----------|-------------|
| 3 | 289 GB | 64 GB - 110 GB | 90 | 44 |

1.2. ネットワーク接続時のオンラインインストール

Red Hat Advanced Cluster Management for Kubernetes は `{olm-long}` でインストールします。Operator Lifecycle Manager は、Red Hat Advanced Cluster Management ハブクラスターを含むコンポーネントのインストール、アップグレード、および削除を管理します。

必要なアクセス権限: クラスターの管理者。OpenShift Container Platform Dedicated 環境に必要なアクセス: **cluster-admin** パーミッションが必要です。デフォルトで、**dedicated-admin** ロールには OpenShift Container Platform Dedicated 環境で namespace を作成するために必要なパーミッションがありません。

- デフォルトでは、ハブクラスターコンポーネントは追加設定なしで OpenShift Container Platform クラスターのワーカーノードにインストールされます。OpenShift Container Platform OperatorHub Web コンソールインターフェイスを使用するか、OpenShift Container Platform CLI を使用してハブクラスターをワーカーノードにインストールできます。
- OpenShift Container Platform クラスターをインフラストラクチャーノードで設定している場合は、追加のリソースパラメーターを使用して、OpenShift Container Platform CLI を使用してハブクラスターをそれらのインフラストラクチャーノードにインストールできます。詳細は、[インフラストラクチャーノードへの Red Hat Advanced Cluster Management ハブクラスターのインストール](#) セクションを参照してください。
- OpenShift Container Platform または Red Hat Advanced Cluster Management で作成されていない Kubernetes クラスターをインポートする予定の場合は、イメージプルシークレットを設定する必要があります。

詳細設定の設定方法は、本書の [MultiClusterHub の詳細設定セクション](#) のオプションを参照してください。

- [前提条件](#)
- [OpenShift Container Platform インストールの確認](#)

- [OperatorHub Web コンソールインターフェイスからのインストール](#)
- [OpenShift Container Platform CLI からのインストール](#)
- [インフラストラクチャーノードへの Red Hat Advanced Cluster Management ハブクラスターのインストール](#)

1.2.1. 前提条件

Red Hat Advanced Cluster Management をインストールする前に、以下の要件を満たす必要があります。

- Red Hat OpenShift Container Platform クラスターは、OpenShift Container Platform コンソールから OperatorHub カタログの Red Hat Advanced Cluster Management Operator にアクセスできる。
- catalog.redhat.com へのアクセスがある。
- お使いの環境に OpenShift Container Platform バージョン 4.13 以降をデプロイし、OpenShift Container Platform CLI でログインしている。OpenShift Container Platform の以下のインストールドキュメントを参照し、[OpenShift Container Platform バージョン 4.13 以前のバージョン](#)に変更してください。
- OpenShift Container Platform のコマンドラインインターフェイス (CLI) は、**oc** コマンドを実行できるように設定している。Red Hat OpenShift CLI 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。
- namespace の作成が可能な OpenShift Container Platform のパーミッションを設定している。namespace がないと、インストールは失敗します。
- operator の依存関係にアクセスするには、インターネット接続が必要。
- **重要:** OpenShift Container Platform Dedicated 環境にインストールするには、以下の要件を参照してください。
 - OpenShift Container Platform Dedicated 環境が設定され、実行している。
 - ハブクラスターのインストール先の OpenShift Container Platform Deplicated 環境での **cluster-admin** がある。
 - インポートするには、2.10 用の klusterlet Operator の **stable-2.0** チャンネルを使用する必要があります。

1.2.2. OpenShift Container Platform インストールの確認

レジストリー、ストレージサービスなど、サポート対象の OpenShift Container Platform バージョンがインストールされ、機能する状態である必要があります。OpenShift Container Platform のインストールの詳細は、OpenShift Container Platform のドキュメントを参照してください。

1. Red Hat Advanced Cluster Management ハブクラスターが OpenShift Container Platform クラスターにインストールされていないことを確認します。Red Hat Advanced Cluster Management では、各 OpenShift Container Platform クラスターでは1つの Red Hat Advanced Cluster Management ハブクラスターのインストールのみが可能です。Red Hat Advanced Cluster Management ハブクラスターがインストールされていない場合は、以下の手順に進みます。

2. OpenShift Container Platform クラスターが正しく設定されていることを確認するには、以下のコマンドを使用して OpenShift Container Platform Web コンソールにアクセスします。

```
kubectl -n openshift-console get route
```

以下の出力例を参照してください。

```
openshift-console console console-openshift-console.apps.new-coral.purple-chesterfield.com
console https reencrypt/Redirect None
```

3. ブラウザーで URL を開き、結果を確認します。コンソール URL の表示が **console-openshift-console.router.default.svc.cluster.local** の場合は、Red Hat OpenShift Container Platform のインストール時に **openshift_master_default_subdomain** を設定します。 <https://console-openshift-console.apps.new-coral.purple-chesterfield.com> の例を参照してください。

コンソールまたは CLI から、Red Hat Advanced Cluster Management のインストールに進みます。どちらの手順も文書化されています。

1.2.3. OperatorHub Web コンソールインターフェイスからのインストール

ベストプラクティス: OpenShift Container Platform ナビゲーションの **Administrator** ビューから、OpenShift Container Platform で提供される OperatorHub Web コンソールインターフェイスをインストールします。

1. **Operators > OperatorHub** を選択して利用可能な Operator のリストにアクセスし、**Advanced Cluster Management for Kubernetes Operator** を選択します。
2. **Operator サブスクリプション** ページで、インストールのオプションを選択します。
 - namespace 情報:
 - Red Hat Advanced Cluster Management ハブクラスターは、独自の namespace またはプロジェクトにインストールする必要があります。
 - デフォルトでは、OperatorHub コンソールのインストールプロセスで **open-cluster-management** という namespace が作成されます。**ベストプラクティス:** 利用可能な場合は **open-cluster-management** namespace を使用してください。
 - **open-cluster-management** という名前の namespace がすでにある場合は、別の namespace を選択します。
 - チャンネル: インストールするリリースに対応するチャンネルを選択します。チャンネルを選択すると、指定のリリースがインストールされ、そのリリース内の今後のエラー更新が取得されます。
 - 更新の承認戦略: 承認戦略では、サブスクリプション先のチャンネルまたはリリースに更新を適用するのに必要な人の間のやり取りを特定します。
 - **Automatic** を選択して、そのリリース内の更新が自動的に適用されるようにします。
 - **Manual** を選択して、更新が利用可能になると通知を受け取ります。更新が、いつ適用されるかについて懸念がある場合は、これがベストプラクティスになる可能性があります。

重要: 次のマイナーリリースにアップグレードするには、**OperatorHub** ページに戻り、最新リリースの新規チャンネルを選択する必要があります。

3. **Install** を選択して変更を適用し、Operator を作成します。
4. **MultiClusterHub** のカスタムリソースを作成します。
 - a. OpenShift Container Platform コンソールのナビゲーションで **Installed Operators** > **Advanced Cluster Management for Kubernetes** を選択します。
 - b. **MultiClusterHub** タブを選択します。
 - c. **Create MultiClusterHub** を選択します。
 - d. YAML ファイルのデフォルト値を更新します。このドキュメントの **MultiClusterHub の詳細設定** のオプションを参照してください。
 - 以下の例は、デフォルトのテンプレートを示しています。**namespace** がお使いのプロジェクトの namespace であることを確認します。サンプルを参照してください。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
```

5. **Create** を選択して、カスタムリソースを初期化します。Red Hat Advanced Cluster Management ハブクラスターのビルドと起動に、最長で 10 分程度かかる場合があります。Red Hat Advanced Cluster Management ハブクラスターが作成されると、Red Hat Advanced Cluster Management Operator の詳細の **MultiClusterHub** タブから **MultiClusterHub** リソースのステータスが **Running** と表示されます。コンソールへのアクセスについては、[コンソールへのアクセス](#) を参照してください。

1.2.4. OpenShift Container Platform CLI からのインストール

1. Operator 要件を満たした Red Hat Advanced Cluster Management ハブクラスター namespace を作成します。以下のコマンドを実行して、**namespace** はお使いの Red Hat Advanced Cluster Management ハブクラスターの namespace 名になります。**namespace** の値は、OpenShift Container Platform 環境では **プロジェクト** と呼ばれる場合があります。

```
oc create namespace <namespace>
```

2. プロジェクトの namespace を、作成した namespace に切り替えます。**namespace** は、手順 1 で作成した Red Hat Advanced Cluster Management ハブクラスター namespace 名に置き換えます。

```
oc project <namespace>
```

3. **OperatorGroup** リソースを設定するために YAML ファイルを作成します。namespace ごとに割り当てることができる Operator グループは 1 つだけです。**default** はお使いの operator グループ名に置き換えます。**namespace** はお使いのプロジェクトの namespace 名に置き換えます。以下のサンプルを参照してください。

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <default>
  namespace: <namespace>
```

```
spec:
  targetNamespaces:
  - <namespace>
```

- 以下のコマンドを実行して **OperatorGroup** リソースを作成します。 **operator-group** は、作成した operator グループの YAML ファイル名に置き換えます。

```
oc apply -f <path-to-file>/<operator-group>.yaml
```

- OpenShift Container Platform サブスクリプションを設定するための YAML ファイルを作成します。ファイルは以下の例に似ており、 **release-2.x** を現在のリリースに置き換えます。

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: acm-operator-subscription
spec:
  sourceNamespace: openshift-marketplace
  source: redhat-operators
  channel: release-2.x
  installPlanApproval: Automatic
  name: advanced-cluster-management
```

注記: Red Hat Advanced Cluster Management ハブクラスターをインフラストラクチャーノードにインストールする場合は、 [Operator Lifecycle Manager サブスクリプションの追加設定](#) セクションを参照してください。

- 以下のコマンドを実行して OpenShift Container Platform サブスクリプションを作成します。 **subscription** は、作成したサブスクリプションファイル名に置き換えます。

```
oc apply -f <path-to-file>/<subscription>.yaml
```

- YAML ファイルを作成して **MultiClusterHub** カスタムリソースを設定します。デフォルトのテンプレートは、以下の例のようになります。 **namespace** はお使いのプロジェクトの namespace 名に置き換えます。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec: {}
```

注記: Red Hat Advanced Cluster Management ハブクラスターをインフラストラクチャーノードにインストールする場合は、 [MultiClusterHub カスタムリソースの追加設定](#) セクションを参照してください。

- 以下のコマンドを実行して **MultiClusterHub** カスタムリソースを作成します。 **custom-resource** は、カスタムリソースファイル名に置き換えます。

```
oc apply -f <path-to-file>/<custom-resource>.yaml
```

以下のエラーで、この手順に失敗した場合でも、リソースは作成され、適用されます。リソースが作成されてから数分後にもう一度コマンドを実行します。


```
error: unable to recognize "/mch.yaml": no matches for kind "MultiClusterHub" in version "operator.open-cluster-management.io/v1"
```

- 以下のコマンドを実行してカスタムリソースを編集します。コマンドを実行して、**MultiClusterHub** カスタムリソースのステータスが **status.phase** フィールドに **Running** と表示されるまで、最長 10 分の時間がかかる可能性があります。

```
oc get mch -o=jsonpath='{.items[0].status.phase}'
```

Red Hat Advanced Cluster Management を再インストールして、Pod が起動しない場合には、この問題の回避手順について [再インストールに失敗する場合のトラブルシューティング](#) を参照してください。

注記:

- **ClusterRoleBinding** が指定された **ServiceAccount** には、Red Hat Advanced Cluster Management がインストールされている namespace にアクセス権があるユーザー認証情報、および Red Hat Advanced Cluster Management に対して、クラスター管理者権限が割り当てられます。
- このインストールでは、**local-cluster** という名前の namespace も作成されます。この namespace は、単独で管理できるように Red Hat Advanced Cluster Management ハブクラスター向けに確保されます。**local-cluster** という既存の namespace を含めることはできません。セキュリティの理由上、**cluster-administrator** のアクセス権がないユーザーには、**local-cluster** namespace へのアクセス権を割り当てないようにしてください。

1.2.5. インフラストラクチャーノードへの Red Hat Advanced Cluster Management ハブクラスターのインストール

OpenShift Container Platform クラスターを、承認された管理コンポーネントを実行するためのインフラストラクチャーノードを組み込むように設定できます。インフラストラクチャーノードでコンポーネントを実行すると、それらの管理コンポーネントを実行しているノードの OpenShift Container Platform サブスクリプションクォータの割り当ての必要がなくなります。

OpenShift Container Platform クラスターにインフラストラクチャーノードを追加した後、[OpenShift Container Platform CLI からのインストール](#) 手順に従い、設定を Operator Lifecycle Manager サブスクリプションおよび **MultiClusterHub** カスタムリソースに追加します。

1.2.5.1. インフラストラクチャーノードを OpenShift Container Platform クラスターに追加する

OpenShift Container Platform ドキュメントの [インフラストラクチャーマシンセットの作成](#) で説明されている手順に従ってください。インフラストラクチャーノードは、Kubernetes の **taint** および **label** で設定され、管理以外のワークロードがそれらで稼働し続けます。

Red Hat Advanced Cluster Management が提供するインフラストラクチャーノードの有効化と互換性を持たせるために、インフラストラクチャーノードに次の **taint** および **label** が適用されていることを確認してください。

```
metadata:
  labels:
    node-role.kubernetes.io/infra: ""
spec:
```

```
taints:  
- effect: NoSchedule  
  key: node-role.kubernetes.io/infra
```

1.2.5.2. Operator Lifecycle Manager サブスクリプションの追加設定

Operator Lifecycle Manager サブスクリプションを適用する前に、以下の追加設定を追加します。

```
spec:  
  config:  
    nodeSelector:  
      node-role.kubernetes.io/infra: ""  
  tolerations:  
  - key: node-role.kubernetes.io/infra  
    effect: NoSchedule  
    operator: Exists
```

1.2.5.3. MultiClusterHub カスタムリソースの追加設定

MultiClusterHub カスタムリソースを適用する前に、以下の設定を追加します。

```
spec:  
  nodeSelector:  
    node-role.kubernetes.io/infra: ""
```

1.3. 切断されたネットワーク環境でのインストール

切断された Red Hat OpenShift Container Platform クラスターに Red Hat Advanced Cluster Management for Kubernetes をインストールする必要がある場合があります。切断されたハブクラスターにインストールするには、接続されたネットワーク環境用の通常のインストールまたはアップグレード手順に加えて、次の手順を実行します。

Required access: すべてのインストールおよびアップグレードタスクには、**cluster administration** アクセス権が必要です。

以下のセクションを参照してください。

- [前提条件](#)
- [OpenShift Container Platform インストールの確認](#)
- [Operator Lifecycle Manager の設定](#)
- [イメージコンテンツソースポリシーの設定](#)
- [Red Hat Advanced Cluster Management for Kubernetes Operator およびハブのインストール](#)

1.3.1. 前提条件

Red Hat Advanced Cluster Management for Kubernetes をインストールする前に、以下の要件を満たす必要があります。

- 切断されたネットワーク環境にインストールしているため、ローカルイメージレジストリーにアクセスして、ミラーリングされた Operator Lifecycle Manager カタログと Operator イメー

ジを保存する必要があります。おそらく、この環境に OpenShift Container Platform クラスターをインストールするときに、ローカルイメージレジストリーをすでにセットアップしているので、同じローカルイメージレジストリーを使用できるはずですが、

- インターネットとローカルミラーレジストリーの両方にアクセスできるワークステーションが必要です。
- お使いの環境にサポート対象の Red Hat OpenShift Container Platform バージョンインストールし、コマンドラインインターフェイス (CLI) でログインしている。Red Hat OpenShift Container Platform のインストールについては、[OpenShift Container Platform バージョン 4.11 のインストールに関するドキュメント](#) を参照してください。Red Hat OpenShift CLI を使用して `oc` コマンドをインストールおよび設定する方法については [CLI の概要](#) を参照してください。
- ハブクラスターの容量の設定に関する詳細は、[クラスターのサイジング](#) を確認してください。

1.3.2. OpenShift Container Platform インストールの確認

- 接続中に、`oc -n openshift-console get route` コマンドを実行して OpenShift Container Platform Web コンソールにアクセスします。以下の出力例を参照してください。

```
openshift-console      console      console-openshift-console.apps.new-coral.purple-
chesterfield.com      console      https reencrypt/Redirect  None
```

ブラウザで URL を開き、結果を確認します。コンソール URL の表示が **console-openshift-console.router.default.svc.cluster.local** の場合は、Red Hat OpenShift Container Platform のインストール時に `openshift_master_default_subdomain` を設定します。

1.3.3. ローカルイメージレジストリーの可用性の確認

ベストプラクティス: Operator Lifecycle Manager Operator 関連のコンテンツには、既存のミラーレジストリーを使用します。

切断された環境に Red Hat Advanced Cluster Management for Kubernetes をインストールするには、ローカルのミラーイメージレジストリーを使用する必要があります。切断された環境での OpenShift Container Platform クラスターのインストールはすでに完了しているため、Red Hat OpenShift Container Platform クラスターのインストール中に使用するミラーレジストリーがすでにセットアップされています。

ローカルイメージレジストリーがまだない場合は、Red Hat OpenShift Container Platform ドキュメントの [非接続インストール用のイメージのミラーリング](#) で説明されている手順を実行して作成します。

1.3.4. Operator Lifecycle Manager の設定

Red Hat Advanced Cluster Management for Kubernetes は Operator としてパッケージ化されているため、インストールは Operator Lifecycle Manager を使用して実行します。

切断された環境では、Operator Lifecycle Manager は、切断されたクラスターからアクセスできないイメージレジストリーでホストされているため、Red Hat が提供する Operator がアクセスできる標準の Operator ソースにアクセスできません。代わりに、クラスター管理者は、ミラー化されたイメージレジストリーと Operator カタログを使用して、切断された環境で Operator のインストールとアップグレードを有効にすることができます。

Red Hat Advanced Cluster Management for Kubernetes をインストールするために切断されたクラスターを準備するには、OpenShift Container Platform ドキュメントの [制限されたネットワークでの Operator Lifecycle Manager の使用](#) で説明されている手順に従います。

1.3.4.1. 追加要件

前の手順を完了したら、Red Hat Advanced Cluster Management for Kubernetes に固有の以下の要件にも注意してください。

1.3.4.1.1. ミラーカタログに Operator パッケージを含める

- 必要な Operator パッケージをミラーカタログに含めるRed Hat は、**registry.redhat.io/redhat/redhat-operator-index** インデックスイメージで提供される Red Hat Operator カタログで、Red Hat Advanced Cluster Management for Kubernetes Operator を提供します。このカタログインデックスイメージのミラーを準備する場合に、Red Hat が提供するカタログ全体をミラーリングするか、使用する Operator パッケージのみを含むサブセットをミラーリングするかを選択できます。フルミラーカタログを作成する場合、Red Hat Advanced Cluster Management for Kubernetes のインストールに必要なすべてのパッケージが含まれているため、特別に考慮する必要はありません。ただし、特定のパッケージを含めるかを特定するために、一部または絞り込んだミラーリングカタログを作成する場合には、次のパッケージ名をリストに含める必要があります。
 - **advanced-cluster-manager**
 - **multicluster-engine**
- 2つのミラーリング手順のいずれかを使用します。
- OPM ユーティリティー **opm index prune** を使用してミラーリングされたカタログまたはレジストリーを作成している場合は、次の例に示すように、**-p** オプションの値に次のパッケージ名を含め、現在のバージョンを **4.x** に置き換えてください。

```
opm index prune \
  -f registry.redhat.io/redhat/redhat-operator-index:v4.x \
  -p advanced-cluster-management,multicluster-engine \
  -t myregistry.example.com:5000/mirror/my-operator-index:v4.x
```

- 代わりに **oc-mirror** プラグインを使用してミラーリングされたカタログまたはレジストリーにデータを入力する場合は、次の例に示すように、**ImageSetConfiguration** のパッケージリスト部分に次のパッケージ名を含め、現在のバージョンを **4.x** に置き換えてください。

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: myregistry.example.com:5000/mirror/oc-mirror-metadata
mirror:
  platform:
    channels:
      - name: stable-4.x
      type: ocp
  operators:
    - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.11
      packages:
```

```

- name: advanced-cluster-management
- name: multicluster-engine
additionalImages: []
helm: {}

```

1.3.4.1.2. ミラーレジストリーの使用設定

Red Hat Advanced Cluster Management for Kubernetes のインストールに必要な以前のパッケージをローカルミラーレジストリーに入力したら、[制限されたネットワークでの Operator Lifecycle Manager の使用](#) のトピックで説明されている手順を完了して、ネットワーク接続されていないクラスターでミラーレジストリーとカタログを利用できるようにします。この手順には以下が含まれます。

1. デフォルトの OperatorHub ソースの無効化
2. Operator カタログのミラーリング
3. ミラーカタログのカタログソースの追加

1.3.4.1.3. カタログのソース名の検索

Red Hat OpenShift Container Platform ドキュメントの手順で説明されているように、切断されたクラスターに **CatalogSource** リソースを追加する必要があります。**重要:** 後で必要になるため、**metadata.name** フィールドの値を書き留めておいてください。

次の例のような YAML ファイルを使用して、**CatalogSource** リソースを **openshift-marketplace** namespace に追加し、**4.x** を現在のバージョンに置き換えます。

```

apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: my-mirror-catalog-source
  namespace: openshift-marketplace
spec:
  image: myregistry.example.com:5000/mirror/my-operator-index:v4.x
  sourceType: grpc

```

後で作成する **MulticlusterHub** リソースのアノテーションには、**metadata.name** フィールドの値が必要です。

1.3.5. 必要なパッケージが利用可能であることの確認

Operator Lifecycle Manager は、一定の間隔で使用可能なパッケージのカタログソースをポーリングします。Operator Lifecycle Manager がミラーリングされたカタログのカタログソースをポーリングした後、利用可能な **PackageManifest** リソースをクエリーして、必要なパッケージが切断されたクラスターから利用可能であることを確認できます。

切断されたクラスターに向けて、次のコマンドを実行します。

```
oc -n openshift-marketplace get packagemanifests
```

表示されるリストには、次のパッケージがミラーカタログのカタログソースによって提供されていることを示すエントリーが含まれている必要があります。

- **advanced-cluster-manager**

- **multicluster-engine**

1.3.6. イメージコンテンツソースポリシーの設定

クラスターが Red Hat Advanced Cluster Management for Kubernetes Operator のコンテナイメージを、インターネットでホストされているレジストリーからではなく、ミラーレジストリーから取得させるには、切断されたクラスターで **ImageContentSourcePolicy** を設定して、イメージ参照をミラーレジストリーにリダイレクトする必要があります。

oc adm catalog mirror コマンドを使用してカタログをミラーリングした場合に、必要なイメージコンテンツソースポリシー設定は、そのコマンドによって作成される **manifests-*** ディレクトリー内の **imageContentSourcePolicy.yaml** ファイルにあります。

代わりに **oc-mirror** プラグインを使用してカタログをミラーリングした場合に、**imageContentSourcePolicy.yaml** ファイルは **oc-mirror** プラグインによって作成された **oc-mirror-workspace/results-*** ディレクトリー内にあります。

いずれの場合も、次のような **oc apply** または **oc replace** コマンドを使用して、切断されたコマンドにポリシーを適用できます。

```
oc replace -f ./<path>/imageContentSourcePolicy.yaml
```

必要なイメージコンテンツソースポリシーステートメントは、ミラーレジストリーの作成方法によって異なりますが、次の例のようになります。

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  labels:
    operators.openshift.org/catalog: "true"
  name: operator-0
spec:
  repositoryDigestMirrors:
  - mirrors:
    - myregistry.example.com:5000/rhacm2
    source: registry.redhat.io/rhacm2
  - mirrors:
    - myregistry.example.com:5000/multicluster-engine
    source: registry.redhat.io/multicluster-engine
  - mirrors:
    - myregistry.example.com:5000/openshift4
    source: registry.redhat.io/openshift4
  - mirrors:
    - myregistry.example.com:5000/redhat
    source: registry.redhat.io/redhat
```

1.3.7. Red Hat Advanced Cluster Management for Kubernetes Operator およびハブクラスターのインストール

前述のように Operator Lifecycle Manager と Red Hat OpenShift Container Platform を設定したら、OperatorHub コンソールまたは CLI を使用して Red Hat Advanced Cluster Management for Kubernetes をインストールできます。[オンライン接続時のインストール](#) のトピックで説明されているガイダンスどおりに実行します。

重要: **MulticlusterHub** リソースを作成すると、ハブクラスターのインストールプロセスが開始します。

Operator をクラスターにインストールするには、ミラーカタログにデフォルト以外のカタログソースを使用する必要があるため、Operator にミラーカタログソースの名前を提供するために、**MulticlusterHub** リソースに特別なアノテーションが必要です。次の例は、必要な **mce-subscription-spec** アノテーションを示しています。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  namespace: open-cluster-management
  name: hub
  annotations:
    installer.open-cluster-management.io/mce-subscription-spec: '{"source": "my-mirror-catalog-source"}'
spec: {}
```

マルチクラスターエンジン Operator は Red Hat Advanced Cluster Management のインストール中に自動的にインストールされるため、**mce-subscription-spec** アノテーションが必要です。CLI でリソースを作成する場合は、**oc apply** コマンドで適用する YAML に **mce-subscription-spec** アノテーションを含めて、**MulticlusterHub** リソースを作成します。

OperatorHub コンソールを使用してリソースを作成する場合は、**YAML ビュー** に切り替えて、前に表示されたようにアノテーションを挿入します。**重要:** OperatorHub コンソールの **Field view** パネルには、**MulticlusterHub** を作成するためのアノテーション用のフィールドがありません。

1.4. MULTICLUSTERHUB 詳細設定

Red Hat Advanced Cluster Management for Kubernetes は、必要なコンポーネントをすべてデプロイする Operator でインストールします。リストされているコンポーネントの一部はデフォルトで有効になっています。コンポーネントが **disabled** になっている場合、そのリソースは有効になるまでクラスターにデプロイされません。オペレーターは、次のコンポーネントのデプロイに取り組みます。

表1.7 デプロイされたコンポーネントの表のリスト

| 名前 | 説明 | 有効 |
|-------------------|---|-------|
| app-lifecycle | アプリケーションとアプリケーションの更新を構築およびデプロイメントするためのオプションを統合および簡素化します。 | True |
| cluster-backup | マネージドクラスター、アプリケーション、ポリシーなどのすべてのハブクラスターリソースのバックアップと復元のサポートを提供します。 | False |
| cluster-lifecycle | OpenShift Container Platform および Red Hat Advanced Cluster Management ハブクラスターにクラスター管理機能を提供します。 | True |

| | | |
|----------------------------|--|------|
| cluster-permission | RBAC リソースをマネージドクラスターに自動的に分散し、それらのリソースのライフサイクルを管理します。 | True |
| console | Red Hat Advanced Cluster Management Web コンソールプラグインを有効にします。 | True |
| grc | クラスターのポリシーを定義するためのセキュリティ強化を有効にします。 | True |
| insights | クラスター内の既存の問題または潜在的な問題を特定します。 | True |
| multicluster-observability | 監視を有効にして、マネージドクラスターの健全性についてさらに詳しい洞察を得ることができます。 | True |
| search | すべてのクラスターにわたる Kubernetes リソースの可視性を提供します。 | True |
| submariner-addon | オンプレミスまたはクラウドの環境内の 2 つ以上のマネージドクラスター間の直接ネットワーキングとサービス検出を可能にします。 | True |
| volsync | クラスター内の永続ボリュームの非同期レプリケーション、またはレプリケーションに互換性のないストレージタイプのクラスター間での永続ボリュームの非同期レプリケーションをサポートします。 | True |

Red Hat Advanced Cluster Management をクラスターにインストールする場合、リストされているコンポーネントのすべてがデフォルトで有効になるわけではありません。

MultiClusterHub カスタムリソースに 1 つ以上の属性を追加することで、インストール中またはインストール後に Red Hat Advanced Cluster Management をさらに設定できます。追加できる属性については、このまま読み進めてください。

1.4.1. コンソールとコンポーネントの設定

次の例では、コンポーネントを有効または無効にするために使用できる **spec.overrides** デフォルトテンプレートを表示します。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
```



```

metadata:
  name: multiclusterhub
  namespace: <namespace> ❶
spec:
  overrides:
    components:
      - name: <name> ❷
        enabled: true

```

1. **namespace** はお使いのプロジェクト名に置き換えます。
2. **name** をコンポーネントの名前に置き換えます。

あるいは、以下のコマンドを実行します。 **namespace** プロジェクトの名前に置き換え、 **name** をコンポーネントの名前に置き換えます。

```

oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p='[{"op": "add", "path":
"/spec/overrides/components/-", "value": {"name": "<name>", "enabled": true}}]'

```

注記: **console** コンポーネントが無効になると、Red Hat OpenShift Container Platform コンソールも無効になります。

1.4.2. カスタムイメージプルシークレット

OpenShift Container Platform または Red Hat Advanced Cluster Management で作成されていない Kubernetes クラスタをインポートする予定がある場合は、OpenShift Container Platform プルシークレットの情報を含むシークレットを生成して、ディストリビューションレジストリーから資格のあるコンテンツにアクセスします

OpenShift Container Platform クラスタのシークレット要件は、OpenShift Container Platform および Red Hat Advanced Cluster Management により自動で解決されるため、他のタイプの Kubernetes クラスタをインポートして管理しない場合は、このシークレットを作成する必要がありません。OpenShift Container Platform プルシークレットは Red Hat カスタマーポータル ID に関連しており、すべての Kubernetes プロバイダーで同じです。

重要: これらのシークレットは、namespace ごとに異なるため、手順1で作成した namespace で操作を行うようにしてください。

1. cloud.redhat.com/openshift/install/pull-secret に移動して、OpenShift Container Platform のプルシークレットファイルをダウンロードします。
2. **Download pull secret** をクリックします。
3. 以下のコマンドを実行してシークレットを作成します。

```

oc create secret generic <secret> -n <namespace> --from-file=.dockerconfigjson=<path-to-pull-secret> --type=kubernetes.io/dockerconfigjson

```

- **secret** は作成するシークレット名に置き換えます。
- シークレットは namespace 固有であるため、**namespace** はプロジェクトの namespace に置き換えます。
- **path-to-pull-secret** はダウンロードした OpenShift Container Platform のプルシークレットへのパスに置き換えます。

以下の例では、カスタムプルシークレットを使用する場合に使用する **spec.imagePullSecret** テンプレートを表示しています。secret は、プルシークレット名に置き換えます。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  imagePullSecret: <secret>
```

1.4.3. availabilityConfig

Red Hat Advanced Cluster Management ハブクラスターには、**High** と **Basic** の2つのアイラビリティがあります。デフォルトでは、ハブクラスターには **High** の可用性があります。これにより、ハブクラスターコンポーネントに **replicaCount 2** が提供されます。これにより、フェイルオーバー時のサポートが向上しますが、**Basic** 可用性よりも多くのリソースを消費します。これにより、コンポーネントには **replicaCount 1** が提供されます。

重要: シングルノード OpenShift クラスターでマルチクラスターエンジン Operator を使用している場合は、**spec.availabilityConfig** を **Basic** に設定します。

以下の例は、**Basic** の可用性のある **spec.availabilityConfig** テンプレートを示しています。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  availabilityConfig: "Basic"
```

1.4.4. nodeSelector

Red Hat Advanced Cluster Management ハブクラスターでノードセレクターのセットを定義して、クラスターの特定のノードにインストールできます。以下の例は、**node-role.kubernetes.io/infra** ラベルの付いたノードに Red Hat Advanced Cluster Management Pod を割り当てる **spec.nodeSelector** を示しています。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
```

1.4.5. tolerations

容認のリストを定義して、Red Hat Advanced Cluster Management ハブクラスターがクラスターで定義された特定のテイントを容認できるようにします。

以下の例は、`node-role.kubernetes.io/infra` テイントに一致する `spec.tolerations` を示しています。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  tolerations:
  - key: node-role.kubernetes.io/infra
    effect: NoSchedule
    operator: Exists
```

以前の `infra-node` 容認は、設定に容認を指定せずにデフォルトで Pod に設定されます。設定で容認をカスタマイズすると、このデフォルトが置き換えられます。

1.4.6. disableHubSelfManagement

デフォルトでは、Red Hat Advanced Cluster Management ハブクラスターは、自動的にインポートされ、管理されます。この **マネージド** ハブクラスターの名前は **local-cluster** です。ハブクラスターが自身を管理するかどうかを指定する設定は、**multiclusterengine** カスタムリソースにあります。Red Hat Advanced Cluster Management で、この設定を変更すると、**multiclusterengine** カスタムリソースの設定が自動的に変更されます。

注記: マルチクラスターエンジン Operator クラスターを管理している Red Hat Advanced Cluster Management ハブクラスターでは、以前の手動設定はすべてこのアクションに置き換えられます。

Red Hat Advanced Cluster Management ハブクラスターが自身を管理しないようにする場合は、**spec.disableHubSelfManagement** の設定を **false** から **true** に変更する必要があります。この設定が、カスタムリソースを定義する YAML ファイルに含まれていない場合は、これを追加する必要があります。ハブクラスターは、このオプションでのみ管理できます。

このオプションを **true** に設定し、ハブの管理を試みると、予期しない動作が発生します。

以下の例は、ハブクラスターの自己管理機能を無効にする場合に使用するデフォルトのテンプレートです。**namespace** はお使いのプロジェクト名に置き換えます。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  disableHubSelfManagement: true
```

デフォルトの **local-cluster** を有効にするには、設定を **false** に戻すか、この設定を削除します。

1.4.7. disableUpdateClusterImageSets

すべてのクラスターに同じリリースイメージを使用する必要がある場合は、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リストを作成できます。

利用可能なリリースイメージを管理し、カスタムイメージリストの上書きを停止する **spec.disableUpdateClusterImageSets** 属性を設定するには、[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](#) の次の手順を参照してください。

以下の例は、クラスターイメージセットへの更新を無効にするデフォルトのテンプレートです。**namespace** はお使いのプロジェクト名に置き換えます。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  disableUpdateClusterImageSets: true
```

1.4.8. customCAConfigmap (非推奨)

デフォルトで、Red Hat OpenShift Container Platform は Ingress Operator を使用して内部 CA を作成します。

以下の例は、カスタマイズされた OpenShift Container Platform のデフォルト Ingress CA 証明書を Red Hat Advanced Cluster Management に提供するのに使用されるデフォルトのテンプレートです。**namespace** はお使いのプロジェクト名に置き換えます。**spec.customCAConfigmap** の値は **ConfigMap** の名前に置き換えます。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  customCAConfigmap: <configmap>
```

1.4.9. sslCiphers (非推奨)

デフォルトでは、Red Hat Advanced Cluster Management ハブクラスターには、サポートされる SSL 暗号の詳細一覧が含まれます。

以下の例は、管理 Ingress の **sslCiphers** をリスト表示するの使用されるデフォルトの **spec.ingress.sslCiphers** テンプレートです。**namespace** はお使いのプロジェクト名に置き換えます。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  ingress:
    sslCiphers:
      - "ECDHE-ECDSA-AES128-GCM-SHA256"
      - "ECDHE-RSA-AES128-GCM-SHA256"
```

1.4.10. ClusterBackup

enableClusterBackup フィールドはサポートされなくなり、このコンポーネントに置き換えられました。

以下の例は、**ClusterBackup** の有効化に使用される **spec.overrides** のデフォルトテンプレートです。**namespace** はお使いのプロジェクト名に置き換えます。

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  overrides:
    components:
      - name: cluster-backup
        enabled: true
```

あるいは、以下のコマンドを実行します。**namespace** はお使いのプロジェクト名に置き換えます。

```
oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p='[{"op": "add", "path": "/spec/overrides/components/-", "value": {"name": "cluster-backup", "enabled": true}}]'
```

1.5. アップグレード

Red Hat OpenShift Container Platform コンソールの Operator サブスクリプション設定を使用して、Red Hat Advanced Cluster Management for Kubernetes のアップグレードを制御できます。Operator Lifecycle Manager の **OperatorCondition** は、バージョンのアップグレード方法を管理するために使用できます。Operator を使用して Red Hat Advanced Cluster Management の初回デプロイ時に、以下の選択を行います。

- **Channel:** チャンネルは、インストールする製品のバージョンに対応します。多くの場合、最初のチャンネル設定は、インストール時に利用可能な最新のチャンネルです。
- **Approval:** チャンネル内での更新に承認が必要であるか、更新を自動で行うかを指定します。
 - **Automatic** に設定されている場合、選択したチャンネルのマイナーリリース (エラータ) の更新は、管理者の介入なしにデプロイされます。
 - **Manual** に設定されている場合は、チャンネル内でマイナーリリース (エラータ) に更新するたびに、管理者が更新を承認する必要があります。

必要なアクセス: OpenShift Container Platform の管理者

これらの設定は、オペレーターを使用して Red Hat Advanced Cluster Management の最新バージョンにアップグレードするときにも使用します。以下の手順を実行して Operator をアップグレードします。

重要: チャンネルの選択で、新しいバージョンにアップグレード後に、以前のバージョンに戻すことはできません。以前のバージョンを使用するには、Operator をアンインストールし、以前のバージョンで再インストールする必要があります。

1. OpenShift Container Platform 3 の Operator ハブにログインします。
2. OpenShift Container Platform ナビゲーションで、**Operators > Installed Operators** に移動します。
3. **Red Hat Advanced Cluster Management for Kubernetes Operator** を選択します。

4. **Subscription** タブを選択して、サブスクリプション設定を編集します。
5. **Upgrade Status** のラベルが **Up to date** であることを確認します。このステータスは、Operator が、選択したチャンネルで利用可能な最新レベルであることを示します。**Upgrade Status** でアップグレード保留中と示されている場合は、以下の手順を実行して、チャンネルで利用可能な最新のマイナーリリースに更新します。
 - a. **Approval** フィールドの **Manual** 設定をクリックして、値を編集します。
 - b. **Automatic** を選択して自動更新を有効にします。
 - c. **Save** を選択して変更をコミットします。
 - d. 自動更新が Operator に適用されるまで待ちます。更新すると、必要な更新が選択したチャンネルの最新バージョンに自動的に追加されます。更新がすべて完了したら、**Upgrade Status** フィールドに **Up to date** と表示されます。
注記: **MultiClusterHub** カスタムリソースのアップグレードが終了するまで最大 10 分かかかる可能性があります。以下のコマンドを入力して、アップグレードが進行中であるかどうかを確認できます。

```
oc get mch
```

アップグレード中は、**Status** フィールドに **Updating** と表示されます。アップグレードが完了すると、**Status** フィールドに **Running** と表示されます。

6. **Upgrade Status** が **Up to date** になったので、**Channel** フィールドの値をクリックして編集します。
7. 次に利用可能な機能リリースのチャンネルを選択しますが、チャンネルをスキップしようとししないでください。
重要: Operator Lifecycle Manager の **operatorcondition** リソースは、現在のアップグレードプロセス中に以前のアップグレードをチェックし、バージョン抜けを防ぎます。同じリソースのステータスをチェックして、アップグレード可能ステータスが **true** と **false** のどちらか確認できます。
8. **Save** を選択して変更を保存します。
9. 自動アップグレードが完了するまで待ちます。次の機能リリースへのアップグレードが完了すると、チャンネル内の最新のパッチリリースへの更新がデプロイされます。
10. 以降の機能リリースにアップグレードする必要がある場合は、Operator が任意のチャンネルで最新レベルになるまで、手順 7 から 9 を繰り返します。すべてのパッチリリースが最終チャンネルにデプロイされていることを確認します。
11. オプション: チャンネル内の今後の更新を手動で承認させる必要がある場合は、**Approval** 設定を **Manual** に設定できます。

Operator のアップグレードの詳細は、OpenShift Container Platform ドキュメントの [Operator](#) を参照してください。

1.5.1. アップグレードによるクラスタープールの管理

[クラスタープール \(テクノロジープレビュー\) を管理する](#) 場合は、アップグレード後にこれらのクラスタープールの自動管理を停止するために追加の設定が必要になります。

ClusterClaim metadata.annotations に **cluster.open-cluster-management.io/createmanagercluster: "false"** を設定します。

この設定を変更しない限り、既存のクラスター要求はすべて、製品のアップグレード時に自動的にインポートされます。

1.6. 切断されたネットワーク環境でのアップグレード

切断されたネットワーク環境で Red Hat Advanced Cluster Management for Kubernetes をアップグレードする手順と情報を参照してください。

注記: この情報は、[アップグレード](#) のアップグレード手順に従います。その手順を確認してから、次の情報を参照してください。

インストールまたはアップグレード中に、Red Hat Advanced Cluster Management と Multi-Cluster Engine Operator 間の相互依存性に関連する重要な情報が表示される場合があります。インストールまたはアップグレード時の考慮事項については [非接続環境でのインストール](#) を参照してください。

接続されたネットワーク環境でのアップグレードの場合と同様に、Red Hat Advanced Cluster Management for Kubernetes の Operator Lifecycle Manager サブスクリプションのアップグレードチャンネルを新しいリリースのアップグレードチャンネルに変更することで、アップグレードプロセスが開始されます。

ただし、ネットワークに接続されていない環境の特性は特殊であるため、更新チャンネルを変更してアップグレードプロセスを開始する前に、次のミラーリング要件に対処する必要があります。

1. 必要なパッケージがミラーカタログで更新されていることを確認します。
インストール時または以前の更新時に、切断されたネットワーク環境に Red Hat Advanced Cluster Management for Kubernetes をインストールするために必要な Operator パッケージとイメージを含むミラーカタログとレジストリーを作成しています。アップグレードするには、ミラーカタログとレジストリーを更新して、更新されたバージョンの Operator パッケージを取得する必要があります。

インストールアクションと同様に、ミラーカタログとレジストリーに含まれる、または更新される Operator のリストに、次の Operator パッケージが含まれていることを確認する必要があります。

- **advanced-cluster-manager**
- **multicluster-engine**

2. **MulticlusterHub** リソースインスタンスを確認します。
インストール時または以前の更新時に、**MulticlusterHub** リソースのインスタンスを作成し、切断された環境用い、そのリソースに **mce-subscription-spec** アノテーションを追加しています。

ミラーカタログとレジストリーの更手順を行い、**CatalogSource** を使用して OpenShift Container Platform クラスターで利用できるカタログが、以前に使用していた名前と同じになった場合に、**MulticlusterHub** リソースに更新して、**mce-subscriptino-spec** アノテーションを更新する必要はありません。

ただし、ミラーリングされたカタログとレジストリーを更新する手順を実行した結果、新しい名前の **CatalogSource** が作成された場合は、**MulticlusterHub** リソースの **mce-subscription-spec** アノテーションを更新して、新しいカタログソース名を反映させます。

1.6.1. カタログミラーリングによるアップグレード

Red Hat Advanced Cluster Management は、関連する Multi-Cluster Engine Operator 機能を使用して、製品の一部として提供される基本サービスを提供します。Red Hat Advanced Cluster Management は、ハブクラスターのインストールとアップグレードの一環として、必要な Multi-Cluster Engine Operator と **MulticlusterEngine** リソースインスタンスを自動的にインストールして管理します。

接続されたネットワーク環境では、クラスター管理者は、特別なミラーカタログやカタログソースなしで、Red Hat Advanced Cluster Management をインストールまたはアップグレードできます。ただし、(前のセクションで説明したように)切断された環境で Operator Lifecycle Manager Operator をインストールするには、特殊なミラーカタログとカタログソースを使用する必要があるため、インストール後にいくつかの追加手順が必要です。

1. ミラーカタログを作成するための手順を更新します。

Red Hat Advanced Cluster Management のインストール時にミラーリング手順によって Red Hat Operators カタログの完全なコピーが作成された場合は、特別なミラーリング更新は必要ありません。カタログを更新して、新しい Operator リリースの更新されたコンテンツを取得します。

実行した手順でミラーカタログが **フィルタリング** された状態で生成された場合には、ミラーリングの手順を更新して、**advanced-cluster-management** パッケージに加えて、**multicluster-engine** operator パッケージがミラーカタログに含まれていることを確認します。

[ミラーカタログに必要な Operator パッケージを含める](#) トピックを参照してください。これには、ミラーカタログを設定するときに使用するオプションの例が示されています。これらの新しい要件に一致するように、実行した手順で使用される Operator パッケージリストを更新します。

2. **MulticlusterHub** リソースインスタンスを更新します。

[切断されたネットワーク環境でのインストール](#) のトピックで説明されているように、切断された環境でハブクラスターをインストールまたはアップグレードする場合は、**MulticlusterHub** リソースに新しいアノテーションが必要です。

ベストプラクティス: アップグレードを開始するために、Operator Lifecycle Manager サブスクリプションの Operator Lifecycle Manager 更新チャンネルを **advanced-cluster-management** Operator パッケージに変更する前に、**MulticlusterHub** リソースインスタンスを更新して必要なアノテーションを含めます。この更新により、アップグレードを遅滞なく進めることができます。

次の例に示すように、**oc edit** コマンドを使用して **Multiclusterhub** リソースを更新し、**mce-subscription-spec** アノテーションを追加します。

```
metadata:
  annotations:
    installer.open-cluster-management.io/mce-subscription-spec: '{"source": "<my-mirror-catalog-source>"}'
```

例の **<my-mirr-catalog-source>** は、ミラーカタログの **openshift-marketplace** namespace にある **CatalogSource** リソースの名前に置き換えます。

重要: アノテーションを追加する前にアップグレードを開始すると、アップグレードは開始されますが、Operator がバックグラウンドで **multicluster-engine** へのサブスクリプションをインストールしようとするると停止します。この間、**MulticlusterHub** リソースのステータスは引き続き **アップグレード** を表示します。

この問題を解決するには、**oc edit** を実行して、前に示したように **mce-subscription-spec** アノテーションを追加します。

1.7. ポリシーを使用した非接続クラスタのアップグレード

Red Hat OpenShift Update Service と Red Hat Advanced Cluster Management for Kubernetes ポリシーを使用すると、非接続環境で複数のクラスタ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を監視して、クラスタ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. コンソールを使用して、クラスタ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](#)
- [クラスタのアップグレード](#)

1.7.1. 前提条件

OpenShift Update Service を使用して非接続クラスタ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。

- 制限付き OLM が設定された Red Hat OpenShift Container Platform バージョン 4.13 以降で実行されているデプロイ済みハブクラスタ。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。
- ハブクラスタによって管理される OpenShift Container Platform クラスタ

- クラスターイメージをミラーリング可能なローカルレジストリーにアクセスするための認証情報。このリポジトリーの作成方法について、詳細は [非接続インストールミラーリング](#) 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。

1.7.2. 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=4.13.2-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、設定の設定、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。

1.7.3. 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。

1.7.4. 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 xvfz 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、ファイルをデプロイメント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、**graph data 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は、コマンドの **podman** を **docker** に置き換えることもできます。

1.7.5. ミラーリングされたレジストリーの証明書の設定

セキュアな外部コンテナーレジストリーを使用してミラーリングされた OpenShift Container Platform

リリースイメージを保存する場合は、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. キー **updateservice-registry** の下に CA 証明書を追加します。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:
  updateservice-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はデプロイメントを再起動します。

1.7.6. 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> ❶
  replicas: 1
  repository: ${LOCAL_REGISTRY}/ocp4/release
  graphDataImage: '<host_name>:<port>/cincinnati-graph-data-container' ❷

```

❶ ❶ **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。

❷ ❷ **spec.graphDataImage** の値は、グラフデータ init container へのパスに置き換えます。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。

1.7.7. 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。**非推奨: PlacementRule**

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、ポリシーの名前を **policy-mirror** としています。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. コンソールから、**Governance > Create policy** を選択します。
3. **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。
4. YAML コードのコンテンツをすべて削除します。
5. 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-mirror

```

```

namespace: default
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
      apiVersion: policy.open-cluster-management.io/v1
      kind: ConfigurationPolicy
      metadata:
        name: policy-image-content-source-policy
      spec:
        object-templates:
        - complianceType: musthave
          objectDefinition:
            apiVersion: operator.openshift.io/v1alpha1
            kind: ImageContentSourcePolicy
            metadata:
              name: <your-local-mirror-name>
            spec:
              repositoryDigestMirrors:
              - mirrors:
                  - <your-registry> ❶
                source: registry.redhat.io
          ---
        apiVersion: policy.open-cluster-management.io/v1
        kind: PlacementBinding
        metadata:
          name: binding-policy-mirror
          namespace: default
        placementRef:
          name: placement-policy-mirror
          kind: PlacementRule
          apiGroup: apps.open-cluster-management.io
        subjects:
        - name: policy-mirror
          kind: Policy
          apiGroup: policy.open-cluster-management.io
          ---
        apiVersion: apps.open-cluster-management.io/v1
        kind: PlacementRule
        metadata:
          name: placement-policy-mirror
          namespace: default
        spec:
          clusterConditions:
          - status: "True"
            type: ManagedClusterConditionAvailable
          clusterSelector:
            matchExpressions:
            [] # selects all clusters if not specified

```

❶ **your-registry** をローカルミラーリポジトリへのパスに置き換えます。 **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。

6. **Enforce if supported** を選択します。

7. **Create** を選択してポリシーを作成します。

1.7.8. 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. コンソールメニューで、**Governance** > **Create policy** を選択します。
2. **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。
3. **YAML** コードのコンテンツをすべて削除します。
4. 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-catalog
  namespace: default
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
    apiVersion: policy.open-cluster-management.io/v1
    kind: ConfigurationPolicy
    metadata:
      name: policy-catalog
    spec:
      object-templates:
      - complianceType: musthave
        objectDefinition:
          apiVersion: config.openshift.io/v1
          kind: OperatorHub
          metadata:
            name: cluster
          spec:
            disableAllDefaultSources: true
      - complianceType: musthave
        objectDefinition:
          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' 1
            displayName: My Operator Catalog
            publisher: grpc
  ---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-catalog

```



```

namespace: default
placementRef:
  name: placement-policy-catalog
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-catalog
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-catalog
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
    - # selects all clusters if not specified

```

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

5. **Enforce if supported** を選択します。

6. **Create** を選択してポリシーを作成します。

1.7.9. 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。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-cluster-version
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards: null
    policy.open-cluster-management.io/categories: null
    policy.open-cluster-management.io/controls: null
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
    apiVersion: policy.open-cluster-management.io/v1
    kind: ConfigurationPolicy
    metadata:
      name: policy-cluster-version
    spec:
      object-templates:
      - complianceType: musthave
        objectDefinition:
          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 1
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-cluster-version
  namespace: default
placementRef:
  name: placement-policy-cluster-version
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-cluster-version
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
```

```

metadata:
  name: placement-policy-cluster-version
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
    [] # selects all clusters if not specified

```

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

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。
7. **Enforce if supported** を選択します。
 8. **Create** を選択してポリシーを作成します。
 9. 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

```

1.7.10. 利用可能なアップグレードの表示

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

1. Kubernetes Operator コンソールのマルチクラスターエンジンにログインします。
2. ナビゲーションメニューから **Infrastructure > Clusters** を選択します。
3. 状態が **Ready** のクラスターを選択します。
4. **Actions** メニューから **Upgrade cluster** を選択します。
5. オプションのアップグレードパスが利用可能であることを確認します。

注記: 現行バージョンがローカルのイメージリポジトリにミラーリングされていないと、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。

1.7.11. チャネルの選択

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。[チャネルの選択](#)の手順を実行して、アップグレードチャネルを指定します。

1.7.12. 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と、アップグレードは利用できません。

以下の手順を実行してアップグレードします。

1. コンソールで、**Infrastructure > Clusters** を選択します。
2. そのクラスターの内、利用可能なアップグレードがあるかどうかを判断するクラスターを特定します。
3. 利用可能なアップグレードがある場合は、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実行し、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。

1.8. アンインストール

Red Hat Advanced Cluster Management for Kubernetes をアンインストールすると、**カスタムリソースの削除**と**完全な Operator のアンインストール**の2つの異なるアンインストールプロセスのレベルが表示されます。アンインストールプロセスに最長 20 分かかる場合があります。

- 最初のレベルは、カスタムリソースの削除です。これは最も基本的なアンインストールの種類で、**MultiClusterHub** インスタンスのカスタムリソースを削除しますが、他の必要なコンポーネントが残されたままになります。このレベルのアンインストールは、同じ設定とコンポーネントを使用して再インストールする予定の場合に役立ちます。
- 2番目のレベルは、より完全なアンインストールで、カスタムリソース定義などのコンポーネントを除き、ほとんどの Operator コンポーネントを削除します。この手順を続行すると、カスタムリソースの削除で削除されていないコンポーネントおよびサブスクリプションがすべて削除されます。アンインストールが済むと、カスタムリソースの前に Operator を再インストールする必要があります。

1.8.1. 前提条件: 有効化されたサービスのデタッチ

Red Hat Advanced Cluster Management のハブクラスターをアンインストールする前に、ハブクラスターが管理するクラスターをすべてデタッチする必要があります。エラーを解決するには、ハブクラスターが管理しているクラスターをすべてデタッチし、再度アンインストールを試みます。

- Discovery を使用する場合は、アンインストールの試行時に以下のエラーが発生することがあります。

```
Cannot delete MultiClusterHub resource because DiscoveryConfig resource(s) exist
```

Discovery を無効にするには、以下の手順を実行します。

- コンソールから **Discovered Clusters** の表に移動し、**Disable cluster discovery** をクリックします。サービスの削除を確認します。
- ターミナルを使用することもできます。以下のコマンドを実行して Discovery を無効にします。

```
$ oc delete discoveryconfigs --all --all-namespaces
```

- マネージドクラスターがアタッチされている場合は、以下のメッセージが表示される可能性があります。**注記:** これには、自己管理のハブクラスターである **local-cluster** は含まれません。

```
Cannot delete MultiClusterHub resource because ManagedCluster resource(s) exist
```

クラスターのデタッチの詳細は、[クラスター作成の概要](#) でお使いのプロバイダーの情報を選択して、[マネージメントからのクラスターの削除](#) セクションを参照してください。

- 可観測性がある場合は、以下が表示される可能性があります。

```
Cannot delete MultiClusterHub resource because MultiClusterObservability resource(s) exist
```

- ターミナルを使用して **MultiClusterObservability** を無効にして削除するには、以下の手順を参照してください。
 - 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て **MultiClusterObservability** カスタムリソースを削除します。

```
oc delete mco observability
```

- コンソールを使用して **MultiClusterObservability** カスタムリソースを削除するには、以下の手順を参照してください。
 - **MultiClusterObservability** カスタムリソースがインストールされている場合は、**MultiClusterObservability** のタブを選択します。
 - **MultiClusterObservability** カスタムリソースの **Options** メニューを選択します。
 - **Delete MultiClusterObservability** を選択します。
リソースを削除すると、Red Hat Advanced Cluster Management ハブクラスターの **open-cluster-management-observability** namespace の Pod と、全マネージドクラスターの **open-cluster-management-addon-observability** namespace の Pod が削除さ

れます。

注記: 可観測性サービスの削除によるオブジェクトストレージへの影響はありません。

1.8.2. コマンドを使用したリソースの削除

1. まだの場合には、**oc** コマンドが実行できるように、OpenShift Container Platform CLI が設定されていることを確認してください。**oc** コマンドの設定方法の詳細は、OpenShift Container Platform ドキュメントの [OpenShift CLI スタートガイド](#) を参照してください。
2. 以下のコマンドを入力してプロジェクトの namespace に移動します。**namespace** はお使いのプロジェクトの namespace 名に置き換えます。

```
oc project <namespace>
```

3. 以下のコマンドを実行して **MultiClusterHub** カスタムリソースを削除します。

```
oc delete multiclusterhub --all
```

以下のコマンドを入力して進捗を表示できます。

```
oc get mch -o yaml
```

4. clean-up スクリプトを実行して、残っているアーティファクトをすべて削除します。同じクラスター上の古いバージョンの Red Hat Advanced Cluster Management を使用して再インストールする予定がある場合は、このクリーンアップスクリプトを実行します。
 - a. [Helm のインストール](#) の手順に従い、Helm CLI バイナリバージョン 3.2.0 以降をインストールします。
 - b. 以下のスクリプトをファイルにコピーします。

```
#!/bin/bash
ACM_NAMESPACE=<namespace>
oc delete mch --all -n $ACM_NAMESPACE
oc delete apiservice v1.admission.cluster.open-cluster-management.io
v1.admission.work.open-cluster-management.io
oc delete clusterimageset --all
oc delete clusterrole multiclusterengines.multicluster.openshift.io-v1-admin
multiclusterengines.multicluster.openshift.io-v1-crdview
multiclusterengines.multicluster.openshift.io-v1-edit
multiclusterengines.multicluster.openshift.io-v1-view open-cluster-
management:addons:application-manager open-cluster-management:admin-aggregate
open-cluster-management:cert-policy-controller-hub open-cluster-management:cluster-
manager-admin-aggregate open-cluster-management:config-policy-controller-hub open-
cluster-management:edit-aggregate open-cluster-management:iam-policy-controller-hub
open-cluster-management:policy-framework-hub open-cluster-management:view-
aggregate
oc delete crd klusterletaddonconfigs.agent.open-cluster-management.io
placementbindings.policy.open-cluster-management.io policies.policy.open-cluster-
management.io userpreferences.console.open-cluster-management.io
discoveredclusters.discovery.open-cluster-management.io
discoveryconfigs.discovery.open-cluster-management.io
oc delete mutatingwebhookconfiguration ocm-mutating-webhook
managedclustermutators.admission.cluster.open-cluster-management.io multicluster-
```

```

observability-operator
oc delete validatingwebhookconfiguration
channels.apps.open.cluster.management.webhook.validator application-webhook-
validator multiclusterhub-operator-validating-webhook ocm-validating-webhook
multicluster-observability-operator multiclusterengines.multicluster.openshift.io

```

スクリプトの **<namespace>** は、Red Hat Advanced Cluster Management がインストールされている namespace 名に置き換えます。

重要: namespace が消去され削除されるため、正しい namespace を指定するようにしてください。

- a. スクリプトを実行して、以前のインストールから残ったままとなっているすべてのアーティファクトを削除します。残っているアーティファクトがない場合は、リソースが見つからなかったことを示すメッセージが返されます。

注記: 同じ Red Hat Advanced Cluster Management バージョンを再インストールする予定の場合は、この手順の次のステップを省略して、カスタムリソースを再インストールします。完全な Operator のアンインストールに進みます。

1. 以下のコマンドを入力して、インストールされている namespace で Red Hat Advanced Cluster Management **ClusterServiceVersion** および **Subscription** を削除します。**2.x.0** の値を現在のメジャーリリースまたはマイナーリリースに置き換えます。

```

> oc get csv
NAME                                DISPLAY                                VERSION REPLACES PHASE
advanced-cluster-management.v2.x.0  Advanced Cluster Management for Kubernetes  2.x.0
Succeeded

> oc delete clusterserviceversion advanced-cluster-management.v2.x.0

> oc get sub
NAME                                PACKAGE                                SOURCE                                CHANNEL
acm-operator-subscription  advanced-cluster-management  acm-custom-registry  release-2.x

> oc delete sub acm-operator-subscription

```

注記: CSV のサブスクリプションおよびバージョンの名前が異なる場合があります。

1.8.3. コンソールを使用したコンポーネントの削除

Red Hat OpenShift Container Platform コンソールを使用してアンインストールする場合に、operator を削除します。コンソールを使用してアンインストールを行うには、以下の手順を実行します。

1. OpenShift Container Platform コンソールのナビゲーションで、**Operators > Installed Operators > Advanced Cluster Manager for Kubernetes** を選択します。
2. **MultiClusterHub** のカスタムリソースを削除します。
 - a. **Multiclusterhub** のタブを選択します。
 - b. MultiClusterHub カスタムリソースの **Options** メニューを選択します。
 - c. **Delete MultiClusterHub** を選択します。
3. [コマンドを使用した MultiClusterHub インスタンスの削除](#) の手順にしたがって、クリーンアップスクリプトを実行します。

注記: 同じ Red Hat Advanced Cluster Management バージョンを再インストールする場合は、残りの手順を省略し、カスタムリソースを再インストールします。

4. **Installed Operators** に移動します。
5. **Options** メニュー、**Uninstall operator** の順に選択して、**Red Hat Advanced Cluster Management operator** を削除します。