



Red Hat Advanced Cluster Management for Kubernetes 2.5

インストール

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

Red Hat Advanced Cluster Management for Kubernetes 2.5 インストール

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

法律上の通知

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

Linux[®] 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をご覧ください。

目次

第1章 インストール	3
1.1. 要件および推奨事項	3
1.2. パフォーマンスおよびスケーラビリティ	7
1.3. ネットワーク接続時のオンラインインストール	15
1.4. ネットワーク切断状態でのインストール	22
1.5. MULTICLUSTERHUB 詳細設定	25
1.6. ネットワーク設定	30
1.7. OPERATOR を使用したアップグレード	37
1.8. OPENSIFT CONTAINER PLATFORM のアップグレード	39
1.9. アンインストール	40

第1章 インストール

Red Hat Advanced Cluster Management for Kubernetes のインストールおよびアンインストールの方法を説明します。Red Hat Advanced Cluster Management for Kubernetes を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。Red Hat Advanced Cluster Management for Kubernetes は、サポート対象の Red Hat OpenShift Container Platform を使用して Linux 上にオンライン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. カタログから Red Hat Advanced Cluster Management for Kubernetes の Operator 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。2.3 の場合は、この一覧に FIPS 承認 **されていない** 暗号が 2 つ含まれます。バージョン 2.3.x 以前からアップグレードし、FIPS コンプライアンスが必要な場合は、**multiclusterhub** リソースから、以下の 2 つの暗号 (**ECDHE-ECDSA-CHACHA20-POLY1305** および **ECDHE-RSA-CHACHA20-POLY1305**) を削除します。

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。

- [要件および推奨事項](#)
- [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](#)
- [ネットワーク設定](#)
- [Operator を使用したアップグレード](#)
- [OpenShift Container Platform のアップグレード](#)
- [アンインストール](#)

1.1. 要件および推奨事項

Red Hat Advanced Cluster Management for Kubernetes をインストールする前に、以下のシステム設定の要件およびオプションを確認します。

- [サポート対象のオペレーティングシステムおよびプラットフォーム](#)
- [サポート対象のブラウザー](#)

1.1.1. サポート対象のオペレーティングシステムおよびプラットフォーム

ハブクラスターおよびマネージドクラスタープラットフォームに関する最近の情報を確認するには、[Red Hat Advanced Cluster Management 2.5 サポートマトリックス](#) を参照してください。

1.1.2. サポート対象のブラウザ

Red Hat Advanced Cluster Management コンソールには、Mozilla Firefox、Google Chrome、Microsoft Edge、および Safari からアクセスできます。以下は、テスト済みでサポートされるバージョンです。

プラットフォーム	サポート対象のブラウザ
Microsoft Windows	Microsoft Edge: 44 以降、Mozilla Firefox: 82.0 以降、Google Chrome: バージョン 86.0 以降
Linux	Mozilla Firefox: 82.0 以降、Google Chrome: バージョン 86.0 以降
macOS	Mozilla Firefox: 82.0 以降、Google Chrome: バージョン 86.0 以降、Safari: 14.0 以降

1.1.3. ネットワーク設定

以下のセクションで接続を許可するようにネットワークを設定します。

1.1.3.1. ハブクラスターネットワークの要件

ハブクラスターネットワークの要件については、以下の表を参照してください。

方向	Protocol	接続	ポート (指定されている場合)
マネージドクラスターへの送信	HTTPS	マネージドクラスターの Pod の Search コンソールから動的にログを取得する。この接続は、マネージドクラスターの open-cluster-management-agent-addon namespace に klusterlet-addon-workmgr という名前のルートを作成します。ルートのホストは <route name>-<namespace>.apps.<cluster domain> です。	443

方向	Protocol	接続	ポート (指定されている場合)
マネージドクラスターへの送信	HTTPS	Klusterlet をインストールするために、インストール時にプロビジョニングされるマネージドクラスターの Kubernetes API サーバー	6443
チャンネルソースへの送信	HTTPS	GitHub、オブジェクトストア、Helm リポジトリなどのチャンネルソース。これは、アプリケーションライフサイクル、OpenShift GitOps または ArgoCD を使用してこれらのソースに接続する場合にのみ必要です。	443
マネージドクラスターからの受信	HTTPS	メトリクスおよびアラートをプッシュするマネージドクラスター (OpenShift Container Platform バージョン 4.8 以降を実行するマネージドクラスターに対してのみアラートが収集されます)	443
マネージドクラスターからの受信	HTTPS	マネージドクラスターから変更を監視するハブクラスターの kube API Server	6443
ObjectStore への送信	HTTPS	ObjectStore での長期保管用に、 あるいは Cluster Backup Operator が実行されている場合に、Observability のメトリクスデータを送信します。	443
イメージリポジトリへの送信	HTTPS	OpenShift Container Platform および Red Hat Advanced Cluster Management のイメージにアクセスします	443

1.1.3.2. マネージドクラスターネットワークの要件

注記: マネージドクラスターの **Registration Agent** および **Work Agent** は、プロキシを通過できない mTLS 接続の確立によりハブクラスターの **apiserver** と通信するため、プロキシ設定をサポートしません。

マネージドクラスターネットワークの要件については、以下の表を参照してください。

方向	Protocol	接続	ポート (指定されている場合)
ハブクラスターからの受信	HTTPS	マネージドクラスターの Pod に対して動的にログを送信します。この接続は、 klusterlet-addon-workmgr という名前のマネージドクラスターで実行されているサービスを使用します。	443
ハブクラスターからの受信	HTTPS	Klusterlet をインストールするために、インストール時にプロビジョニングされるマネージドクラスターの Kubernetes API サーバー	6443
イメージリポジトリへの送信	HTTPS	OpenShift Container Platform および Red Hat Advanced Cluster Management のイメージにアクセスします	443
ハブクラスターへの送信	HTTPS	メトリクスおよびアラートをプッシュするマネージドクラスター (OpenShift Container Platform バージョン 4.8 以降を実行するマネージドクラスターに対してのみアラートが収集されます)	443
ハブクラスターへの送信	HTTPS	ハブクラスターの Kubernetes API サーバーで変更の有無を監視します。	6443

方向	Protocol	接続	ポート (指定されている場合)
チャンネルソースへの送信	HTTPS	GitHub、オブジェクトストア、Helm リポジトリなど、チャンネルソースへのマネージドクラスター。この送信接続は、アプリケーションライフサイクルを使用してこのようなソースに接続する場合にのみ必要です。	443
ハブクラスターへの送信	HTTPS	マネージドクラスターのクラスタープロキシアドオンを登録する場合。	443

1.2. 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、特定のスケー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ーラビリティと検索パフォーマンスです。

この情報を使用すると、お使いの環境のプランニングに役立ちます。

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。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](#)
- [可観測性のスケーリング](#)

1.2.1. マネージドクラスターの最大数

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

- クラスター内のリソース数。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、ハブクラスターの設定。

以下の表は、今回のテストに使用した Amazon Web Services クラウドプラットフォームのクラスターの設定情報を示しています。

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

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	m5.2xlarge	8	32	gp2	100	3	us-east-1
ワーカー または インフラストラクチャー	m5.2xlarge	8	32	gp2	100	ノード3つまたは5つ	us-east-1

インフラストラクチャーノードの詳細は、[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。 [インフラストラクチャーマシンの作成](#) も参照してください。

1.2.2. 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。検索パフォーマンスの分析には、以下の変数が重要です。

- 物理メモリー
- 書き込みスループット (キャッシュのリカバリー時間)
- クエリー実行時間

1.2.2.1. 物理メモリー

検索は、データをインメモリーに保持し、応答時間を早めます。必要なメモリーは、クラスター内の Kubernetes リソース数とその関係に比例します。

クラスター	Kubernetes リソース	関係	確認済みのサイズ (シミュレーションデータあり)
medium 1 台	5000	9500	50 Mi
medium 5 台	25,000	75,000	120 Mi
medium 15 台	75,000	20,0000	492 Mi
medium 30 台	150,000	450,000	1 Gi
medium 50 台	250,000	750,000	2 Gi

検索コンポーネントに使用するメモリー量を変更する方法については、[再ディスクグラフメモリーを増やすためのオプション](#) を参照してください。

1.2.2.2. 書き込みスループット (キャッシュのリカバリー時間)

安定状態のクラスターの多くは、少数のリソース更新を生成します。RedisGraph データの消去時には、更新の割合が高くなり、その結果、ほぼ同時にリモートのコレクターが完全な状態を同期します。データストアの消去時に、さまざまな数のマネージドクラスターの復元時間が測定されます。

クラスター	Kubernetes リソース	関係	シミュレーションからの平均リカバリー時間
medium 1 台	5000	9500	2 秒未満
medium 5 台	25,000	75,000	15 秒未満
medium 15 台	75,000	200,000	2 分 40 秒
medium 30 台	150,000	450,000	5 ~ 8 分

注記: ハブへのネットワーク接続の速度が遅いクラスターの場合は、所要時間が伸びる可能性があります。前述の書き込みスループットの情報は、**persistence** が無効の場合にのみ適用されます。

1.2.2.3. クエリー実行に関する考慮事項

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。
多数のクラスター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。

1.2.3. 可観測性のスケーリング

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。

1.2.3.1. 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、以下のトポロジー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クス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。

1.2.3.2. 書き込みスループット

Pod	間隔 (分)	時系列 (分)
400	1	83000

1.2.3.3. 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 クラスター	400
20 x クラスター	800

1.2.3.4. 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` のメトリクスから取得。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 クラスター	9.84	4.93
20 x クラスター	13.10	8.76

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

重要: メトリクスは、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 クラスター	2
20 x クラスター	3

1.2.3.6. 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

1.2.3.7. Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3) の合計使用量は増加します。メトリクスデータは、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

1.2.4. ハブ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。通常のクラスターには、3つ以上のアベイラビリティゾーンでほぼ同等の容量のワーカーノードが必要です。
- 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 のコンピュータ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。
- メモリーの予約と制限。コンテナに割り当てるワーカーノードのメモリー容量を確立します。
- 製品によって管理され、Kubernetes が使用する etcd クラスターに保存される永続データ。

重要: OpenShift Container Platform では、3つのアベイラビリティゾーンにクラスターのマスターノードを分散させてください。

1.2.4.1. 製品環境

注記: 以下の要件は、最小要件ではありません。

表1.1 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。

1.2.4.1.1. Amazon Web Services 上の OpenShift Container Platform

詳細は、[OpenShift Container Platform 製品ドキュメントの Amazon Web Services の情報](#) を参照してください。

また、[マシンタイプ](#) についても確認してください。

- ノード数: 3

- アベイラビリティゾーン: 3
- インスタンスサイズ: m5.xlarge
 - vCPU: 4
 - メモリー: 16 GB
 - ストレージサイズ: 120 GB

1.2.4.1.2. Google Cloud Platform での OpenShift クラスタ

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。

また、[マシンタイプ](#) についても確認してください。

- ノード数: 3
- アベイラビリティゾーン: 3
- インスタンスサイズ: N1-standard-4 (0.95–6.5 GB)
 - vCPU: 4
 - メモリー: 15 GB
 - ストレージサイズ: 120 GB

1.2.4.1.3. Microsoft Azure での OpenShift クラスタ

詳細は、以下の [製品ドキュメント](#) を参照してください。

- ノード数: 3
- アベイラビリティゾーン: 3
- インスタンスサイズ: Standard_D4_v3
 - vCPU: 4
 - メモリー: 16 GB
 - ストレージサイズ: 120 GB

1.2.4.1.4. VMware vSphere でのクラスタ

詳細は、以下の [製品ドキュメント](#) を参照してください。

- ノード数: 3
- アベイラビリティゾーン: 3
- インスタンスサイズ:
 - メモリー: 16 GB
 - ストレージサイズ: 120 GB

- vCPUs: 4
- ソケットごとのコア: 2

1.2.4.1.5. IBM Z システムの OpenShift Container Platform

詳細は、OpenShift Container Platform ドキュメントの [クラスターの IBM Z システムへのインストール](#) を参照してください。

- ノード数: 3
- アベイラビリティゾーン: 3
- インスタンスサイズ:
 - メモリー: 16 GB
 - ストレージサイズ: 100 GB

- vCPU: 10

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。SMT を設定している場合は、1つの物理コア (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 を提供できません。

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。これが有効になっていると、数式 (コアごとのスレッド × コア数) × ソケット数 = vCPU を使用して対応する比率を計算します。

SMT の詳細は、[Simultaneous multithreading](#) を参照してください。

1.2.4.1.6. IBM Power Systems 上の OpenShift Container Platform

詳細は、OpenShift Container Platform ドキュメントの [クラスターの Power システムへのインストール](#) を参照してください。

- ノード数: 3
- アベイラビリティゾーン: 3
- インスタンスサイズ:
 - メモリー: 16 GB
 - ストレージサイズ: 120 GB

- vCPU: 16

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。

- SMT-8 (IBM PowerVM を実行しているシステムのデフォルト設定) 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。

1.2.4.1.7. ベアメタルアセット上の OpenShift Container Platform クラスター

詳細は、以下の [製品ドキュメント](#) を参照してください。

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トポロジーで実行できます。

- ノード数: 3
- アベイラビリティゾーン: 3
- インスタンスサイズ:
 - メモリー: 16 GB
 - ストレージサイズ: 120 GB
 - vCPUs: 4

1.2.4.1.8. 単一ノードの OpenShift Container Platform クラスターの作成および管理

2200 の単一ノードの OpenShift Container Platform クラスターの作成および管理の要件例を参照してください。Red Hat Advanced Cluster Management を使用して単一ノード OpenShift (SNO) 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。

- マスター (スケジュール可能)
 - ノード数: 3
 - メモリー: 289 GB (クラスターの最大値)
 - メモリー: 110 GB (単一ノードの最大値)
 - CPU クラスターの最大値: 90
 - CPU 単一ノードの最大値: 44

注記: 複数のクラスターが同時に作成された間、CPU 使用率の値はピーク化されます。

1.3. 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。

作業を開始する前に、[要件および推奨事項](#) セクションを参照してから、以下のドキュメントを確認してください。

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については、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](#)

1.3.1. 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 にアクセスできる必要があります。
- 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 4.8 以降をデプロイし、OpenShift Container Platform CLI でログインしている。お使いの環境に OpenShift Container Platform バージョン 4.8 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 4.10 の以前のバージョンに変更します。
- 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、2.5 の klusterlet Operator の **stable-2.0** チャンネルを使用する必要があります。

1.3.2. 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。

1.3.3. 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 の一覧にアクセスし、**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、サブスクライブ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。これで、Red Hat Advanced Cluster Management ハブクラスターのコンソールにアクセスできるようになりました。以下の手順を参照してください。
6. OpenShift Container Platform コンソールナビゲーションで **Networking** > **Routes** を選択します。

7. リストで Red Hat Advanced Cluster Management ハブクラスターの URL を確認して、その URL に移動してコンソールにアクセスします。

1.3.4. 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>
spec:
  targetNamespaces:
  - <namespace>
```

4. 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
```

5. 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のようになります。

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: acm-operator-subscription
spec:
  sourceNamespace: openshift-marketplace
  source: redhat-operators
  channel: release-2.5
  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}'
```

- ステータスが **Running** になってから、ルートの一覧を確認してルートを探し出します。

```
oc get routes
```

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。

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

OpenShift Container Platform クラスタを、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。

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

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います。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
```

1.3.5.2. 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
```

1.3.5.3. 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: ""
```

1.4. ネットワーク切断状態でのインストール

インターネットに接続していない Red Hat OpenShift Container Platform クラスターに Red Hat Advanced Cluster Management for Kubernetes をインストールしないといけない場合があります。ネットワーク接続のないハブにインストールする手順でも一部、オンラインインストールと同じ手順が必要になります。

インストール時にネットワークから直接パッケージにアクセスするのではなく、パッケージをダウンロードしておき、インストール時にアクセスできるようにする必要があります。

作業を開始する前に、[要件および推奨事項](#) セクションを参照してから、以下のドキュメントを確認してください。

- [前提条件](#)
- [OpenShift Container Platform インストールの確認](#)
- [非接続環境でのインストール](#)

1.4.1. 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。

- お使いの環境に Red Hat OpenShift Container Platform バージョン 4.8 以降をインストールし、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。
- [catalog.redhat.com](#) へのアクセスがある。
注記: ベアメタルクラスターを管理する場合は、Red Hat OpenShift Container Platform バージョン 4.8 以降が必要です。

[OpenShift Container Platform バージョン 4.10](#)、[OpenShift Container Platform バージョン 4.10](#) を参照してください。

- Red Hat OpenShift Container Platform の CLI バージョンは 4.8 以降を使用し、**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 Red Hat 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に失敗します。
- Operator の依存関係をダウンロードするために、インターネット接続のあるワークステーションが必要。

1.4.2. 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 バージョンがクラスターにインストールされ、機能する状態である必要があります。OpenShift Container Platform バージョン 4.10 の詳細は、[OpenShift Container Platform ドキュメント](#) を参照してください。

- 接続されている場合は、**kubectl -n openshift-console get route** コマンドを実行して、OpenShift Container Platform の 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
```

この例のコンソール URL は **https:// console-openshift-console.apps.new-coral.purple-chesterfield.com** です。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。

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](#) を参照してください。

1.4.3. 非接続環境でのインストール

重要: 必要なイメージをミラーリングレジストリーにダウンロードし、非接続環境で Operator をインストールする必要があります。ダウンロードがないと、デプロイメント時に **ImagePullBackOff** エラーが表示される可能性があります。

以下の手順を実行して Red Hat Advanced Cluster Management を非接続環境でインストールします。

1. ミラーレジストリーを作成します。ミラーレジストリーがない場合は、Red Hat OpenShift Container Platform ドキュメントの [非接続インストールのイメージのミラーリング](#) の手順を実行してミラーレジストリーを作成してください。
ミラーレジストリーがすでにある場合は、既存のレジストリーを設定して使用できます。
- 注記:** [ミラーリングされた Operator カタログからの OperatorHub の入力](#) に記載の OpenShift Container Platform ドキュメントの手順に従うようにしてください。
2. Operator カタログのミラーリングを行います。[非接続クラスターで使用する Operator カタログのミラーリング](#) の手順に従い、Operator カタログがミラーリングされるようにします。

注記:

- 既存の Red Hat Operator インデックスイメージからパッケージをプルーニングする場合は、**advanced-cluster-management** パッケージおよび **multicluster-engine** パッケージの両方がプルーニングされていることを確認してください。詳細は、[SQLite ベースのインデックスイメージのフィルターリング](#) を参照してください。
- [生成されたマニフェスト](#) のプロセス中に、マニフェストディレクトリーに **catalogSource.yaml** ファイルを生成します。切断された Operator Lifecycle Manager を設定するときに、このサンプルファイルを使用します。
- ベアメタルの場合のみ、**install-config.yaml** ファイルに、接続なしのレジストリーの証明書情報を指定する必要があります。保護されたオフラインレジストリーでイメージにアクセスするには、Red Hat Advanced Cluster Management がレジストリーにアクセスできるように証明書情報を指定する必要があります。
 - a. レジストリーから証明書情報をコピーします。
 - b. エディターで **install-config.yaml** ファイルを開きます。
 - c. **additionalTrustBundle:** | のエントリーを検索します。

- d. **additionalTrustBundle** の行の後に証明書情報を追加します。コンテンツの結果は以下の例のようになります。

```
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  certificate_content
  -----END CERTIFICATE-----
sshKey: >-
```

重要: 以下のガバナンスポリシーが必要な場合は、非接続イメージレジストリーの追加ミラーが必要です。

- Container セキュリティー Operator ポリシー: イメージはソース **registry.redhat.io/quay** にあります。
 - Compliance Operator ポリシー: イメージはソース **registry.redhat.io/compliance** にあります。
 - Gatekeeper Operator ポリシー: イメージはソース **registry.redhat.io/rhacm2** にあります。
- 3つのすべての Operator については、以下のミラー一覧を参照してください。

```
- mirrors:
  - <your_registry>/rhacm2
  source: registry.redhat.io/rhacm2
- mirrors:
  - <your_registry>/quay
  source: registry.redhat.io/quay
- mirrors:
  - <your_registry>/compliance
  source: registry.redhat.io/compliance
```

1. **install-config.yaml** ファイルを保存します。
2. **rhacm-policy.yaml** という名前の **ImageContentSourcePolicy** を含めて yaml ファイルを作成します。**注記:** 実行中のクラスターでこれを変更すると、すべてのノードのローリング再起動が実行されます。

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: rhacm-repo
spec:
  repositoryDigestMirrors:
    - mirrors:
      - mirror.registry.com:5000/rhacm2
    source: registry.redhat.io/rhacm2
```

3. 以下のコマンドを入力して **ImageContentSourcePolicy** ファイルを適用します。

```
oc apply -f rhacm-policy.yaml
```

4. ネットワーク接続されていない Operator Lifecycle Manager の Red Hat Operator と コミュニティーの Operator を有効にします。Red Hat Advanced Cluster Management は Operator Lifecycle Manager Red Hat Operator カタログに含まれ

ます。

5. Red Hat Operator カタログの非接続 Operator Lifecycle Manager を設定します。Red Hat OpenShift Container Platform ドキュメントの [ネットワークが制限された環境での Operator Lifecycle Manager の使用](#) の手順を実行します。
 - [クラスターへのカタログソースの追加](#) 手順では、オペレーターカタログをミラーリングするときに作成した **catalogSource.yaml** ファイルを使用します。
 - 独自の **catalogSource.yaml** ファイルを使用していて、カタログソース名が予想される **redhat-operator-index** と異なる場合は、**my-operator-catalog** の代わりにカタログソースを使用して **MultiClusterHub** カスタムリソースに次のアノテーションを追加する必要があります。

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

非接続 Operator Lifecycle Manager にイメージが設定されたので、Operator Lifecycle Manager カタログからの Red Hat Advanced Cluster Management for Kubernetes のインストールを続行してください。

必要な手順については、[ネットワーク接続時のオンラインインストール](#) を参照するか、[インストール](#) の概要に戻ります。

1.5. 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。Red Hat Advanced Cluster Management は、以下の1つ以上の属性を **MultiClusterHub** カスタムリソースに追加することで、インストール時にまたはインストール後にさらに設定できます。

1.5.1. 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はありません。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>
```

1.5.2. 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。

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"
```

1.5.3. 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: ""
```

1.5.4. tolerations

容認の一覧を定義して、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。

1.5.5. 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。

Red Hat Advanced Cluster Management ハブクラスターが自己管理しない場合は、**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
```

1.5.6. 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ように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一覧を作成できます。

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
```

1.5.7. customCAConfigmap

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>
```

1.5.8. sslCiphers

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** を一覧表示するの使用するデフォルトの **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"
```


1.5.9. ClusterProxyAddon (テクノロジープレビュー)

以下の例は、**cluster-proxy-addon** を有効にするのに使用されるデフォルトの **spec.overrides** 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-proxy-addon
        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-proxy-addon", "enabled": true}}]'
```

enableClusterProxyAddon フィールドの使用はサポートされなくなり、上記に置き換えられました。

1.5.10. 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}}]'
```

1.5.11. ManagedServiceAccount アドオン (テクノロジープレビュー)

以下の例は、**ManagedServiceAccount** を有効にするために使用される **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: managedserviceaccount-preview
        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": "managedserviceaccount-preview", "enabled": true}}]'
```

1.5.12. hypershift アドオン (テクノロジープレビュー)

以下の例は、**Hypershift** 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: hypershift-preview
        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": "hypershift-preview", "enabled": true}}]'
```

1.6. ネットワーク設定

ハブクラスターとマネージドクラスターネットワークの設定、および追加のネットワーク情報を参照できます。

- [ハブクラスターのネットワーク設定表](#)
- [マネージドクラスターのネットワーク設定表](#)
- [インフラストラクチャーオペレータテーブルの追加のネットワーク要件](#)
- [Submariner networking requirements table](#)
- [Hive テーブルの追加のネットワーク要件](#)
- [アプリケーションデプロイメントのネットワーク要件表](#)
- [namespace 接続のネットワーク要件表](#)

1.6.1. ハブクラスターのネットワーク設定表

次の表のハブクラスターネットワーク要件を参照してください。

方向	Protocol	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
マネージドクラスターへのアウトバウンド	HTTPS	マネージドクラスターの Pod のログを Search コンソールから動的に取得し、マネージドクラスターで実行している klusterlet-addon-workmgr サービスを使用します。	443	なし	マネージドクラスタールートにアクセスするための IP アドレス
マネージドクラスターへのアウトバウンド	HTTPS	klusterlet をインストールするために、インストール時にプロビジョニングされるマネージドクラスターの Kubernetes API サーバー	6443	なし	Kubernetes マネージドクラスター API サーバーの IP
チャンネルソースへの送信	HTTPS	アプリケーションライフサイクル、OpenShift GitOps、または ArgoCD を使用して接続する場合にのみ必要となる、GitHub、Object Store、および Helm リポジトリを含むチャンネルソース	443	なし	チャンネルソースの IP

方向	Protocol	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
マネージドクラスターからの受信	HTTPS	メトリクスおよびアラートをプッシュするマネージドクラスターは、OpenShift Container Platform バージョン 4.8 以降を実行するマネージドクラスターに対してのみアラートが収集されます	443	なし	ハブクラスターアクセスルートへの IP アドレス
マネージドクラスターからの受信	HTTPS	マネージドクラスターからの変更を監視するハブクラスターの Kubernetes API サーバー	6443	なし	ハブクラスター Kubernetes API サーバーの IP アドレス
ObjectStore へのアウトバウンド	HTTPS	Cluster Backup Operator の実行時に、長期保存用の可観測性メトリクスデータを送信します。	443	なし	ObjectStore の IP アドレス
イメージリポジトリへのアウトバウンド	HTTPS	OpenShift Container Platform および Red Hat Advanced Cluster Management のイメージにアクセスします	443	なし	イメージリポジトリの IP アドレス

1.6.2. マネージドクラスターのネットワーク設定表

注記: マネージドクラスターの **Registration Agent** および **Work Agent** は、プロキシを通過できない mTLS 接続の確立によりハブクラスターの **apiserver** と通信するため、プロキシ設定をサポートしません。

次の表のマネージドクラスターネットワーク要件を参照してください。

方向	Protocol	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
ハブクラスターからの受信	HTTPS	マネージドクラスターの Pod の Search コンソールからログを動的に送信するには、マネージドクラスターで実行している klusterlet-addon-workmgr サービスを使用します。	443	なし	マネージドクラスタールートにアクセスするための IP アドレス
ハブクラスターからの受信	HTTPS	klusterlet をインストールするために、インストール時にプロビジョニングされるマネージドクラスターの Kubernetes API サーバー	6443	なし	Kubernetes マネージドクラスター API サーバーの IP
イメージリポジトリへのアウトバウンド	HTTPS	OpenShift Container Platform および Red Hat Advanced Cluster Management のイメージにアクセスします	443	なし	イメージリポジトリの IP アドレス

方向	Protocol	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
ハブクラスターへの送信	HTTPS	メトリクスおよびアラートをプッシュするマネージドクラスターは、OpenShift Container Platform バージョン 4.8 以降を実行するマネージドクラスターに対してのみアラートが収集されます	443	なし	ハブクラスターアクセスルートへの IP アドレス
ハブクラスターへの送信	HTTPS	ハブクラスターの Kubernetes API サーバーで変更の有無を監視します。	6443	なし	ハブクラスター Kubernetes API サーバーの IP アドレス
チャンネルソースへの送信	HTTPS	アプリケーションライフサイクル、OpenShift GitOps、または ArgoCD を使用して接続する場合にのみ必要となる、GitHub、Object Store、および Helm リポジトリを含むチャンネルソース	443	なし	チャンネルソースの IP

1.6.3. インフラストラクチャーオペレーターテーブルの追加のネットワーク要件

Infrastructure Operator を使用してベアメタルマネージドクラスターをインストールする場合は、以下の表で追加のネットワーク要件について参照してください。

方向	Protocol	接続	ポート (指定されている場合)
ISO/rootfs イメージリポジトリへのハブクラスターの送信	HTTPS (非接続環境では HTTP)	Red Hat Advanced Cluster Management ハブで ISO イメージを作成するのに使用します。	443 (非接続環境では 80)
単一ノードの OpenShift Container Platform マネージドクラスターでの BMC インターフェイスへのハブクラスター送信	HTTPS (非接続環境では HTTP)	OpenShift Container Platform クラスターをブートします。	443
OpenShift Container Platform マネージドクラスターからハブクラスターへの送信	HTTPS	assistedService ルートを使用してハードウェア情報を報告します。	443
OpenShift Container Platform マネージドクラスターから ISO/rootfs イメージリポジトリへの送信	HTTP	rootfs イメージをダウンロードします。	80

1.6.4. Submariner のネットワーク要件表

Submariner を使用するクラスターに対して、ポートを 3 つ開放する必要があります。以下の表は、どのポートを使用できるかを示しています。

方向	Protocol	接続	ポート (指定されている場合)
送信および受信	UDP	各マネージドクラスター	4800
送信および受信	UDP	各マネージドクラスター	4500、500、およびゲートウェイノード上の IPsec トラフィックに使用されるその他のポート
受信	TCP	各マネージドクラスター	8080

1.6.5. Hive テーブルの追加のネットワーク要件

Central Infrastructure Management の使用が含まれる Hive Operator を使用してベアメタルマネージドクラスターをインストールする場合は、ハブクラスターと **libvirt** プロビジョニングホスト間で、レイヤー 2 またはレイヤー 3 のポート接続を設定する必要があります。プロビジョニングホストへのこの接続は、Hive を使用したベースベアメタルクラスターの作成時に必要になります。詳細は、以下の表を参照してください。

方向	Protocol	接続	ポート (指定されている場合)
libvirt プロビジョニング ゲストへのハブクラスターの送信および受信	IP	Hive Operator がインストールされているハブクラスターを、ベアメタルクラスターの作成時にブートストラップとして機能する libvirt プロビジョニングホストに接続します。	

注記:これらの要件はインストール時にのみ適用され、Infrastructure Operator でインストールされたクラスターのアップグレード時には必要ありません。

1.6.6. アプリケーションデプロイメントのネットワーク要件表

通常、アプリケーションデプロイメントの通信は、マネージドクラスターからハブクラスターへの1つの方法です。接続は、マネージドクラスターのエージェントによって設定される **kubeconfig** を使用します。マネージドクラスターでのアプリケーションデプロイメントは、ハブクラスターの以下の名前空間にアクセスする必要があります。

- チャネルリソースの名前空間
- マネージドクラスターの名前空間

1.6.7. namespace 接続のネットワーク要件表

- アプリケーションライフサイクル接続:
 - namespace の **open-cluster-management** は、ポート 4000 のコンソール API にアクセスする必要があります。
 - namespace の **open-cluster-management** は、ポート 3001 でアプリケーション UI を公開する必要があります。
- アプリケーションライフサイクルバックエンドコンポーネント (Pod):
ハブクラスターで、アプリケーションライフサイクル Pod はすべて、以下の Pod を含む **open-cluster-management** namespace にインストールされます。
 - multicluster-operators-hub-subscription
 - multicluster-operators-standalone-subscription
 - multicluster-operators-channel
 - multicluster-operators-application
 - multicluster-integrations
これらの Pod が **open-cluster-management** namespace に作成されると、以下のようになります。
 - namespace の **open-cluster-management** は、ポート 6443 で Kube API にアクセスする必要があります。

マネージドクラスターでは、**klusterlet-addon-appmgr** アプリケーションライフサイクル Pod のみが **open-cluster-management-agent-addon** namespace にインストールされます。

- namespace **open-cluster-management-agent-addon** は、ポート 6443 で Kube API にアクセスする必要があります。
- ガバナンスおよびリスク:
ハブクラスターでは、以下のアクセスが必要です。
 - namespace の **open-cluster-management** は、ポート 6443 で Kube API にアクセスする必要があります。
 - namespace **open-cluster-management** は、ポート 5353 で OpenShift DNS にアクセスする必要があります。

マネージドクラスターでは、以下のアクセスが必要です。

- namespace **open-cluster-management-addon** は、ポート 6443 の Kube API にアクセスする必要があります。

詳細は、[Red Hat Advanced Cluster Management for Kubernetes 2.5 Support Matrix](#) を参照してください。

1.7. OPERATOR を使用したアップグレード

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 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:** インストールする製品のバージョンに合わせます。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、または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。

operator を使用して Red Hat Advanced Cluster Management をアップグレードする場合にも、上記の設定を使用します。

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

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。
ヒント: 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。**非推奨: release-2.4** および **release-2.3** チャンネルは更新を受信しません。インポートするには、2.5 の klusterlet Operator の **stable-2.0** チャンネルを使用する必要があります。アップグレード時は、チャンネルをスキップできません。たとえば、バージョン 2.2.z から 2.4 をスキップすることはできません。
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。

Red Hat Advanced Cluster Management は、選択したチャンネルの最新バージョンで稼働しています。

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。

1.7.1. 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。

1.8. OPENSIFT CONTAINER PLATFORM のアップグレード

Red Hat Advanced Cluster Management for Kubernetes ハブクラスタをホストする Red Hat OpenShift Container Platform のバージョンをアップグレードしてください。クラスタ全体のアップグレードを開始する前に、データをバックアップします。

OpenShift Container Platform バージョンのアップグレード時に、Red Hat Advanced Cluster Management Web コンソールに短期間、ページまたはデータを利用できないと表示されることがあります。インジケータには、HTTP 500 (内部サーバーエラー)、HTTP 504 (ゲートウェイタイムアウトエラー)、以前に利用できたデータが利用できないというエラーなどがあります。これも通常のアップグレードの一部で、このようなエラーが発生してもデータが失われることはありません。最終的にページまたはデータは利用できるようになります。

検索インデックスもこのアップグレード中に再ビルドされるため、アップグレード中に送信されるクエリは完全でない可能性があります。

以下の表には、OpenShift Container Platform バージョン 4.4.3 から 4.4.10 へのアップグレードでの主な観察内容についてまとめています。

表1.2 OpenShift Container Platform バージョン 4.3.3 から 4.4.10 へのアップグレードでの観察内容の表

アップグレードプロセスの経過時間 (分:秒)	確認された変化	期間
03:40	ガバナンスコンソールでの HTTP 500 の発生	サービスが 20 秒以内に復元
05:30	AppUI で HTTP 504 ゲートウェイタイムアウトの発生	サービスが 60 秒以内に復元
06:05	Cluster および Search コンソールでの HTTP 504 Gateway Timeout の発生	サービスが 20 秒以内に復元
07:00	Cluster および Search コンソールでの HTTP 504 Gateway Timeout の発生	サービスが 20 秒以内に復元
07:10	Topology および Cluster コンソール内でのエラーメッセージの表示	サービスが 20 秒以内に復元
07:35	多くのコンソールページでの HTTP 500	サービスが 60 秒以内に復元

アップグレードプロセスの経過時間 (分:秒)	確認された変化	期間
08:30	全ページのサービスの復元	

1.9. 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。

1.9.1. 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定します。
- 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、[クラスターの作成](#) 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。

- ベアメタルアセットが存在する場合は、以下が表示される場合があります。

-

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

ベアメタルアセットの削除に関する詳細は、[ベアメタルアセットの削除](#) 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。
 - a. ハブクラスターにログインします。
 - b. 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。
 - a. **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。
 - b. **MultiClusterObservability** カスタムリソースの **Options** メニューを選択します。
 - c. **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。

1.9.2. 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は、Red Hat OpenShift Container Platform ドキュメントの [OpenShift CLI の使用方法](#) 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 バイナリーバージョン 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
helm ls --namespace $ACM_NAMESPACE | cut -f 1 | tail -n +2 | xargs -n 1 helm delete -
--namespace $ACM_NAMESPACE
oc delete apiservice v1beta2.webhook.certmanager.k8s.io v1.admission.cluster.open-
cluster-management.io v1.admission.work.open-cluster-management.io
oc delete clusterimageset --all
oc delete configmap -n $ACM_NAMESPACE cert-manager-controller cert-manager-
cainjector-leader-election cert-manager-cainjector-leader-election-core
oc delete consolelink acm-console-link
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
searchservices.search.acm.com discoveredclusters.discovery.open-cluster-
management.io discoveryconfigs.discovery.open-cluster-management.io
oc delete mutatingwebhookconfiguration cert-manager-webhook cert-manager-webhook-
v1alpha1 ocm-mutating-webhook managedclustermutators.admission.cluster.open-
cluster-management.io
oc delete oauthclient multicloudingress
oc delete rolebinding -n kube-system cert-manager-webhook-webhook-authentication-
reader
oc delete scc kui-proxy-scc
oc delete validatingwebhookconfiguration cert-manager-webhook cert-manager-
webhook-v1alpha1 channels.apps.open.cluster.management.webhook.validator
application-webhook-validator multiclusterhub-operator-validating-webhook ocm-
validating-webhook
```

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。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。

- c. 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。

5. 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。

```
> oc get csv
NAME                                DISPLAY                                VERSION  REPLACES  PHASE
advanced-cluster-management.v2.4.0  Advanced Cluster Management for Kubernetes  2.4.0
Succeeded
```

```
> oc delete clusterserviceversion advanced-cluster-management.v2.4.0
```

```
> oc get sub
```

NAME	PACKAGE	SOURCE	CHANNEL
acm-operator-subscription	advanced-cluster-management	acm-custom-registry	release-2.5

```
> 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。

1.9.3. 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。

ヒント: 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。