Red Hat Advanced Cluster Management for Kubernetes 2.10

ビジネス継続性

ビジネス継続性

Last Updated: 2024-05-24

Red Hat Advanced Cluster Management for Kubernetes 2.10 ビジネス継続性

ビジネス継続性

法律上の通知

概要

クラスターの復元、災害復旧などの詳細は、こちらをお読みください。

目次

第1章 ビジネス継続性

災害復旧ソリューション、ハブクラスター、およびマネージドクラスターについては、次のトピックを参照してください。

- バックアップおよび復元
  - Operator アーキテクチャーのバックアップと復元
  - アクティブパッシブハブクラスターの設定
  - バックアップおよび復元 Operator のインストール
  - バックアップのスケジュールと復元
- VolSync を使用した永続ボリュームの複製
  - VolSync を使用した永続ボリュームの複製
  - 複製されたイメージを使用可能な永続的なボリュームクレームに変換
  - 同期のスケジューリング

1.1. バックアップおよび復元

クラスターのバックアップおよび復元 Operator は、ハブクラスターで実行され、Red Hat Advanced Cluster Management for Kubernetes ハブクラスターの障害に対する災害復旧ソリューションを提供します。ハブクラスターで障害が発生すると、すべてのマネージドクラスターが引き続き正常に動作していても、ポリシー設定ベースのアラートやクラスター更新などの一部の機能が動作しなくなります。ハブクラスターが利用できなくなったら、回復が可能かどうか、新しくデプロイメントされたハブクラスターでデータを回復する必要があるかどうかを判断するための回復計画が必要です。

バックアップおよび復元コンポーネントは、ポリシーを使用してアラートを送信し、メインハブクラスターが使用できなくなり、復元操作が必要な場合に管理者に通知します。このポリシーは、メインハブクラスターがアクティブでクラスターを管理していても、バックアップソリューションが期待どおりに機能しない場合は同じポリシーで管理者にアラートを送信し、バックアップデータの問題を報告します。

クラスターのバックアップと復元の Operator は、OADP Operator に依存して Velero をインストールし、ハブクラスターからデータが保存されているバックアップストレージの場所への接続を作成します。Velero は、バックアップおよび復元操作を実行するコンポーネントです。クラスターのバックアップと復元の Operator ソリューションは、マネージドクラスター、アプリケーション、およびポリシーを含むすべての Red Hat Advanced Cluster Management 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。

クラスターのバックアップおよび復元の Operator は、ハブクラスターのインストールを拡張するサードパーティーリソースのバックアップをサポートします。このバックアップソリューションを使用すると、指定した時間間隔で実行する cron ベースのバックアップスケジュールを定義できます。ハブクラスターで障害が発生した場合、新しいハブクラスターをデプロイし、バックアップされたデータを新しいハブクラスターに移動できます。

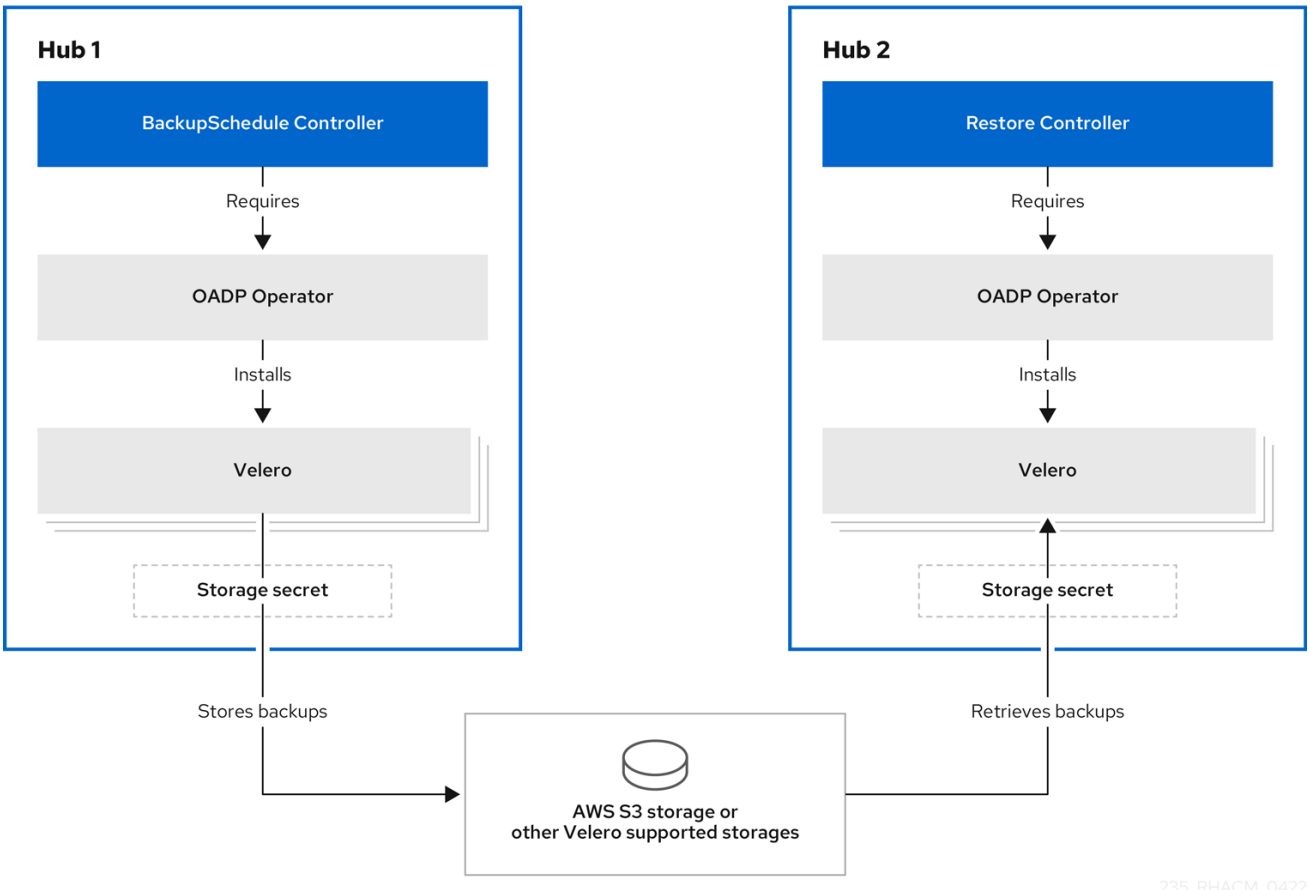
次のトピックを読み続けて、バックアップおよび復元 operator の詳細を確認してください。

- Operator アーキテクチャーのバックアップと復元
- アクティブパッシブハブクラスターの設定

- バックアップおよび復元 Operator のインストール
- バックアップのスケジュールと復元
- バックアップの復元
- バックアップまたは復元設定の検証
- マネージドサービスアカウントを使用してクラスターを自動的に接続する
- 高度な設定のバックアップと復元

1.1.1. Operator アーキテクチャーのバックアップと復元

Operator は、Red Hat Advanced Cluster Management のバックアップスケジュールの設定に使用される **backupSchedule.cluster.open-cluster-management.io** リソースと、バックアップの処理および復元に使用される **restore.cluster.open-cluster-management.io** リソースを定義します。この Operator は、対応する Velero リソースを作成し、リモートクラスターと、復元を必要とする他のハブクラスターリソースのバックアップに必要なオプションを定義します。次の図を表示します。



1.1.1.1. バックアップされるリソース

クラスターのバックアップと復元 Operator のソリューションは、マネージドクラスター、アプリケーション、ポリシーなど、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。このソリューションを使用して、基本的なハブクラスターのインストールを拡張するサードパーティーリソースをバックアップできます。このバックアップソリューションを使用すると、cron ベースのバックアップスケジュールを定義できます。これは、指定された時間間隔で実行し、ハブクラスターのコンテンツの最新バージョンを継続的にバックアップします。

ハブクラスターを交換する必要がある場合、またはハブクラスターに障害が発生したときに災害シナリオにある場合は、新しいハブクラスターをデプロイし、バックアップデータを新しいハブクラスターに移動できます。

バックアップデータを識別するために、次のクラスターバックアップおよび復元プロセスの順序付きリストを表示します。

- **MultiClusterHub** namespace のすべてのリソースを除外します。これは、現在のハブクラスター ID にリンクされているため、バックアップする必要のないインストールリソースのバックアップを回避するためです。

- API バージョンの接尾辞が **.open-cluster-management.io** および **.hive.openshift.io** であるすべてのリソースをバックアップします。これらの接尾辞は、すべての Red Hat Advanced Cluster Management リソースがバックアップされていることを示します。

- **argoproj.io**、**app.k8s.io**、**core.observatorium.io**、**hive.openshift.io** からすべてのリソースをバックアップします。これらのリソースは **acm-resources-schedule** バックアップの中に、バックアップされます。ただし、**agent-install.openshift.io** API グループのリソースは除きます。これらのリソースは、**acm-managed-clusters-schedule** バックアップの中に、バックアップされます。

- API グループ **internal.open-cluster-management.io**、**operator.open-cluster-management.io**、**work.open-cluster-management.io**、**search.open-cluster-management.io**、**admission.hive.openshift.io**、**proxy.open-cluster-management.io**、**action.open-cluster-management.io**、**view.open-cluster-management.io**、**clusterview.open-cluster-management.io**、**velero.io** リソースをすべて除外します。

- 含まれる API グループの一部である全リソース (**clustermanagementaddon**、**observabilityaddon**、**applicationmanager**、**certpolicycontroller**、**iampolicycontroller**、**policycontroller**、**searchcollector**、**workmanager**、**backupschedule**、**restore**、**clusterclaim.cluster.open-cluster-management.io**) を除外しますが、これらは必要ないか、所有者リソースによって再作成され、バックアップされます。

- 次のいずれかのラベルがあるシークレットまたは ConfigMaps をバックアップします: **cluster.open-cluster-management.io/type**、**hive.openshift.io/secret-type**、**cluster.open-cluster-management.io/backup**。

- バックアップが必要で、前述の基準に含まれていないその他のリソースには、**cluster.open-cluster-management.io/backup** ラベルを使用します。以下の例を参照してください。

```
apiVersion: my.group/v1alpha1
kind: MyResource
 metadata:
   labels:
    cluster.open-cluster-management.io/backup: ""
```

  注意: hive.openshift.io.ClusterDeployment リソースによって使用されるシークレットはバックアップする必要があり、クラスターがコンソールを使用して作成された場合にのみ、**cluster.open-cluster-management.io/backup** ラベルで自動的にアノテーションが付けられます。代わりに GitOps を使用して Hive クラスターをデプロイする場合は、**cluster.open-cluster-management.io/backup** ラベルを **ClusterDeployment** で使用されるシークレットに手動で追加する必要があります。

- バックアップしたくない特定のリソースを除外します。バックアッププロセスから Velero リソースを除外するには、次の例を参照してください。

```
apiVersion: my.group/v1alpha1
kind: MyResource
 metadata:
  labels:
    velero.io/exclude-from-backup: "true"
```

1.1.1.2. Red Hat Advanced Cluster Management スケジュールで作成されたバックアップファイル

Red Hat Advanced Cluster Management スケジュールを使用して、リソースタイプまたはラベルアノテーションに基づいて個別のバックアップファイルにグループ化されたハブリソースをバックアップできます。

BackupSchedule.cluster.open-cluster-management.io リソースは、4 つの **Schedule.velero.io** リソースのセットを作成します。これらの **schedule.velero.io** リソースは、リソースとも呼ばれるバックアップファイルを生成します。

スケジュールされたバックアップファイルの一覧を表示するには、**oc get schedules -A | grep acm** のコマンドを実行します。

スケジュールされたバックアップファイルは **backup.velero.io** です。これらのスケジュールされたバックアップファイルの説明を表示するには、次の表を参照してください。

表1.1 スケジュールされたバックアップの表

スケジュールされたバックアップ	説明
Credentials backup	Hive 認証情報、Red Hat Advanced Cluster Management、ユーザーが作成した認証情報、および **ConfigMap** を保存します。このバックアップファイルの名前は、**acm-credentials-schedule-<timestamp>** です。
Resources backup	Red Hat Advanced Cluster Management リソースのバックアップが 1 つ (**acm-resources-schedule-<timestamp> バックアップ)**、汎用リソースのバックアップが 1 つ **(acm-resources-generic-schedule-<timestamp>)** 含まれています。バックアップラベル、**cluster.open-cluster-management.io/backup** のアノテーションが付けられたリソースはすべて、**バックアップ acm-resources-generic-schedule-backup** の下に保存されます。例外は、バックアップ **acm-credentials-schedule-<timestamp>** の下に保存される **Secret** または **ConfigMap** リソースです。
Managed clusters backup	バックアップが復元されるハブクラスターへのマネージドクラスター接続をアクティブにするリソースのみ格納されています。このバックアップファイルの名前は、**acm-managed-clusters-schedule-<timestamp>** です。

1.1.1.3. マネージドクラスターのアクティブ化時に復元されるリソース

cluster.open-cluster-management.io/backup ラベルをリソースに追加すると、リソースは **acm-resources-generic-schedule** バックアップで自動的にバックアップされます。いずれかのリソースを復元する必要がある場合は、ラベル値を **cluster-activation** に設定する必要があります。これは、マネージドクラスターが新しいハブクラスターに移動された後、復元されたリソースで **veleroManagedClustersBackupName:latest** が使用された場合に限ります。これにより、マネージドクラスターのアクティブ化が呼び出されない限り、リソースが復元されなくなります。以下の例を参照してください。

```
apiVersion: my.group/v1alpha1
kind: MyResource
 metadata:
  labels:
    cluster.open-cluster-management.io/backup: cluster-activation
```

注記: 管理対象クラスターの名前空間、またはその中のリソースについては、クラスターのアクティベーション手順でいずれか 1 つを復元する必要があります。したがって、マネージドクラスターの namespace に作成されたバックアップリソースに追加する必要がある場合は、**cluster.open-cluster-management.io/backup** ラベルの **cluster-activation** 値を使用してください。復元プロセスを理解するには、次の情報を参照してください。

- 名前空間を復元すると、**managedcluster-import-controller** によって名前空間が削除されます。
- **managedCluster** カスタムリソースを復元すると、**cluster-manager-registration-controller** によって名前空間が作成されます。

cluster.open-cluster-management.io/backup: cluster-activation ラベルを使用して識別され、**acm-resources-generic-schedule** バックアップによって保存されるアクティベーションデータリソースとは別に、クラスターのバックアップおよび復元 Operator には、デフォルトでは、アクティベーションセット内のいくつかのリソースが含まれます。次のリソースは、**acm-managed-clusters-schedule** バックアップによってバックアップされます。

- **managedcluster.cluster.open-cluster-management.io**
- **managedcluster.clusterview.open-cluster-management.io**
- **klusterletaddonconfig.agent.open-cluster-management.io**
- **managedclusteraddon.addon.open-cluster-management.io**
- **managedclusterset.cluster.open-cluster-management.io**
- **managedclusterset.clusterview.open-cluster-management.io**
- **managedclustersetbinding.cluster.open-cluster-management.io**
- **clusterpool.hive.openshift.io**
- **clusterclaim.hive.openshift.io**
- **clustercurator.cluster.open-cluster-management.io**

1.1.2. アクティブ/パッシブハブクラスターの設定

ここでは、アクティブ/パッシブハブクラスター設定を設定する方法について説明します。この設定では、最初のハブクラスターがデータをバックアップし、アクティブクラスターが使用できなくなったと

きにマネージドクラスターを制御するために 1 つ以上のパッシブハブクラスターがスタンバイになります。

1.1.2.1. アクティブ/パッシブ設定

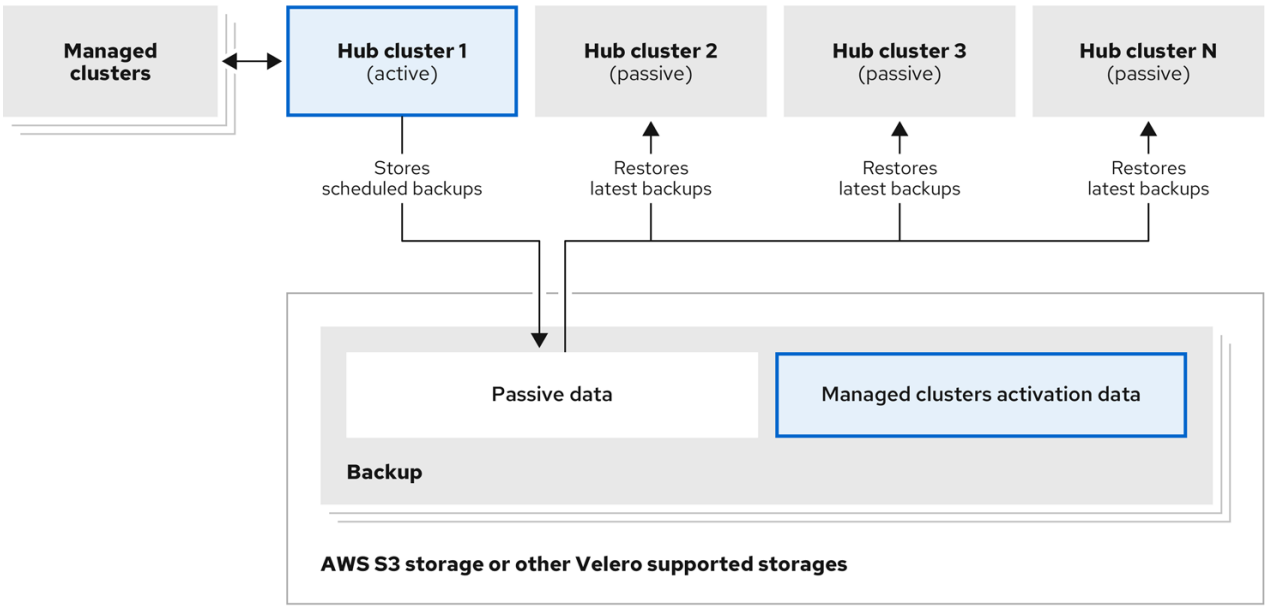
アクティブ/パッシブ設定では、アクティブなハブクラスターが 1 つと、パッシブなハブクラスターが複数あります。アクティブなハブクラスターは、プライマリーハブクラスターとも見なされ、プライマリーハブクラスターは、**BackupSchedule.cluster.open-cluster-management.io** リソースを使用して、クラスターを管理し、定義された時間間隔でリソースをバックアップします。

注: プライマリーハブクラスターのデータをバックアップするには、**アクティブ/パッシブ** 設定は必要ありません。ハブクラスターデータのバックアップと保存のみが可能です。これにより、問題や障害が発生した場合に、新しいハブクラスターをデプロイし、この新しいハブクラスター上にプライマリーハブクラスターデータを復元できます。プライマリーハブクラスターのデータを回復する時間を短縮するには、**アクティブ/パッシブ** 設定を使用します。ただし、これは必須ではありません。

パッシブハブクラスターは、最新のバックアップを継続的に取得し、パッシブデータを復元します。パッシブハブは、**Restore.cluster.open-cluster-management.io** リソースを使用して、新規バックアップデータが利用可能な場合に、プライマリーハブクラスターからパッシブデータを復元します。これらのハブクラスターは、プライマリーハブクラスターで障害が発生した場合にプライマリーハブに切り替えられるように、スタンバイ状態にあります。

アクティブ/パッシブのハブクラスターは同じストレージの場所に接続されており、プライマリーハブクラスターは、プライマリーハブクラスターは、バックアップにアクセスするために、パッシブハブクラスターのデータをバックアップします。この自動復元の設定方法の詳細については、**バックアップを確認しながら、パッシブリソースを復元する** を参照してください。

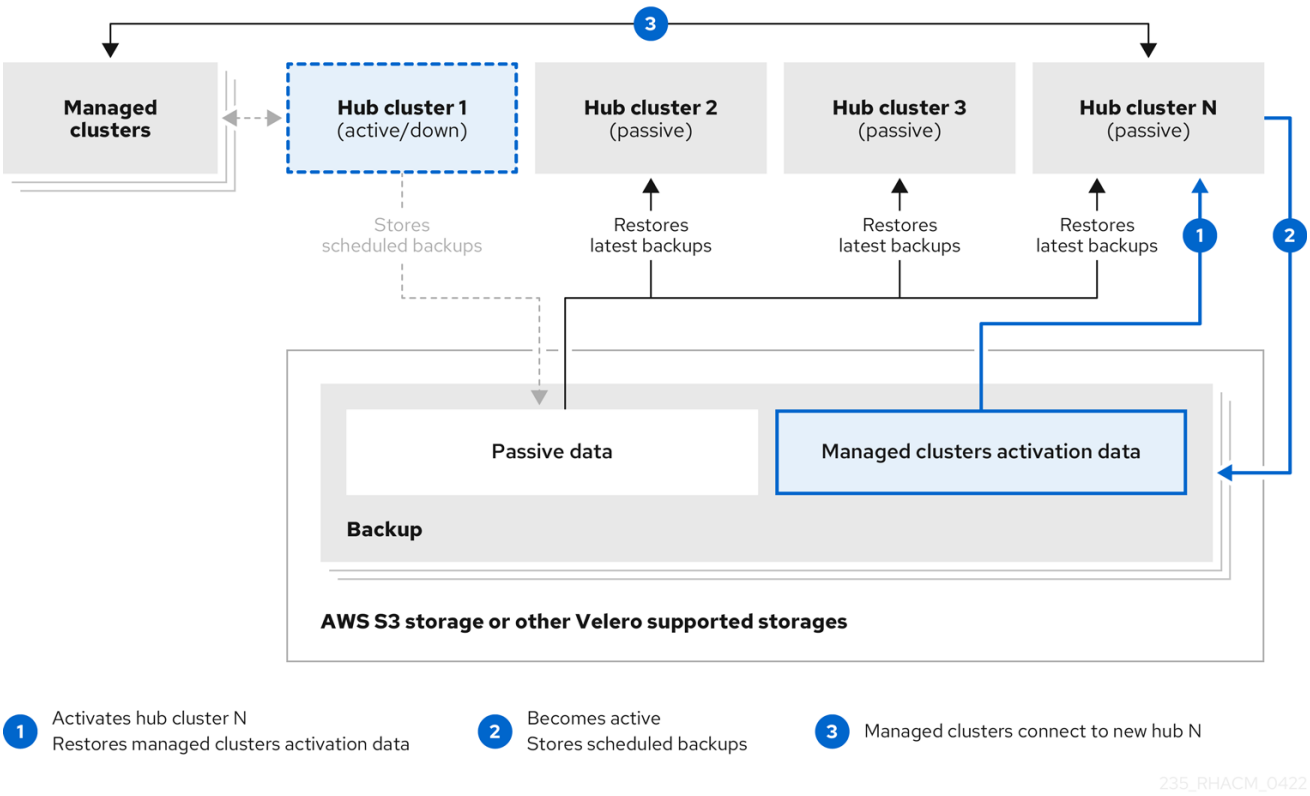
以下の図は、アクティブなハブクラスターがローカルクラスターを管理し、ハブクラスターデータを一定間隔でバックアップします。



パッシブハブクラスターは、マネージドクラスターをパッシブハブクラスターに移動するマネージドクラスターアクティベーションデータを除いて、このデータを復元します。パッシブハブクラスターは、パッシブデータを継続的に復元できます。パッシブハブクラスターは、パッシブデータを 1 回限りの操作として復元できます。詳細は、**パッシブリソースの復元** を参照してください。

1.1.2.2. 障害復旧

プライマリーハブクラスターに障害が発生した場合、管理者はパッシブハブクラスターを選択してマネージドクラスターを引き継ぎます。以下のイメージでは、管理者は **ハブクラスター N** を新しいプライマリーハブクラスターとして使用するように決めます。



ハブクラスター N は、マネージドクラスターのアクティブ化データを復元します。この時点で、マネージドクラスターは、**ハブクラスター N** に接続されます。管理者は、**BackupSchedule.cluster.open-cluster-management.io** リソースを作成し、最初のプライマリーハブクラスターと同じストレージの場所にバックアップを保存することにより、新しいプライマリーハブクラスターである **ハブクラスター N** のバックアップをアクティブ化します。

その他のパッシブハブクラスターはすべて、新しいプライマリーハブクラスターで作成したバックアップデータを使用してパッシブデータを復元するようになりました。**ハブクラスター N** がプライマリーハブクラスターとなり、クラスターの管理とデータのバックアップを行います。

注記:

- 前の図のプロセス 1 は自動化されていません。これは、プライマリーハブクラスターに障害が発生して交換する必要があるかどうか、ハブクラスターとマネージドクラスターの間にネットワーク通信エラーがあるかどうかを管理者が判断する必要があるためです。また、管理者は、どのパッシブハブクラスターがプライマリーハブクラスターになるかを決定します。:aap: ジョブとのポリシー統合は、バックアップポリシーがバックアップエラーを報告したときにジョブを実行することで、この手順を自動化します。

- 前の図のプロセス 2 は手動です。管理者が新しいプライマリーハブクラスターからバックアップを作成しない場合、cron ジョブとしてアクティブに実行されているバックアップを使用して、管理者に通知されます。

1.1.2.3. 関連情報

- Restoring passive resources while checking for backups を参照してください。

- パッシブリソースの復元 を参照してください。

1.1.3. バックアップおよび復元 Operator のインストール

クラスターのバックアップおよび復元 Operator は自動的にインストールされません。Operator のインストールと有効化の方法については、このまま読み進めてください。

注記:

- カスタムリソース定義はクラスタースコープであるため、同じクラスターに 2 つの異なるバージョンの OADP または Velero をインストールすることはできません。2 つの異なるバージョンがある場合、一方のバージョンは間違ったカスタムリソース定義で実行されます。
- **MultiClusterHub** リソースでクラスターのバックアップおよび復元 Operator を有効にしなかった場合でも、OADP Operator と Velero カスタムリソース定義はハブクラスターにインストールされたままになります。**MultiClusterHub** リソースは、OADP および Velero カスタムリソース定義を、クラスターのバックアップおよび復元 Operator を有効にした場合にインストールされる OADP Operator によって使用されるバージョンと調整します。その結果、バックアップおよび復元 Operator を有効にしたときにインストールされる OADP Operator と同じカスタムリソース定義をそのバージョンで使用しない限り、ハブクラスターに別のバージョンの OADP または Velero をインストールできません。
- バックアップコンポーネントは、コンポーネントの namespace にインストールされている OADP Operator と連携して動作します。
- バックアップおよび復元 Operator を使用する前に、ハブクラスターを設定する必要があります。

重要:

OADP Operator を手動でインストールする場合、OADP operator と Velereo のカスタムリソース定義バージョンは一致する必要があります。これらのバージョンが相互に完全一致しない場合は、問題が発生します。以前に、バックアップコンポーネントの namespace とは異なる namespace のハブクラスターに OADP Operator をインストールして使用していた場合は、このバージョンをアンインストールしてください。

Velero と OADP Operator は、Red Hat Advanced Cluster Management for Kubernetes ハブクラスターにインストールされます。これは、Red Hat Advanced Cluster Management ハブクラスターリソースのバックアップおよび復元に使用されます。

Velero でサポートされているストレージプロバイダーのリストについては、OADP のインストールについて を参照してください。

Operator をインストールして有効にするには、次のタスクを実行する必要があります。

- バックアップおよび復元 Operator 用のハブクラスターのセットアップ
- バックアップおよび復元 Operator の有効化

1.1.3.1. バックアップおよび復元 Operator 用のハブクラスターのセットアップ

バックアップおよび復元 Operator を使用するには、ハブクラスターを設定する必要があります。

1.1.3.1.1. ストレージの場所シークレットの作成

ストレージの場所のシークレットを作成するには、以下の手順を実行します。

1. バックアップの保存先となるクラウドストレージの デフォルトシークレットの作成 の手順を完了します。

2. バックアップコンポーネントの namespace にある OADP Operator の namespace にシークレットリソースを作成します。

1.1.3.1.2. バックアップ Operator の有効化

アクティブ/パッシブハブクラスターのバックアップ Operator を有効にするには、以下の手順を実行します。

1. Red Hat OpenShift Container Platform クラスターから、Red Hat Advanced Cluster Management for Kubernetes Operator バージョン 2.10.x をインストールします。**MultiClusterHub** リソースは、Red Hat Advanced Cluster Management のインストール時に自動的に作成され、**Running** のステータスを表示します。

2. クラスターのバックアップおよび復元 Operator を手動でインストールします。クラスターのバックアップおよび復元 Operator (**cluster-backup**) を有効にします。**cluster-backup** パラメーターを **true** に設定して **MultiClusterHub** リソースを編集します。これにより、バックアップコンポーネントと同じネームスペースに OADP オペレータがインストールされます。

3. パッシブハブクラスターで復元操作を実行する前に、次の手順を実行します。

   a. ハブクラスターを手動で設定し、すべての Operator をアクティブハブクラスター上およびアクティブハブクラスターと同じ namespace にインストールします。

   b. Ansible Automation Platform、Red Hat OpenShift Container Platform GitOps、または証明書マネージャーなどの他の Operator がインストールされていることを確認します。検証することで、新しいハブクラスターが最初のハブクラスターと同じ方法で設定されていることを確認します。

   c. パッシブハブクラスターは、バックアップと復元 Operator、および前のハブクラスターで設定された他の Operator をインストールするときに、最初のハブクラスターと同じ namespace 名を使用していることを確認してください。

4. パッシブハブクラスターに **DataProtectionApplication リソース** を作成します。

5. パッシブハブクラスターを、初期ハブクラスターがデータをバックアップしたのと同じストレージの場所に接続します。

1.1.3.1.3. DataProtectionApplication リソースの作成

アクティブ/パッシブハブクラスターの **DataProtectionApplication** リソースのインスタンスを作成するには、以下の手順を実行します。

1. Red Hat OpenShift Container Platform コンソールから、**Operators** > **Installed Operators** を選択します。

2. DataProtectionApplication の下の **Create instance** をクリックします。

3. {ocp-short) コンソールを使用して設定を選択するか、**DataProtectionApplication** の例で説明されているように YAML ファイルを使用して、Velero インスタンスを作成します。

4. **DataProtectionApplication** namespaceを **open-cluster-management-backup** に設定します。

5. **DataProtectionApplication** リソースの仕様 (spec) 値を適切に設定します。次に、**Create** を

5. **DataProtectionApplication** リソースの仕様 (**spec:**) 値を適切に設定します。次に、**Create** をクリックします。
デフォルトのバックアップストレージの場所を使用する場合は、**backupStorageLocations** セクションで値 **default: true** を設定します。以下の **DataProtectionApplication** リソースの例を確認します。

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-sample
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - aws
    restic:
      enable: true
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: my-bucket
          prefix: my-prefix
        config:
          region: us-east-1
          profile: "default"
        credential:
          name: cloud-credentials
          key: cloud
  snapshotLocations:
    - name: default
      velero:
        provider: aws
        config:
          region: us-west-2
          profile: "default"
```

1.1.3.1.4. 非接続環境でのバックアップおよび復元コンポーネントの有効化

非接続環境で Red Hat OpenShift Container Platform を使用してバックアップおよび復元コンポーネントを有効にするには、以下の手順を実行します。

1. 次のアノテーションを使用して **MultiClusterHub** リソースを更新し、OADP Operator のインストール元のソースをオーバーライドします。**MultiClusterHub** リソースで **cluster-backup** コンポーネントを有効にする前に、アノテーションを作成します。

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

2. **redhat-operator-index** はカスタム名であり、非接続環境で Red Hat OpenShift Operator にアクセスするために定義および使用する **CatalogSource** リソースの名前を表します。次のコマンドを実行して、**catalogsource** を取得します。

```
oc get catalogsource -A
```

出力は次の例のような内容になります。

```
NAMESPACE              NAME                        DISPLAY                        TYPE   PUBLISHER
AGE
openshift-marketplace  acm-custom-registry          Advanced Cluster Management    grpc
Red Hat     42h
openshift-marketplace  multiclusterengine-catalog   MultiCluster Engine            grpc   Red Hat
42h
openshift-marketplace  redhat-operator-index                                       grpc             42h
```

1.1.3.2. バックアップおよび復元 Operator の有効化

クラスターのバックアップおよび復元 Operator は、**MultiClusterHub** リソースの初回作成時に有効にできます。**cluster-backup** パラメーターは **true** に設定します。Operator を有効にすると、Operator リソースがインストールされます。

MultiClusterHub リソースがすでに作成されている場合には、**MultiClusterHub** リソースを編集して、クラスターバックアップ Operator をインストールまたはアンインストールできます。クラスターバックアップ Operator をアンインストールする場合は、**cluster-backup** を **false** に設定します。

バックアップおよび復元 Operator が有効にされている場合には、**MultiClusterHub** リソースは以下の YAML ファイルのようになります。

```
apiVersion: operator.open-cluster-management.io/v1
 kind: MultiClusterHub
 metadata:
  name: multiclusterhub
  namespace: open-cluster-management
 spec:
  availabilityConfig: High
  enableClusterBackup: false
  imagePullSecret: multiclusterhub-operator-pull-secret
  ingress:
   sslCiphers:
    - ECDHE-ECDSA-AES256-GCM-SHA384
    - ECDHE-RSA-AES256-GCM-SHA384
    - ECDHE-ECDSA-AES128-GCM-SHA256
    - ECDHE-RSA-AES128-GCM-SHA256
  overrides:
   components:
    - enabled: true
      name: multiclusterhub-repo
    - enabled: true
      name: search
    - enabled: true
      name: management-ingress
    - enabled: true
      name: console
    - enabled: true
```

```
    name: insights
  - enabled: true
    name: grc
  - enabled: true
    name: cluster-lifecycle
  - enabled: true
    name: volsync
  - enabled: true
    name: multicluster-engine
  - enabled: true
    name: cluster-backup
separateCertificateManagement: false
```

1.1.3.3. 関連情報

- Velero を参照してください。
- サポートされている Velero ストレージプロバイダーのリストは、OpenShift Container Platform ドキュメントの AWS S3 互換バックアップストレージプロバイダー を参照してください。
- DataProtectionApplication リソースの詳細を参照してください。

1.1.4. バックアップのスケジュールと復元

バックアップをスケジュールおよび復元するには、以下の手順を実行します。

1. バックアップおよび復元 Operator **backupschedule.cluster.open-cluster-management.io** を使用してバックアップスケジュールを作成し、**restore.cluster.open-cluster-management.io** リソースを使用してバックアップを復元します。
2. 次のコマンドを実行して、**backupschedule.cluster.open-cluster-management.io** リソースを作成します。

```
oc create -f cluster_v1beta1_backupschedule.yaml
```

cluster_v1beta1_backupschedule.yaml リソースは、次のファイルのようになる場合があります。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name: schedule-acm
  namespace: open-cluster-management-backup
spec:
  veleroSchedule: 0 */2 * * * (1)
  veleroTtl: 120h (2)
```

(1) 2 時間ごとにバックアップを作成します。

(2) **オプション:** 120 時間後にスケジュールされたバックアップを削除します。指定されていない場合は、Velero のデフォルト最大値である 720h が使用されます。

backupschedule.cluster.open-cluster-management.io spec プロパティーに関する以下の説明を確認してください。

- **veleroSchedule** は必須のプロパティーで、バックアップをスケジュールする cron ジョブを定義します。
- **veleroTtl** は任意のプロパティーで、スケジュールされているバックアップリソースの有効期限を定義します。指定されていない場合には、Velero で設定された最大デフォルト値 (**720h**) が使用されます。

3. **backupschedule.cluster.open-cluster-management.io** リソースの状態をチェックします。3 つの **schedule.velero.io** リソースの定義が表示されます。以下のコマンドを実行します。

```
oc get BackupSchedule -n open-cluster-management-backup
```

4. 注意: 復元操作は、復元シナリオ向けに別のハブクラスターで実行します。復元操作を開始するには、バックアップを復元するハブクラスターに **restore.cluster.open-cluster-management.io** リソースを作成します。
注記: 新しいハブクラスターにバックアップを復元する場合は、バックアップを作成した以前のバブクラスターがシャットダウンされていることを確認します。実行中の場合には、前のハブクラスターは、マネージドクラスターの調整機能により、マネージドクラスターが使用できなくなったことが検出されるとすぐに、マネージドクラスターの再インポートが試行されます。

   クラスターのバックアップおよび復元 Operator、backup **schedule.cluster.open-cluster-management.io** および **restore.cluster.open-cluster-management.io** リソースを使用して、バックアップまたは復元リソースを作成できます。**cluster-backup-operator** の例を参照してください。

5. 次のコマンドを実行して、**restore.cluster.open-cluster-management.io** リソースを作成します。

```
oc create -f cluster_v1beta1_backupschedule.yaml
```

   リソースは以下のファイルのようになります。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
  namespace: open-cluster-management-backup
spec:
  veleroManagedClustersBackupName: latest
  veleroCredentialsBackupName: latest
  veleroResourcesBackupName: latest
```

6. 以下のコマンドを実行して Velero **Restore** リソースを表示します。

```
oc get restore.velero.io -n open-cluster-management-backup
```

7. 次のコマンドを実行して、Red Hat Advanced Cluster Management **Restore** イベントを表示します。

```
oc describe restore.cluster.open-cluster-management.io -n open-cluster-management-backup
```

パラメーターの説明と **Restore** YAML リソースの例は、**Restoring a backup** セクションで確認してください。

1.1.4.1. Extending backup data

cluster.open-cluster-management.io/backup ラベルをリソースに追加することで、クラスターのバックアップおよび復元を使用してサードパーティーのリソースをバックアップできます。ラベルの値は、空の文字列を含む任意の文字列にすることができます。バックアップするコンポーネントを識別するのに役立つ値を使用してください。たとえば、コンポーネントが IDP ソリューションによって提供される場合は、**cluster.open-cluster-management.io/backup: idp** ラベルを使用します。

注意: マネージドクラスターのアクティブ化リソースが復元されたときにリソースを復元する場合は、**cluster.open-cluster-management.io/backup** ラベルに **cluster-activation** 値を使用します。マネージドクラスターのアクティブ化リソースを復元すると、マネージドクラスターは、復元が開始されたハブクラスターによってアクティブに管理されます。

1.1.4.2. Scheduling a cluster backup

backupschedule.cluster.open-cluster-management.io リソースを作成すると、バックアップスケジュールが有効になります。以下の **backupschedule.cluster.open-cluster-management.io** サンプルを表示します。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name: schedule-acm
  namespace: open-cluster-management-backup
spec:
  veleroSchedule: 0 */2 * * *
  veleroTtl: 120h
```

backupschedule.cluster.open-cluster-management.io リソースを作成したら、以下のコマンドを実行してスケジュールされたクラスターバックアップのステータスを取得します。

```
oc get BackupSchedule -n open-cluster-management-backup
```

backupschedule.cluster.open-cluster-management.io リソースは、バックアップの生成に使用される **schedule.velero.io** リソースを 6 つ作成します。以下のコマンドを実行して、スケジュールされるバックアップのリストを表示します。

```
oc get schedules -A | grep acm
```

次の表が示すとおり、リソースはグループ内で個別にバックアップされます。

表1.2 リソースグループの表

リソース	説明
Credentials backup	Hive 認証情報、Red Hat Advanced Cluster Management、ユーザーが作成した認証情報と ConfigMap を格納するバックアップファイル。

リソース	説明
Resources backup	Red Hat Advanced Cluster Management リソースと汎用リソースのバックアップが 1 つずつ格納されています。これらのリソースには **cluster.open-cluster-management.io/backup** ラベルが使用されます。
Managed clusters backup	バックアップが復元されるハブクラスターへのマネージドクラスター接続をアクティブにするリソースのみ格納されています。

注記: リソースバックアップ ファイルには、マネージドクラスター固有のリソースが含まれていますが、マネージドクラスターをハブクラスターに接続するリソースのサブセットは含まれていません。マネージドクラスターを接続するリソースは、アクティベーションリソースと呼ばれ、マネージドクラスターのバックアップに含まれます。新しいハブクラスターで **認証情報** と **リソース** のバックアップのみのバックアップを復元すると、新しいハブクラスターには、Hive API を使用して作成されたすべてのマネージドクラスターが切り離された状態で表示されます。インポート操作を使用してプライマリーハブクラスターにインポートされたマネージドクラスターは、アクティベーションデータがパッシブハブクラスターに復元された場合にのみ表示されます。マネージドクラスターは、バックアップファイルを作成した元のハブクラスターに引き続き接続されます。

アクティベーションデータが復元されると、Hive API を使用して作成されたマネージドクラスターのみが新しいハブクラスターに自動的に接続されます。その他のマネージドクラスターは、すべて **Pending** 状態になります。それらを新しいクラスターに手動で再割り当てする必要があります。

1.1.4.2.1. バックアップ競合の回避

ハブクラスターがパッシブハブクラスターからプライマリーハブクラスターに、またはその逆に変更され、異なるマネージドクラスターが同じストレージ場所にデータをバックアップする場合、バックアップが競合する可能性があります。

その結果、最新のバックアップは、プライマリーハブクラスターではなくなったハブクラスターによって生成されます。**BackupSchedule.cluster.open-cluster-management.io** リソースがまだ有効であるため、このハブクラスターは引き続きバックアップを生成します。

バックアップの競合を引き起こす可能性のある 2 つのシナリオについては、次のリストを参照してください。

1. プライマリーハブクラスターが予期せず失敗する。これは、次の状況で発生します。
    - プライマリーハブクラスターから Hub1 への通信が失敗する。
    - Hub1 バックアップデータが Hub2 と呼ばれるセカンダリーハブクラスターで復元さされる。
    - 管理者が Hub2 上に **BackupSchedule.cluster.open-cluster-management.io** リソースを作成する。Hub2 はプライマリーハブクラスターとなり、共通のストレージ場所にバックアップデータを生成する。
    - Hub1 が予期せず再び機能し始める。**BackupSchedule.cluster.open-cluster-management.io** リソースは Hub1 でまだ有効に

なっているため、Hub1 は Hub2 と同じストレージ場所へのバックアップの書き込みを再開します。両方のハブクラスターが同じストレージ場所にバックアップデータを書き込んでいます。このストレージの場所から最新のバックアップを復元するハブクラスターは、Hub2 データの代わりに Hub1 データを使用する可能性があります。

2. 管理者は、Hub2 をプライマリーハブクラスターにして、次の条件によって引き起こされる障害シナリオをテストします。

   - Hub1 が停止している。
   - Hub1 バックアップデータが Hub2 で復元される。
   - 管理者が Hub2 上に **BackupSchedule.cluster.open-cluster-management.io** リソースを作成する。Hub2 はプライマリーハブクラスターとなり、共通のストレージ場所にバックアップデータを生成する。
   - 障害テストが完了すると、管理者は以前の状態に戻し、Hub1 を再びプライマリーハブクラスターに指定する。
   - Hub2 がアクティブなまま Hub1 が開始される。**BackupSchedule.cluster.open-cluster-management.io** リソースは Hub2 でまだ有効になっているため、バックアップデータを破損するのと同じストレージ場所にバックアップが書き込まれます。この場所から最新のバックアップを復元するハブクラスターは、Hub1 データの代わりに Hub2 データを使用する可能性があります。このシナリオでは、最初に Hub2 を停止するか、Hub1 を起動する前に Hub2 の **BackupSchedule.cluster.open-cluster-management.io** リソースを削除すると、バックアップの競合の問題が修正されます。

バックアップの競合を回避して報告するために、**BackupSchedule.cluster.open-cluster-management.io** リソースに **BackupCollision** という状態が存在します。コントローラーは、保管場所の最新のバックアップが現在のハブクラスターから生成されたかどうかを定期的に確認します。そうでない場合は、別のハブクラスターが最近バックアップデータを保存場所に書き込んだことになり、ハブクラスターが別のハブクラスターと競合していることを示します。

この場合、現在のハブクラスターの **BackupSchedule.cluster.open-cluster-management.io** リソースのステータスは **BackupCollision** に設定され、データの破損を避けるために、このリソースによって作成された **Schedule.velero.io** リソースが削除されます。**BackupCollision** は、バックアップポリシーによって報告されます。管理者は、無効なハブクラスターから **BackupSchedule.cluster.open-cluster-management.io** リソースを削除し、有効なプライマリーハブクラスターに新たに **BackupSchedule.cluster.open-cluster-management.io** リソースを作成してバックアップを再開する前に、ストレージの場所に書き込むハブクラスターがどれなのかを確認します。

以下のコマンドを実行して、バックアップの競合があるかどうかを確認します。

```
oc get backupschedule -A
```

バックアップの競合がある場合は、出力は次の例のようになります。

```
NAMESPACE       NAME              PHASE             MESSAGE
openshift-adp   schedule-hub-1    BackupCollision   Backup acm-resources-schedule-20220301234625, from cluster with id [be97a9eb-60b8-4511-805c-298e7c0898b3] is using the same storage location. This is a backup collision with current cluster [1f30bfe5-0588-441c-889e-eaf0ae55f941] backup. Review and resolve the collision then create a new BackupSchedule resource to  resume backups from this cluster.
```

1.1.4.3. 関連情報

- cluster-backup-operator の例 を参照してください。
- パラメーターの説明と **Restore** YAML リソースの例は、バックアップの復元 セクションで確認してください。
- バックアップのスケジュールと復元 に戻りってください。

1.1.5. バックアップの復元

一般的な復元シナリオでは、バックアップが実行されるハブクラスターが利用できなくなり、バックアップデータを新しいハブクラスターに移動する必要があります。これには、新しいハブクラスターでクラスター復元操作を実行します。この場合、バックアップが作成されたのとは異なるハブクラスターで復元操作を実行します。

以前のスナップショットからデータを復元できるように、バックアップデータを取得したハブクラスターでデータを復元することもあります。その場合は、復元とバックアップ操作の両方が同じハブクラスターで実行されます。

ハブクラスターに **restore.cluster.open-cluster-management.io** リソースを作成したら、次のコマンドを実行して復元操作のステータスを取得します。

```
oc get restore -n open-cluster-management-backup
```

バックアップファイルに含まれるバックアップリソースが作成されたことも確認できます。

注記: Restore passive resources セクションで説明されているように、**syncRestoreWithNewBackups** オプションを使用して **true** に設定しない限り、**restore.cluster.open-cluster-management.io** リソースは 1 回実行されます。復元操作の完了後に同じ復元操作を再度実行する場合は、同じ **spec** オプションで新しい **restore.cluster.open-cluster-management.io** リソースを作成する必要があります。

復元操作は、バックアップ操作で作成された 3 種類のバックアップをすべて復元するために使用されます。特定の種類のバックアップ (マネージドクラスターのみ、ユーザー認証情報のみ、またはハブクラスターリソースのみ) のみをインストールするように選択できます。

復元では、以下の 3 つの必要な **spec** プロパティーを定義します。ここでは、バックアップしたファイルのタイプに対して復元ロジックが定義されます。

- **veleroManagedClustersBackupName** は、マネージドクラスターのアクティベーションリソースの復元オプションを定義するのに使用されます。
- **veleroCredentialsBackupName** は、ユーザーの認証情報の復元オプションを定義するために使用されます。
- **veleroResourcesBackupName** は、ハブクラスターリソース (**Applications**、**Policy**、その他のハブクラスターリソース (マネージドクラスターパッシブデータなど)) の復元オプションを定義するのに使用されます。
  前述のプロパティーの有効な値は次のとおりです。
  - **latest**: このプロパティーは、このタイプのバックアップで使用可能な、最後のバックアップファイルを復元します。
  - **skip:** このプロパティーは、現在の復元操作でこのタイプのバックアップの復元は試行しません。

- **<backup_name>**: このプロパティーは、名前で参照する指定のバックアップを復元します。

restore.cluster.open-cluster-management.io で作成された **restore.velero.io** リソースの名前は、**<restore.cluster.open-cluster-management.io name>-<velero-backup-resource-name>** のテンプレートルールを使用して生成されます。以下の説明を参照してください。

- **restore.cluster.open-cluster-management.io** は、復元を開始する現在の **restore.cluster.open-cluster-management.io** リソースの名前です。
- **Velero-backup-resource-name** は、データの復元に使用される Velero バックアップファイルの名前です。たとえば、**restore.cluster.open-cluster-management.io** リソース **restore-acm** は **restore.velero.io** 復元リソースを作成します。フォーマットについては、以下の例を参照してください。
  - **restore-acm-acm-managed-clusters-schedule-20210902205438** は、マネージドクラスターのアクティベーションデータのバックアップを復元するのに使用されます。このサンプルでは、リソースの復元に使用される **backup.velero.io** バックアップ名は **acm-managed-clusters-schedule-20210902205438** です。
  - **restore-acm-acm-credentials-schedule-20210902206789** は、認証情報バックアップの復元に使用されます。このサンプルでは、リソースの復元に使用される **backup.velero.io** バックアップ名は **acm-managed-clusters-schedule-20210902206789** です。
  - **restore-acm-acm-resources-schedule-20210902201234** は、アプリケーション、ポリシー、およびマネージドクラスターパッシブデータバックアップなどの他のハブクラスターリソースを復元するのに使用されます。このサンプルでは、リソースの復元に使用される **backup.velero.io** バックアップ名は **acm-managed-clusters-schedule-20210902201234** です。

注記: skip がバックアップタイプに使用されている場合は、**restore.velero.io** は作成されません。

以下の YAML サンプルで、クラスターの **リストア** リソースを参照してください。この例では、利用可能な最新のバックアップファイルを使用して、3 つのタイプのバックアップファイルがすべて復元されています。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
  namespace: open-cluster-management-backup
spec:
  veleroManagedClustersBackupName: latest
  veleroCredentialsBackupName: latest
  veleroResourcesBackupName: latest
```

注記 Hive API によって作成されたマネージドクラスターのみが、マネージドクラスターのバックアップからの **acm-managed-clusters** バックアップが別のハブクラスターに復元されるときに、新しいハブクラスターに自動的に接続されます。他のすべてのマネージドクラスターは **Pending Import** 状態のままであり、新しいハブクラスターにインポートし直す必要があります。詳細は、インポートしたマネージドクラスターの復元 を参照してください。

1.1.5.1. 最初のプライマリーハブへのデータの復元

クラスターでバックアップデータを復元する必要がある場合は、新しいクラスターを作成します。ハブクラスターの復元操作中に、復元されるバックアップデータの一部でない場合に、既存のリソースをク

リーンアップするようにハブクラスターのバックアップの復元を設定できます。復元では、以前のバックアップによって作成されたリソースは消去されますが、ユーザーリソースは消去されません。その結果、このハブクラスター上でユーザーが作成したリソースは消去されないため、このハブクラスター上のデータには、復元されたリソースで使用可能なデータが反映されません。

障害復旧テストは、既存のハブクラスターを使用できる状況の例です。復旧テストでは、ハブのバックアップシナリオのみをテストします。この状況では、最初のプライマリーハブクラスターは新しいリソースを作成しません。代わりに、バックアップデータはプライマリーハブクラスターからパッシブハブクラスターに一時的に立場を変更しました。

1.1.5.2. 新規ハブクラスターの準備

新しいハブクラスターで復元操作を実行する前に、ハブクラスターを手動で設定し、初期ハブクラスターと同じ Operator をインストールする必要があります。Red Hat Advanced Cluster Management Operator は、初期ハブクラスターと同じ namespace にインストールし、**DataProtectionApplication** リソースを作成してから、初期ハブクラスターがデータをバックアップしたのと同じストレージの場所に接続する必要があります。

MultiClusterEngine リソースへの変更を含め、Red Hat Advanced Cluster Management Operator によって作成された **MultiClusterHub** リソースの最初のハブクラスターと同じ設定を使用します。

たとえば、初期ハブクラスターに Ansible Automation Platform、Red Hat OpenShift GitOps、**cert-manager** などの他の Operator がインストールされている場合は、復元操作を実行する前にそれらをインストールする必要があります。これにより、新しいハブクラスターが初期のハブクラスターと同じ方法で設定されます。

1.1.5.3. Cleaning the hub cluster after restore

現在復元されているバックアップで既存のリソースが変更されている場合、Velero は既存のリソースを更新します。Velero はデルタリソースをクリーンアップしません。これは、以前の復元によって作成されたリソースであり、現在復元されているバックアップの一部ではありません。これにより、新しいハブクラスターでハブクラスターデータを復元するときに使用できるシナリオが制限されます。復元が 1 回だけ適用されない限り、新しいハブクラスターをパッシブ設定として確実に使用することはできません。ハブクラスターのデータは、復元されたリソースで利用できるデータを反映していません。

この制限に対処するために、**Restore.cluster.open-cluster-management.io** リソースが作成されると、バックアップ Operator は、ハブクラスターをクリーンアップする復元後の操作を実行します。この操作は、現在復元されているバックアップの一部ではない、以前の Red Hat Advanced Cluster Management 復元によって作成されたすべてのリソースを削除します。

復元後のクリーンアップでは、**cleanupBeforeRestore** プロパティーを使用して、クリーンアップするオブジェクトのサブセットを識別します。復元後のクリーンアップには、次の 2 つのオプションを使用できます。

- **None**: クリーンアップは必要なく、Velero の復元を開始するだけです。真新しいハブクラスターでは **None** を使用します。
- **CleanupRestored**: 現在復元されているバックアップの一部ではない、以前の Red Hat Advanced Cluster Management 復元によって作成されたすべてのリソースをクリーンアップします。
- **CleanupAll**: Red Hat Advanced Cluster Management バックアップの一部である可能性があるハブクラスター上のすべてのリソースを、復元操作の結果として作成されたものではない場合でもクリーンアップします。これは、復元操作が開始される前にハブクラスターで追加のコンテンツが作成される場合に使用されます。
  ベストプラクティス: CleanupAll オプションは使用しないでください。細心の注意を払って最

後の手段としてのみ使用してください。**CleanupAll** は、以前に復元されたバックアップによって作成されたリソースに加えて、ユーザーによって作成されたハブクラスター上のリソースもクリーンアップします。代わりに、CleanupRestored オプションを使用して、ハブクラスターが災害シナリオのパッシブ候補として指定されている場合に、ハブクラスターのコンテンツを更新しないようにします。クリーンハブクラスターをパッシブクラスターとして使用します。

注記:

- Velero は、復元されたバックアップにリソースがない場合に、velero 復元リソースのステータス **PartiallyFailed** を設定します。これは、対応するバックアップが空であるために作成された **restore.velero.io** リソースのいずれかによりリソースが復元されない場合には、**restore.cluster.open-cluster-management.io** リソースが **PartiallyFailed** ステータスになる可能性があることを意味します。

- **syncRestoreWithNewBackups:true** を使用して新規バックアップが利用可能な場合にパッシブデータの復元を継続しない限り、**restore.cluster.open-cluster-management.io** リソースは 1 回実行されます。この場合、同期サンプルで復元パッシブに従います。バックアップの確認時のパッシブリソースの復元 を参照してください。復元操作が完了し、同じハブクラスターで別の復元操作を実行する場合は、新しい **restore.cluster.open-cluster-management.io** リソースを作成する必要があります。

- 複数の **restore.cluster.open-cluster-management.io** リソースを作成できますが、いつでもアクティブにできるのは 1 つだけです。

1.1.5.4. バックアップの確認中のパッシブリソースの復元

新しいバックアップが利用可能かどうかを引き続き確認し、それらを自動的に復元しながら、**restore-passive-sync** サンプルを使用してパッシブデータを復元します。新しいバックアップを自動的に復元するには、**syncRestoreWithNewBackups** パラメーターを **true** に設定する必要があります。また、最新のパッシブデータだけを復元する必要もあります。サンプルの例は、このセクションの最後にあります。

VeleroResourcesBackupName および **VeleroCredentialsBackupName** パラメーターを **latest** に設定し、**VeleroManagedClustersBackupName** パラメーターを省略して **スキップ** します。**VeleroManagedClustersBackupName** が **latest** に設定された直後に、マネージドクラスターは新しいハブクラスターでアクティベートされ、プライマリーハブクラスターになります。

アクティベートされたマネージドクラスターがプライマリーハブクラスターになると、復元リソースが **Finished** に設定され、**true** に設定されていても **syncRestoreWithNewBackups** は無視されます。

デフォルトでは、コントローラーは **syncRestoreWithNewBackups** が **true** に設定されると、30 分ごとに新規バックアップをチェックします。新しいバックアップが見つかった場合は、バックアップされたリソースを復元します。**restoreSyncInterval** パラメーターを更新してチェックの期間を変更できます。

たとえば、10 分ごとにバックアップをチェックする次のリソースを参照してください。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm-passive-sync
  namespace: open-cluster-management-backup
spec:
  syncRestoreWithNewBackups: true # restore again when new backups are available
  restoreSyncInterval: 10m # check for new backups every 10 minutes
  cleanupBeforeRestore: CleanupRestored
```

```
  veleroManagedClustersBackupName: skip
  veleroCredentialsBackupName: latest
  veleroResourcesBackupName: latest
```

1.1.5.5. Restoring passive resources

パッシブ設定でハブクラスターリソースを復元するには、**restore-acm-passive** サンプルを使用します。パッシブデータは、シークレット、ConfigMap、アプリケーション、ポリシー、およびすべてのマネージドクラスターカスタムリソースなどのバックアップデータで、マネージドクラスターとハブクラスター間の接続をアクティブ化しません。バックアップリソースは、認証情報のバックアップおよび復元リソースによりハブクラスターで復元されます。

以下のサンプルを参照してください。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm-passive
  namespace: open-cluster-management-backup
spec:
  cleanupBeforeRestore: CleanupRestored
  veleroManagedClustersBackupName: skip
  veleroCredentialsBackupName: latest
  veleroResourcesBackupName: latest
```

1.1.5.6. アクティベーションリソースの復元

パッシブハブクラスターにアクティベーションデータを復元する前に、バックアップが作成された以前のハブクラスターをシャットダウンします。プライマリーハブクラスターがまだ実行中の場合は、このハブクラスターで実行されている調整手順に基づいて、使用できなくなったマネージドクラスターへの再接続を試行します。

ハブクラスターでクラスターを管理する場合は、**restore-acm-passive-activate** サンプルを使用します。この場合、パッシブリソースを使用するハブクラスターで他のデータがすでに復元されていることを前提とします。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm-passive-activate
  namespace: open-cluster-management-backup
spec:
  cleanupBeforeRestore: CleanupRestored
  veleroManagedClustersBackupName: latest
  veleroCredentialsBackupName: skip
  veleroResourcesBackupName: skip
```

パッシブリソースを復元した方法に応じて、アクティベーションリソースを復元するいくつかのオプションがあります。

- Restore passive resources while checking for backups to restore passive dataセクションに記載されているように、**restore-acm-passive-sync cluster.open-cluster-management.io** リソースを使用した場合は、このリソースで **veleroManagedClustersBackupName** の値を

latest の値に更新します。その結果、マネージドクラスターリソースと **restore-acm-passive-sync** リソースが復元されます。

- パッシブリソースを 1 回限りの操作で復元した場合、またはリソースをまだ復元していない場合は、**Restoring all resources** セクションで指定されているように、すべてのリソースを復元することを選択します。

1.1.5.7. マネージドクラスターのアクティベーションデータの復元

cluster.open-cluster-management.io/backup: cluster-activation ラベルを使用すると、マネージドクラスターのアクティベーションデータまたはその他のアクティベーションデータリソースは、マネージドクラスターのバックアップおよび resource-generic バックアップにより保存されます。アクティベーションデータが新しいハブクラスターに復元すると、マネージドクラスターは、復元が実行するハブクラスターによりアクティブに管理されます。Operator の使用方法については、**バックアップのスケジュールと復元** を参照してください。

1.1.5.8. 全リソースの復元

一度にすべてのデータを復元し、ハブクラスターがマネージドクラスターを 1 つのステップで管理するようにする場合は、**restore-acm** サンプルを使用します。ハブクラスターに **restore.cluster.open-cluster-management.io** リソースを作成したら、次のコマンドを実行して復元操作のステータスを取得します。

```
oc get restore -n open-cluster-management-backup
```

サンプルは、次のリソースに酷似している可能性があります。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
  namespace: open-cluster-management-backup
spec:
  cleanupBeforeRestore: CleanupRestored
  veleroManagedClustersBackupName: latest
  veleroCredentialsBackupName: latest
  veleroResourcesBackupName: latest
```

ハブクラスターから、バックアップファイルに含まれるバックアップされたリソースが作成されていることを確認します。

1.1.5.9. インポートされたマネージドクラスターの復元

Hive API を使用してプライマリーハブクラスターに接続されたマネージドクラスターのみが、アクティベーションデータが復元される新しいハブクラスターに自動的に接続されます。これらのクラスターは、**Clusters** タブの **Create cluster** ボタンを使用して、プライマリーハブクラスター上に作成されています。**Import cluster** ボタンを使用して最初のハブクラスターに接続されたマネージドクラスターは、アクティベーションデータが復元されると **Pending Import** として表示され、新しいハブクラスターにインポートし直す必要があります。

Hive がマネージドクラスター **kubeconfig** をハブクラスターのマネージドクラスター namespace に格納するため、Hive マネージドクラスターを新しいハブクラスターに接続できます。これは、新しいハブクラスターでバックアップおよび復元されます。次に、インポートコントローラーは、復元された設定

を使用してマネージドクラスターのブートストラップ **kubeconfig** を更新します。これは、Hive API を使用して作成されたマネージドクラスターでのみ使用できます。インポートされたクラスターでは使用できません。

インポートされたクラスターを新しいハブクラスターに再接続するには、復元操作の開始後に **auto-import-secret** リソースを手動で作成します。詳細は、**自動インポートシークレットを使用したクラスターのインポート** を参照してください。

Pending Import 状態のクラスターごとに、マネージドクラスターの namespace に **auto-import-secret** リソースを作成します。インポートコンポーネントが新しいハブクラスターで自動インポートを開始するのに十分な権限を持つ **kubeconfig** またはトークンを使用します。マネージドクラスターに接続するには、トークンを使用して各マネージドクラスターにアクセスできる必要があります。トークンには、**klusterlet** ロールバインディングまたは同じアクセス権限を持つロールが必要です。

1.1.5.10. 他の復元サンプルの使用

次の復元セクションを参照して、さまざまな種類のバックアップファイルを復元するための YAML の例を確認してください。

- 3 種類のバックアップリソースをすべて復元します。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
  namespace: open-cluster-management-backup
spec:
  veleroManagedClustersBackupSchedule: latest
  veleroCredentialsBackupSchedule: latest
  veleroResourcesBackupSchedule: latest
```

- マネージドクラスターリソースのみを復元します。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
  namespace: open-cluster-management-backup
spec:
  veleroManagedClustersBackupName: latest
  veleroCredentialsBackupName: skip
  veleroResourcesBackupName: skip
```

- **acm-managed-clusters-schedule-20210902205438** バックアップを使用して、マネージドクラスターのリソースのみを復元します。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm
  namespace: open-cluster-management-backup
spec:
  veleroManagedClustersBackupName: acm-managed-clusters-schedule-20210902205438
  veleroCredentialsBackupName: skip
  veleroResourcesBackupName: skip
```

注記:

- **restore.cluster.open-cluster-management.io** リソースは 1 回実行されます。復元操作が完了したら、オプションで同じハブクラスターで別の復元操作を実行できます。新しい復元操作を実行するには、**restore.cluster.open-cluster-management.io** リソースを新規作成する必要があります。
- 複数の **restore.cluster.open-cluster-management.io** を作成できますが、同時に実行できるのは 1 つのみです。

1.1.5.11. 復元イベントの表示

以下のコマンドを使用して、復元イベントに関する情報を取得します。

```
oc describe -n open-cluster-management-backup <restore-name>
```

イベント一覧は以下の例のようになります。

```
Spec:
  Cleanup Before Restore:             CleanupRestored
  Restore Sync Interval:              4m
  Sync Restore With New Backups:      true
  Velero Credentials Backup Name:     latest
  Velero Managed Clusters Backup Name:  skip
  Velero Resources Backup Name:       latest
Status:
  Last Message:                  Velero restores have run to completion, restore will continue to sync
with new backups
  Phase:                         Enabled
  Velero Credentials Restore Name:  example-acm-credentials-schedule-20220406171919
  Velero Resources Restore Name:    example-acm-resources-schedule-20220406171920
Events:
  Type    Reason                 Age   From              Message
  ----    ------                 ----  ----              -------
  Normal  Prepare to restore:    76m   Restore controller  Cleaning up resources for backup acm-
credentials-hive-schedule-20220406155817
  Normal  Prepare to restore:    76m   Restore controller  Cleaning up resources for backup acm-
credentials-cluster-schedule-20220406155817
  Normal  Prepare to restore:    76m   Restore controller  Cleaning up resources for backup acm-
credentials-schedule-20220406155817
  Normal  Prepare to restore:    76m   Restore controller  Cleaning up resources for backup acm-
resources-generic-schedule-20220406155817
  Normal  Prepare to restore:    76m   Restore controller  Cleaning up resources for backup acm-
resources-schedule-20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-credentials-schedule-
20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-resources-generic-
schedule-20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-resources-schedule-
20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-credentials-cluster-
schedule-20220406155817
  Normal  Velero restore created:  74m   Restore controller  example-acm-credentials-hive-schedule-
20220406155817
  Normal  Prepare to restore:    64m   Restore controller  Cleaning up resources for backup acm-
```

```
resources-schedule-20220406165328
  Normal  Prepare to restore:     62m   Restore controller  Cleaning up resources for backup acm-
credentials-hive-schedule-20220406165328
  Normal  Prepare to restore:     62m   Restore controller  Cleaning up resources for backup acm-
credentials-cluster-schedule-20220406165328
  Normal  Prepare to restore:     62m   Restore controller  Cleaning up resources for backup acm-
credentials-schedule-20220406165328
  Normal  Prepare to restore:     62m   Restore controller  Cleaning up resources for backup acm-
resources-generic-schedule-20220406165328
  Normal  Velero restore created: 61m   Restore controller  example-acm-credentials-cluster-
schedule-20220406165328
  Normal  Velero restore created: 61m   Restore controller  example-acm-credentials-schedule-
20220406165328
  Normal  Velero restore created: 61m   Restore controller  example-acm-resources-generic-
schedule-20220406165328
  Normal  Velero restore created: 61m   Restore controller  example-acm-resources-schedule-
20220406165328
  Normal  Velero restore created: 61m   Restore controller  example-acm-credentials-hive-schedule-
20220406165328
  Normal  Prepare to restore:     38m   Restore controller  Cleaning up resources for backup acm-
resources-generic-schedule-20220406171920
  Normal  Prepare to restore:     38m   Restore controller  Cleaning up resources for backup acm-
resources-schedule-20220406171920
  Normal  Prepare to restore:     36m   Restore controller  Cleaning up resources for backup acm-
credentials-hive-schedule-20220406171919
  Normal  Prepare to restore:     36m   Restore controller  Cleaning up resources for backup acm-
credentials-cluster-schedule-20220406171919
  Normal  Prepare to restore:     36m   Restore controller  Cleaning up resources for backup acm-
credentials-schedule-20220406171919
  Normal  Velero restore created: 36m   Restore controller  example-acm-credentials-cluster-
schedule-20220406171919
  Normal  Velero restore created: 36m   Restore controller  example-acm-credentials-schedule-
20220406171919
  Normal  Velero restore created: 36m   Restore controller  example-acm-resources-generic-
schedule-20220406171920
  Normal  Velero restore created: 36m   Restore controller  example-acm-resources-schedule-
20220406171920
  Normal  Velero restore created: 36m   Restore controller  example-acm-credentials-hive-schedule-
20220406171919
```

1.1.5.12. 関連情報

- DataProtectionApplication を参照してください。
- 自動インポートシークレットを使用したクラスターのインポート を参照してください。
- バックアップのスケジューリングと復元 を参照してください。

1.1.6. マネージドサービスアカウントを使用してクラスターを自動的に接続する

バックアップコントローラーは、マネージドサービスアカウントコンポーネントを使用して、インポートされたクラスターを新しいハブクラスターに自動的に接続します。マネージドサービスアカウントは、マネージドクラスターの namespace ごとに、それぞれのインポートされたクラスターにバックアップされるトークンを作成します。トークンは **klusterlet-bootstrap-kubeconfig ClusterRole** バインディングを使用します。これにより、自動インポート操作でトークンを使用できるようになりま

す。**klusterlet-bootstrap-kubeconfig ClusterRole** は、**bootstrap-hub-kubeconfig** シークレットのみを取得または更新できます。Managed Service Account コンポーネントの詳細は、**Managed Service Account とは** を参照してください。

アクティベーションデータが新しいハブクラスターに復元されると、復元コントローラーは復元後の操作を実行し、**Pending Import** 状態のすべてのマネージドクラスターを探します。マネージドサービスアカウントによって生成された有効なトークンが見つかった場合、コントローラーはそのトークンを使用して **auto-import-secret** を作成します。その結果、インポートコンポーネントはマネージドクラスターの再接続を試みます。クラスターにアクセスできる場合、操作は成功です。

1.1.6.1. Enabling automatic import

マネージドサービスアカウントコンポーネントを使用した自動インポート機能は、デフォルトでは無効になっています。自動インポート機能を有効にするには、次の手順を実行します。

1. **MultiClusterEngine** リソースで **managedserviceaccount enabled** パラメーターを **true** に設定して、マネージドサービスアカウントコンポーネントを有効にします。以下の例を参照してください。

```
apiVersion: multicluster.openshift.io/v1
kind: MultiClusterEngine
metadata:
  name: multiclusterhub
spec:
  overrides:
    components:
      - enabled: true
        name: managedserviceaccount
```

2. **useManagedServiceAccount** パラメーターを **true** に設定して、**BackupSchedule.cluster.open-cluster-management.io** リソースの自動インポート機能を有効にします。以下の例を参照してください。

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

デフォルトのトークン有効期間は、ライフサイクル全体にわたってトークンを保存するすべてのバックアップに対してトークンが有効になる可能性を高くするために、**veleroTtl** の値の 2 倍に設定されます。場合によっては、オプションの **managedServiceAccountTTL** プロパティーの値を設定することで、トークンの有効期間を制御する必要がある場合があります。

生成されたトークンのデフォルトのトークン有効期限を更新する必要がある場合は、注意して **manageServiceAccountTTL** を使用してください。トークンの有効期限をデフォルト値から変更すると、バックアップのライフサイクル中に有効期限が切れるように設定されたトークンを使用してバックアップが作成される可能性があります。その結果、インポート機能はマネージドクラスターでは機能しません。

重要: トークンの有効期間を制御する必要がない限り、**managedServiceAccountTTL** を使用しないでください。

protectedServiceAccountTTL プロパティーの使用例は、次の例を参照してください。

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name: schedule-acm-msa
  namespace: open-cluster-management-backup
spec:
  veleroSchedule:
  veleroTtl: 120h
  useManagedServiceAccount: true
  managedServiceAccountTTL: 300h
```

自動インポート機能を有効にすると、バックアップコンポーネントは以下を作成して、インポートされたマネージドクラスターの処理を開始します。

- **managed-serviceaccount** という名前の **ManagedServiceAddon**。
- **auto-import-account** という名前の **ManagedServiceAccount**。
- マネージドクラスターで **ManagedServiceAccount** トークンの **klusterlet-bootstrap-kubeconfig RoleBinding** を設定するための、各 **ManagedServiceAccount** の **ManifestWork**。

トークンは、マネージドサービスアカウントの作成時にマネージドクラスターにアクセスできる場合にのみ作成されます。それ以外の場合は、後でマネージドクラスターが利用可能になったときに作成されます。

1.1.6.2. 自動インポートに関する考慮事項

次のシナリオでは、新しいハブクラスターに移動するときに、マネージドクラスターが自動的にインポートされなくなる可能性があります。

- **ManagedServiceAccount** トークンを使用せずにハブバックアップを実行する場合 (たとえば、マネージドクラスターにアクセスできないときに **ManagedServiceAccount** リソースを作成する場合)、マネージドクラスターを自動インポートするためのトークンがバックアップに含まれません。
- **auto-import-account** シークレットトークンが有効でバックアップされている場合、自動インポート操作は失敗しますが、バックアップで使用可能なトークンの有効期限がすでに切れている場合、復元操作が実行されます。**restore.cluster.open-cluster-management.io** リソースは、各マネージドクラスターの無効なトークンの問題を報告します。
- 復元時に作成される **auto-import-secret** は **ManagedServiceAccount** トークンを使用してマネージドクラスターに接続するため、マネージドクラスターは **kube apiserver** 情報も提供する必要があります。**ManagedCluster** リソースに **apiserver** を設定する必要があります。以下の例を参照してください。

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

```
hubAcceptsClient: true
leaseDurationSeconds: 60
managedClusterClientConfigs:
  url: <apiserver>
```

クラスターがハブクラスターにインポートされると、**apiserver** は OpenShift Container Platform クラスターでのみ自動的にセットアップされます。EKS クラスターなどの他のタイプのマネージドクラスターでは、**apiserver** を手動で設定する必要があります。そうしないと、自動インポート機能でクラスターが無視されます。その結果、クラスターを復元ハブクラスターに移動すると、クラスターは **Pending Import** 状態のままになります。

- **ManagedServiceAccount** シークレットにバックアップラベルが設定される前にバックアップスケジュールが実行された場合、**ManagedServiceAccount** シークレットがバックアップに含まれない可能性があります。**ManagedServiceAccount** シークレットには、作成時に設定されたクラスター **open-cluster-management.io/backup** ラベルがありません。このため、バックアップコントローラーはマネージドクラスターの namespace で **ManagedServiceAccount** シークレットを定期的に検索し、見つからない場合はバックアップラベルを追加します。

1.1.6.3. 自動インポートの無効化

BackupSchedule リソースで **useManagedServiceAccount** パラメーターを **false** に設定することで、クラスターの自動インポート機能を無効にすることができます。以下の例を参照してください。

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

デフォルト値は **false** です。値を **false** に設定した後、バックアップ Operator は、**ManagedServiceAddon**、**ManagedServiceAccount**、および **ManifestWork** を含む、作成されたすべてのリソースを削除します。リソースを削除すると、ハブクラスターとマネージドクラスターの自動インポートトークンが削除されます。

1.1.6.4. 関連情報

- Managed Service Account の詳細は、Managed Service Account とは を参照してください。.
- Managed Service Account を使用したクラスターの自動接続 に戻ってください。

1.1.7. バックアップまたは復元設定の検証

MultiClusterHub リソースで **cluster-backup** オプションを **true** に設定すると、マルチクラスターエンジン Operator により、クラスターのバックアップおよび復元 Operator の Helm チャートがインストールされます。この Helm チャートは **cluster-backup-chart** という名前です。このチャートにより、**backup-restore-enabled** ポリシーと **backup-restore-auto-import** ポリシーがインストールされます。これらのポリシーを使用して、バックアップおよび復元コンポーネントの問題に関する情報を表示します。

注: ハブクラスターは自動的にインポートされ、**ローカルクラスター** 管理クラスターを使用してそれ自体で管理されます。**MultiClusterHub** リソースで **disableHubSelfManagement=true** を設定して無効

にすると、**backup-restore-enabled** ポリシーはハブクラスターに配置されず、ポリシーテンプレートでレポートは生成されません。

クラスターハブがグローバルハブクラスターによって管理されている場合、またはマネージドクラスターインスタンスにインストールされている場合は、**disableHubSelfManagement=true** の設定が無効になります。この場合、**backup-restore-enabled** ポリシーを有効にできます。マネージドハブクラスターを表す **ManagedCluster** リソースに **is-hub=true** ラベルを設定して、ポリシーを有効にします。

backup-restore-enabled ポリシーには、以下の制約を確認するテンプレートセットが含まれます。

- **OADP チャネルの検証**
    - **MultiClusterHub** でバックアップコンポーネントを有効にすると、クラスターのバックアップおよび復元 Operator の Helm チャートによって OADP Operator がインストールされます。**OADP-channel** テンプレートは、インストールされている Red Hat OADP Operator のバージョンが、Red Hat Advanced Cluster Management クラスターのバックアップおよび復元 Operator によって設定されたバージョンと一致するかどうかを確認します。
    - このテンプレートは、ハブクラスター上にインストールされている Red Hat OADP Operator を検出し、その Red Hat OADP Operator が、Red Hat Advanced Cluster Management クラスターのバックアップおよび復元 Operator の Helm チャートによってインストールされたバージョンと一致しない場合に、違反を表示します。この違反により、クラスター上で間違ったバージョンの OADP Operator が検出され、表示されます。OADP Operator と Velero カスタムリソース定義 (CRD) は **cluster-scoped** であるため、それらの複数のバージョンを同じクラスターにインストールすることはできません。代わりに、正しいバージョンだけをインストールする必要があります。
    - 次の例に該当する場合、バックアップおよび復元 Operator が間違った CRD を使用して実行され、誤った動作が発生する可能性があります。
        - Red Hat Advanced Cluster Management に多くのバージョンの OADP がインストールされている。
        - **MultiClusterHub** によってインストールされた OADP バージョンがアンインストールされていて、別のバージョンを手動でインストールした場合。
- **Pod の検証**
  以下のテンプレートは、Pod のステータスでバックアップコンポーネントおよび依存関係の有無を確認します。
    - **acm-backup-pod-running** テンプレートは、バックアップおよび復元 Operator Pod が実行されているかどうかを確認します。
    - **OADP-pod-running** テンプレートは、OADP Operator Pod が実行されているかどうかを確認します。
    - **velero-pod-running** テンプレートは Velero Pod が実行されているかどうかを確認します。
- **Data Protection Application の検証**
    - **data-protection-application-available** テンプレートは、**DataProtectioApplicatio.oadp.openshift.io** リソースが作成されるかどうかを確認します。この OADP リソースは Velero 設定をセットアップします。
- **バックアップストレージの検証**

- **backup-storage-location-available** テンプレートは、**BackupStorageLocation.velero.io** リソースが作成され、ステータス値が **Available** かどうかを確認します。これは、バックアップストレージへの接続が有効であることを意味します。

- BackupSchedule **競合検証**

  - **acm-backup-clusters-collision-report** テンプレートは、**BackupSchedule.cluster.open-cluster-management.io** が現在のハブクラスターに存在する場合に、ステータスが **BackupCollision** ではないことを検証します。これにより、バックアップデータをストレージの場所に書き込むときに、現在のハブクラスターが他のハブクラスターと競合していないことを確認できます。
    BackupCollision の定義については、バックアップ競合の回避 を参照してください。

- BackupSchedule **および復元ステータスの検証**

  - **acm-backup-phase-validation** テンプレートは、**BackupSchedule.cluster.open-cluster-management.io** が現在のクラスターに存在する場合に、ステータスが **Failed** でないこと、または **空** の状態であることを確認します。これにより、このクラスターがプライマリーハブクラスターであり、バックアップを生成している場合に **BackupSchedule.cluster.open-cluster-management.io** ステータスが正常であることが保証されます。

  - 同じテンプレートは、**Restore.cluster.open-cluster-management.io** が現在のクラスターに存在する場合に、ステータスが **失敗** でないこと、または **空** の状態にないことを確認します。これにより、このクラスターがセカンダリーハブクラスターであり、バックアップを復元する場合に、**Restore.cluster.open-cluster-management.io** のステータスが正常であることが保証されます。

- **バックアップの存在検証**

  - **acm-managed-clusters-schedule-backups-available** テンプレートは、**BackupStorageLocation.velero.io** で指定された場所で **Backup.velero.io** リソースが利用可能かどうかを確認し、バックアップが **BackupSchedule.cluster.open-cluster-management.io** リソースによって作成されるかどうかを確認します。これにより、バックアップが少なくとも 1 回実行され、バックアップと復元 Operator が検証されます。

- **完了するためのバックアップ**

  - **acm-backup-in-progress-report** テンプレートは、**Backup.velero.io** リソースが **InProgress** 状態で停止していないか確認します。この検証が追加されるのは、多数のリソースがある場合、バックアップの実行中に velero Pod が再起動し、バックアップが完了せずに進行中のままになるためです。通常のバックアップ中、バックアップリソースは、実行中のどこかの時点で進行中になりますが、停止しているわけではなく、完了まで実行されます。スケジュールの実行中およびバックアップの進行中に **acm-backup-in-progress-report** テンプレートが警告を報告するのは正常です。

- cron **ジョブとしてアクティブに実行されるバックアップ**

  - **BackupSchedule.cluster.open-cluster-management.io** はアクティブに実行され、ストレージの場所に新しいバックアップを保存します。この検証は、**backup-schedule-cron-enabled** ポリシーテンプレートにより行われます。テンプレートは、ストレージの場所に **velero.io/schedule-name: acm-validation-policy-schedule** ラベルの付いた **Backup.velero.io** があることを確認します。

  - **acm-validation-policy-schedule** バックアップは、バックアップ cron スケジュールの時刻が設定された後に期限切れになるように設定されています。バックアップを作成するために実行されている cron ジョブがない場合には、古い **acm-validation-policy-schedule**

バックアップは期限切れになり、新しいバックアップが作成されないので削除されます。したがって、現在 **acm-validation-policy-schedule backups** が存在しない場合には、アクティブな cron ジョブがバックアップを生成することはありません。

- このポリシーは、ハブクラスターがアクティブで、バックアップを作成または復元するときに、バックアップの問題をハブクラスター管理者に通知することを目的としています。

backup-restore-auto-import ポリシーには、次の制約を確認するテンプレートのセットが含まれています。

- **自動インポートのシークレット検証**
  - **auto-import-account-secret** テンプレートは、**ManagedServiceAccount** シークレットが **local-cluster** 以外のマネージドクラスターの namespace に作成されているかどうかを確認します。バックアップコントローラーにより、インポートされたマネージドクラスターが定期的にスキャンされます。マネージドクラスターが検出されるとすぐに、バックアップコントローラーはマネージドクラスターの namespace に **ManagedServiceAccount** リソースを作成します。このプロセスにより、マネージドクラスター上でトークンの作成が開始されます。ただし、この操作の時点でマネージドクラスターにアクセスできない場合、**ManagedServiceAccount** はトークンを作成できません。たとえば、マネージドクラスターが休止状態の場合、トークンを作成できません。したがって、この期間中にハブのバックアップが実行されると、マネージドクラスターを自動インポートするためのトークンがバックアップに不足します。
- **自動インポートのバックアップラベル検証**
  - **auto-import-backup-label** テンプレートは、**local-cluster** 以外のマネージドクラスターの namespace に **ManagedServiceAccount** シークレットが存在することを検証します。このテンプレートが **ManagedServiceAccount** シークレットを検出した場合、テンプレートはシークレットに **cluster.open-cluster-management.io/backup** ラベルを適用します。このラベルは、Red Hat Advanced Cluster Management のバックアップに **ManagedServiceAccount** シークレットを含める場合に重要です。

1.1.7.1. サーバー側の暗号化を使用したデータの保護

サーバー側の暗号化は、保存場所でデータを受信するアプリケーションまたはサービスのデータ暗号化です。バックアップメカニズム自体は、転送中 (バックアップストレージの場所との間を移動するとき) または保存中 (バックアップストレージの場所のディスクに保存されている間) にデータを暗号化しません。代わりに、オブジェクトおよびスナップショットシステムのネイティブメカニズムに依存しています。

ベストプラクティス: 使用可能なバックアップストレージのサーバー側の暗号化を使用して、宛先でデータを暗号化します。バックアップには、認証情報や設定ファイルなど、ハブクラスターの外部に保存するときに暗号化する必要があるリソースが含まれています。

serverSideEncryption パラメーターおよび **kmsKeyId** パラメーターを使用して、Amazon S3 に保存されているバックアップの暗号化を有効にすることができます。詳細は、**バックアップストレージの場所の YAML** を参照してください。次のサンプルは、**DataProtectionApplication** リソースを設定するときに AWS KMS キー ID を指定します。

```
spec:
  backupLocations:
    - velero:
        config:
```

```
      kmsKeyId: 502b409c-4da1-419f-a16e-eif453b3i49f
      profile: default
      region: us-east-1
```

他のストレージプロバイダーの設定可能なすべてのパラメーターについては、**Velero がサポートするストレージプロバイダー** を参照してください。

1.1.7.2. 関連情報

- バックアップストレージの場所の YAML を参照してください。
- Velero のサポート対象ストレージプロバイダー を参照してください。
- バックアップまたは復元設定の検証 に戻ってください。

1.1.8. 高度な設定のバックアップと復元

次のセクションを参照して、バックアップと復元をさらに詳細に設定できます。

1.1.8.1. リソース要求および制限のカスタマイズ

Velero の初回インストール時に、Velero Pod は以下のサンプルで定義されるデフォルトの CPU およびメモリー制限に設定されます。

```
resources:
 limits:
   cpu: "1"
   memory: 256Mi
 requests:
   cpu: 500m
   memory: 128Mi
```

前のサンプルの制限は一部のシナリオでうまく機能しますが、クラスターが多数のリソースをバックアップする場合には更新する必要がある場合があります。たとえば、2000 個のクラスターを管理するハブクラスターでバックアップを実行すると、メモリー不足 (OOM) エラーが原因で、Velero Pod が失敗します。以下の設定では、このシナリオでバックアップを完了できます。

```
  limits:
    cpu: "2"
    memory: 1Gi
  requests:
    cpu: 500m
    memory: 256Mi
```

Velero Pod リソースの制限および要求を更新するには、**DataProtectionApplication** リソースを更新し、Velero Pod の **resourceAllocation** テンプレートを挿入する必要があります。以下のサンプルを参照してください。

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero
  namespace: open-cluster-management-backup
spec:
```

```
...
  configuration:
...
    velero:
      podConfig:
        resourceAllocations:
          limits:
            cpu: "2"
            memory: 1Gi
          requests:
            cpu: 500m
            memory: 256Mi
```

1.1.8.2. 関連情報

- **DataProtectionApplication** パラメーターの詳細は、Red Hat OpenShift Container Platform ドキュメントの デフォルトの Velero クラウドプロバイダーのプラグイン トピックを参照してください。
- クラスターの使用状況に基づくバックアップおよびリストアの CPU およびメモリー要件 の詳細は、OpenShift Container Platform ドキュメントの設定の CPU およびメモリー要件トピックを参照してください。

1.2. VOLSYNC の永続ボリューム複製サービス

VolSync は、レプリケーションの互換性がないストレージタイプが指定されたクラスター全体、または 1 つの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を有効にする Kubernetes Operator です。これは Container Storage Interface (CSI) を使用して互換性の制限を解消します。VolSync Operator を環境にデプロイした後、それを活用して永続データのコピーを作成および保守できます。VolSync は、バージョン 4.13 以降の Red Hat OpenShift Container Platform クラスターでのみ永続ボリューム要求を複製できます。

重要: VolSync は、**Filesystem** の **volumeMode** を使用した永続ボリューム要求の複製のみをサポートします。**volumeMode** を選択しない場合、デフォルトで **Filesystem** になります。

- VolSync を使用した永続ボリュームの複製
  - マネージドクラスターへの VolSync のインストール
  - Rsync-TLS レプリケーションの設定
  - Rsync レプリケーションの設定
  - restic バックアップの設定
  - Rclone レプリケーションの設定
- 複製されたイメージを使用可能な永続的なボリュームクレームに変換
- 同期のスケジューリング

1.2.1. VolSync を使用した永続ボリュームの複製

サポート対象の 3 つの方法を使用して、VolSync で永続ボリュームを複製できます。これは、rsync、rsync-tls、restic、または Rclone などの所有する同期ロケーションの数により異なります。

1.2.1.1. 前提条件

クラスターに VolSync をインストールする前に、以下の要件が必要です。

- Red Hat Advanced Cluster Management バージョン 2.10 以降のハブクラスターで実行中で、設定済みの OpenShift Container Platform 環境。
- 同じ Red Hat Advanced Cluster Management ハブクラスターが管理する 2 つ以上のクラスター。
- VolSync で設定しているクラスター間のネットワーク接続。クラスターが同じネットワーク上にない場合は、Submariner multicluster networking and service discovery を設定し、**ServiceType** の **ClusterIP** 値を使用してクラスターをネットワーク化するか、**ServiceType** に **LoadBalancer** の値を指定してロードバランサーを使用できます。
- ソース永続ボリュームに使用するストレージドライバーは、CSI 互換であり、スナップショットをサポートできる必要があります。

1.2.1.2. マネージドクラスターへの VolSync のインストール

VolSync が 1 つのクラスターの永続ボリューム要求を別のクラスターの Persistent Volume Claims に複製できるようにするには、ソースとターゲットの両方のマネージドクラスターに VolSync をインストールする必要があります。

VolSync は独自の namespace を作成しないため、他の OpenShift Container Platform のすべての namespace Operator と同じ namespace にあります。VolSync の Operator 設定に加えた変更は、チャネル更新の手動承認に変更した場合など、同じ namespace 内の他の Operator にも影響します。

2 つの方法のいずれかを使用して、環境内の 2 つのクラスターに VolSync をインストールできます。次のセクションで説明するように、ハブクラスター内の各マネージドクラスターにラベルを追加するか、**ManagedClusterAddOn** を手動で作成して適用することができます。

1.2.1.2.1. ラベルを使用した VolSync のインストール

ラベルを追加して、マネージドクラスターに VolSync をインストールします。

- Red Hat Advanced Cluster Management コンソールから以下のステップを実行します。
  1. 詳細を表示するには、ハブクラスターコンソールの **Clusters** ページからマネージドクラスターの 1 つを選択します。
  2. **Labels** フィールドに、次のラベルを追加します。

     ```
     addons.open-cluster-management.io/volsync=true
     ```

     VolSync サービス Pod はマネージドクラスターにインストールされます。
  3. 他のマネージドクラスターに同じラベルを追加します。
  4. 各マネージドクラスターで次のコマンドを実行して、VolSync Operator がインストールされていることを確認します。

     ```
     oc get csv -n openshift-operators
     ```

     インストール時に VolSync の Operator がリストされています。

- コマンドラインインターフェイスから次の手順を実行します。

  1. ハブクラスターでコマンドラインセッションを開始します。

  2. 次のコマンドを入力して、最初のクラスターにラベルを追加します。

     ```
     oc label managedcluster <managed-cluster-1> "addons.open-cluster-management.io/volsync"="true"
     ```

     managed-cluster-1 をマネージドクラスターの 1 つの名前に置き換えます。

  3. 次のコマンドを入力して、ラベルを 2 番目のクラスターに追加します。

     ```
     oc label managedcluster <managed-cluster-2> "addons.open-cluster-management.io/volsync"="true"
     ```

     managed-cluster-2 を他のマネージドクラスターの名前に置き換えます。

     ManagedClusterAddOn リソースは、対応する各マネージドクラスターの namespace 内のハブクラスターに自動的に作成される必要があります。

1.2.1.2.2. ManagedClusterAddOn を使用した VolSync のインストール

ManagedClusterAddOn を手動で追加して VolSync をマネージドクラスターにインストールするには、次の手順を実行します。

1. ハブクラスターで、次の例のようなコンテンツを含む **volsync-mcao.yaml** という YAML ファイルを作成します。

   ```
   apiVersion: addon.open-cluster-management.io/v1alpha1
   kind: ManagedClusterAddOn
   metadata:
     name: volsync
     namespace: <managed-cluster-1-namespace>
   spec: {}
   ```

   managed-cluster-1-namespace を、マネージドクラスターの 1 つの namespace に置き換えます。この namespace は、マネージドクラスターの名前と同じです。

   注: 名前は **volsync** である必要があります。

2. 次の例のようなコマンドを入力して、ファイルを設定に適用します。

   ```
   oc apply -f volsync-mcao.yaml
   ```

3. 他のマネージドクラスターに対して手順を繰り返します。
   ManagedClusterAddOn リソースは、対応する各マネージドクラスターの namespace 内のハブクラスターに自動的に作成される必要があります。

1.2.1.2.3. VolSync の ManagedClusterAddOn の更新

使用している Red Hat Advanced Cluster Management のバージョンによっては、VolSync のバージョンを更新する必要がある場合があります。VolSync の **ManagedClusterAddOn** リソースを更新するには、次の手順を実行します。

1. 次のアノテーションを **ManagedClusterAddOn** リソースに追加します。

```
annotations:
    operator-subscription-channel: stable-0.9
```

2. VolSync のデプロイ元となる **operator-subscription-channel** を定義します。

3. **ManagedClusterAddOn** リソースに移動し、選択した **operator-subscription-channel** が含まれていることを確認して、Volsync バージョンが更新されたことを確認します。

1.2.1.3. Rsync-TLS レプリケーションの設定

Rsync-TLS レプリケーションを使用して、永続ボリュームの 1:1 非同期レプリケーションを作成できます。Rsync-TLS ベースのレプリケーションを災害復旧やリモートサイトへのデータ送信に使用できます。Rsync-TLS を使用する場合、VolSync は、stunnel によって提供される TLS で保護されたトンネル全体で Rsync を使用してデータを同期します。詳細は stunnel のドキュメント を参照してください。

次の例は、Rsync-TLS メソッドを使用して設定する方法を示しています。Rsync-TLS の追加情報は、VolSync ドキュメントの Usage を参照してください。

1.2.1.3.1. マネージドクラスター全体での Rsync-TLS レプリケーションの設定

Rsync-TLS ベースのレプリケーションの場合、ソースクラスターと宛先クラスターでカスタムリソースを設定します。カスタムリソースは、**address** 値を使用して送信元を宛先に接続し、stunnel によって提供される TLS で保護されたトンネルを使用して、転送されたデータの安全性を確保します。

Rsync-TLS レプリケーションを、**source-ns** namespace の **source** クラスターの永続ボリュームクレームから **destination-ns** namespace の **destination** クラスターの永続ボリュームクレームに設定するには、次の情報と例を参照してください。必要に応じて値を置き換えます。

1. 宛先クラスターを設定します。

   a. 宛先クラスターで次のコマンドを実行して、ネームスペースを作成します。

   ```
   oc create ns <destination-ns>
   ```

   destination-ns をレプリケーション先が配置されている namespace に置き換えます。

   b. **replication_destination** という名前の新しい YAML ファイルを作成し、次の内容をコピーします。

```
apiVersion: volsync.backube/v1alpha1
kind: ReplicationDestination
metadata:
  name: <destination>
  namespace: <destination-ns>
spec:
  rsyncTLS:
    serviceType: LoadBalancer (1)
    copyMethod: Snapshot
    capacity: 2Gi (2)
    accessModes: [ReadWriteOnce]
    storageClassName: gp2-csi
    volumeSnapshotClassName: csi-aws-vsc
```

1. この例では、**LoadBalancer** の **ServiceType** 値が使用されます。ロードバランサーサービスはソースクラスターによって作成され、ソースマネージドクラスターが別の
2. **capacity** 値が、レプリケートされている永続ボリューム要求の容量と一致していることを確認してください。

任意: 環境のデフォルト値とは異なるストレージクラス名とボリュームスナップショットクラス名を使用している場合は、**storageClassName** パラメーターと **volumeSnapshotClassName** パラメーターの値を指定します。

c. 宛先クラスターで以下のコマンドを実行し、**replicationdestination** リソースを作成します。

```
oc create -n <destination-ns> -f replication_destination.yaml
```

destination-ns は、宛先の namespace の名前に置き換えます。

replicationdestination リソースが作成されると、以下のパラメーターおよび値がリソースに追加されます。

パラメーター	値
.status.rsyncTLS.address	送信元クラスターと宛先クラスターが通信できるようにするために使用される宛先クラスターの IP アドレス。
.status.rsyncTLS.keySecret	ソースクラスターとの接続を認証する TLS キーを含むシークレットの名前。

d. 以下のコマンドを実行して、ソースクラスターで使用する **.status.rsyncTLS.address** の値をコピーします。**destination** は、レプリケーション先のカスタムリソースの名前に置き換えます。**destination-ns** は、宛先の namespace の名前に置き換えます。

```
ADDRESS=`oc get replicationdestination <destination> -n <destination-ns> --template=
{{.status.rsyncTLS.address}}`
echo $ADDRESS
```

出力は次のようになります。これは Amazon Web Services 環境のものです。

```
a831264645yhrjrjyer6f9e4a02eb2-5592c0b3d94dd376.elb.us-east-1.amazonaws.com
```

e. 次のコマンドを実行して、シークレットの名前をコピーします。

```
KEYSECRET=`oc get replicationdestination <destination> -n <destination-ns> --
template={{.status.rsyncTLS.keySecret}}`
echo $KEYSECRET
```

destination は、レプリケーション先のカスタムリソースの名前に置き換えます。

destination-ns は、宛先の namespace の名前に置き換えます。

ソースの設定時に、ソースクラスターで入力する必要があります。出力は、SSH キーシークレットファイルの名前である必要があります。これは、次の名前のようになります。

```
volsync-rsync-tls-destination-name
```

f. 宛先クラスター宛先クラスターに対して次のコマンドを入力して、宛先クラスターから SSH シークレットをコピーします。

```
oc get secret -n <destination-ns> $KEYSECRET -o yaml > /tmp/secret.yaml
```

destination-ns をレプリケーション先が配置されている namespace に置き換えます。

g. 以下のコマンドを入力して、**vi** エディターでシークレットファイルを開きます。

```
vi /tmp/secret.yaml
```

h. 宛先クラスターのオープンシークレットファイルで、次の変更を行います。

- namespace をソースクラスターの namespace に変更します。この例では、**source-ns** です。
- 所有者の参照を削除します (**.metadata.ownerReferences**)。

i. ソースクラスターで、ソースクラスターで次のコマンドを入力してシークレットファイルを作成します。

```
oc create -f /tmp/secret.yaml
```

2. 複製するソース永続ボリュームクレームを特定します。
注記: ソース永続ボリューム要求は CSI ストレージクラスにある必要があります。

3. **ReplicationSource** アイテムを作成します。

a. ソースクラスター上に **replication_source** という名前の新しい YAML ファイルを作成し、次の内容をコピーします。

```
apiVersion: volsync.backube/v1alpha1
kind: ReplicationSource
metadata:
  name: <source> 1
  namespace: <source-ns> 2
spec:
  sourcePVC: <persistent_volume_claim> 3
  trigger:
    schedule: "*/3 * * * *" #/*
  rsyncTLS:
    keySecret: <mykeysecret> 4
    address: <my.host.com> 5
    copyMethod: Snapshot
    storageClassName: gp2-csi
    volumeSnapshotClassName: csi-aws-vsc
```

1 **source** は、レプリケーションソースカスタムリソースの名前に置き換えます。これを自動的に置き換える方法については、この手順のステップ **3-vi** を参照してください。

2. **source-ns** をソースが置かれている Persistent Volume Claim の namespace に置き換えます。これを自動的に置き換える方法については、この手順のステップ **3-vi** を参照
3. **persistent_volume_claim** は、ソース永続ボリューム要求の名前に置き換えます。
4. **mykeysecret** を宛先クラスターからソースクラスターにコピーしたシークレットの名前 (**$KEYSECRET** の値) に置き換えます。
5. **my.host.com** は、設定時に **ReplicationDestination** の **.status.rsyncTLS.address** フィールドからコピーしたホストアドレスに置き換えます。**sed** コマンドの例は次のステップで見つけることができます。

ストレージドライバーがクローン作成をサポートする場合は、**copyMethod** の値に **Clone** を使用すると、レプリケーションのより効率的なプロセスになる可能性があります。

任意: 環境のデフォルト値とは異なるストレージクラス名とボリュームスナップショットクラス名を使用している場合は、**storageClassName** パラメーターと **volumeSnapshotClassName** パラメーターの値を指定します。

永続ボリュームの同期方法を設定できるようになりました。

b. ソースクラスターで、以下のコマンドを入力して **ReplicationSource** オブジェクトの **address** および **keySecret** の値を、宛先クラスターから書き留めた値に置き換えて **replication_source.yaml** ファイルを変更します。

```
sed -i "s/<my.host.com>/$ADDRESS/g" replication_source.yaml
sed -i "s/<mykeysecret>/$KEYSECRET/g" replication_source.yaml
oc create -n <source> -f replication_source.yaml
```

my.host.com は、設定時に **ReplicationDestination** の **.status.rsyncTLS.address** フィールドからコピーしたホストアドレスに置き換えます。

keySecret を設定時に **ReplicationDestination** の **.status.rsyncTLS.keySecret** フィールドからコピーしたキーに置き換えます。

source を、ソース が置かれている永続ボリューム要求の名前に置き換えます。

注記: 複製する永続ボリューム要求と同じ namespace にファイルを作成する必要があります。

c. **ReplicationSource** オブジェクトで以下のコマンドを実行して、レプリケーションが完了したことを確認します。

```
oc describe ReplicationSource -n <source-ns> <source>
```

source-ns をソースが置かれている Persistent Volume Claim の namespace に置き換えます。

source は、レプリケーションソースのカスタムリソースの名前に置き換えます。

レプリケーションが成功した場合、出力は次の例のようになります。

```
Status:
  Conditions:
    Last Transition Time:  2021-10-14T20:48:00Z
```

```
  Message:               Synchronization in-progress
  Reason:                SyncInProgress
  Status:                True
  Type:                  Synchronizing
  Last Transition Time:  2021-10-14T20:41:41Z
  Message:               Reconcile complete
  Reason:                ReconcileComplete
  Status:                True
  Type:                  Reconciled
 Last Sync Duration:     5m20.764642395s
 Last Sync Time:         2021-10-14T20:47:01Z
 Next Sync Time:         2021-10-14T20:48:00Z
```

Last Sync Time に時間がリストされていない場合は、レプリケーションが完了していません。

元の永続ボリュームのレプリカがあります。

1.2.1.4. Rsync レプリケーションの設定

重要: セキュリティーを強化するには、Rsync の代わりに Rsync-TLS を使用してください。Rsync-TLS を使用すると、永続ボリュームのレプリケーションに必要のない昇格されたユーザー権限の使用を回避できます。

Rsync レプリケーションを使用して、永続ボリュームの 1:1 非同期レプリケーションを作成できます。Rsync ベースのレプリケーションを災害復旧やリモートサイトへのデータ送信に使用できます。

次の例は、Rsync メソッドを使用して設定する方法を示しています。

1.2.1.4.1. マネージドクラスター間での Rsync レプリケーションの設定

Rsync ベースのレプリケーションの場合は、ソースクラスターおよび宛先クラスターでカスタムリソースを設定します。カスタムリソースは、**address** 値を使用してソースを宛先に接続し、**sshKeys** を使用して転送されたデータがセキュアであることを確認します。

注記: address および **sshKeys** の値を宛先からソースにコピーし、ソースを設定する前に宛先を設定する必要があります。

この例では、**source-ns** namespace の **source** クラスターの永続ボリューム要求から **destination-ns** namespace の **destination** クラスターの永続ボリューム要求に Rsync レプリケーションを設定する手順を説明します。必要に応じて、これらの値を他の値に置き換えることができます。

1. 宛先クラスターを設定します。

   a. 宛先クラスターで次のコマンドを実行して、ネームスペースを作成します。

   ```
   oc create ns <destination-ns>
   ```

   destination-ns を、オンサイトのボリューム要求ごとに宛先が含まれる namespace の名前に置き換えます。

   b. 以下の YAML コンテンツをコピーし、**replication_destination.yaml** という名前の新規ファイルを作成します。

   ```
   apiVersion: volsync.backube/v1alpha1
   kind: ReplicationDestination
   ```

```
  metadata:
    name: <destination>
    namespace: <destination-ns>
  spec:
    rsync:
      serviceType: LoadBalancer
      copyMethod: Snapshot
      capacity: 2Gi
      accessModes: [ReadWriteOnce]
      storageClassName: gp2-csi
      volumeSnapshotClassName: csi-aws-vsc
```

注記: capacity の値は、レプリケートされる永続ボリューム要求 (PVC) の容量と一致する必要があります。

destination は、宛先 CR の名前に置き換えます。

destination-ns は、宛先の namespace の名前に置き換えます。

この例では、**LoadBalancer** の **ServiceType** 値が使用されます。ロードバランサーサービスはソースクラスターによって作成され、ソースマネージドクラスターが別の宛先マネージドクラスターに情報を転送できるようにします。ソースと宛先が同じクラスター上にある場合、または Submariner ネットワークサービスが設定されている場合は、サービスタイプとして **ClusterIP** を使用できます。ソースクラスターを設定するときに参照するシークレットのアドレスと名前をメモします。

storageClassName および **volumeSnapshotClassName** は任意のパラメーターです。特に、環境のデフォルト値とは異なるストレージクラスおよびボリュームスナップショットクラス名を使用している場合は、環境の値を指定してください。

c. 宛先クラスターで以下のコマンドを実行し、**replicationdestination** リソースを作成します。

```
oc create -n <destination-ns> -f replication_destination.yaml
```

destination-ns は、宛先の namespace の名前に置き換えます。

replicationdestination リソースが作成されると、以下のパラメーターおよび値がリソースに追加されます。

パラメーター	値
.status.rsync.address	送信元クラスターと宛先クラスターが通信できるようにするために使用される宛先クラスターの IP アドレス。
.status.rsync.sshKeys	ソースクラスターから宛先クラスターへの安全なデータ転送を可能にする SSH キーファイルの名前。

d. 以下のコマンドを実行して、ソースクラスターで使用する **.status.rsync.address** の値をコピーします。

```
ADDRESS=`oc get replicationdestination <destination> -n <destination-ns> --template=
{{.status.rsync.address}}`
echo $ADDRESS
```

destination は、レプリケーション先のカスタムリソースの名前に置き換えます。

destination-ns は、宛先の namespace の名前に置き換えます。

出力は、Amazon Web Services 環境の次の出力のように表示されます。

```
a831264645yhrjrjyer6f9e4a02eb2-5592c0b3d94dd376.elb.us-east-1.amazonaws.com
```

e. 次のコマンドを実行して、シークレットの名前をコピーします。

```
SSHKEYS=`oc get replicationdestination <destination> -n <destination-ns> --template=
{{.status.rsync.sshKeys}}`
echo $SSHKEYS
```

destination は、レプリケーション先のカスタムリソースの名前に置き換えます。

destination-ns は、宛先の namespace の名前に置き換えます。

ソースの設定時に、ソースクラスターで入力する必要があります。出力は、SSH キーシークレットファイルの名前である必要があります。これは、次の名前のようになります。

```
volsync-rsync-dst-src-destination-name
```

f. 宛先クラスターに対して次のコマンドを入力して、宛先クラスターから SSH シークレットをコピーします。

```
oc get secret -n <destination-ns> $SSHKEYS -o yaml > /tmp/secret.yaml
```

destination-ns を、宛先が置かれている永続ボリューム要求の namespace に置き換えます。

g. 以下のコマンドを入力して、**vi** エディターでシークレットファイルを開きます。

```
vi /tmp/secret.yaml
```

h. 宛先クラスターのオープンシークレットファイルで、次の変更を行います。

- namespace をソースクラスターの namespace に変更します。この例では、**source-ns** です。
- 所有者の参照を削除します (**.metadata.ownerReferences**)。

i. ソースクラスターで、ソースクラスターで次のコマンドを入力してシークレットファイルを作成します。

```
oc create -f /tmp/secret.yaml
```

2. 複製するソース永続ボリュームクレームを特定します。
注記: ソース永続ボリューム要求は CSI ストレージクラスにある必要があります。

3. **ReplicationSource** アイテムを作成します。

   a. 以下の YAML コンテンツをコピーして、ソースクラスターに **replication_source.yaml** という名前の新規ファイルを作成します。

```
apiVersion: volsync.backube/v1alpha1
kind: ReplicationSource
metadata:
  name: <source>
  namespace: <source-ns>
spec:
  sourcePVC: <persistent_volume_claim>
  trigger:
    schedule: "*/3 * * * *" #/*
  rsync:
    sshKeys: <mysshkeys>
    address: <my.host.com>
    copyMethod: Snapshot
    storageClassName: gp2-csi
    volumeSnapshotClassName: csi-aws-vsc
```

   source は、レプリケーションソースカスタムリソースの名前に置き換えます。これを自動的に置き換える方法については、この手順のステップ **3-vi** を参照してください。

   source-ns をソースが置かれている Persistent Volume Claim の namespace に置き換えます。これを自動的に置き換える方法については、この手順のステップ **3-vi** を参照してください。

   persistent_volume_claim は、ソース永続ボリューム要求の名前に置き換えます。

   mysshkeys は、設定時に **ReplicationDestination** の **.status.rsync.sshKeys** フィールドからコピーしたキーに置き換えます。

   my.host.com は、設定時に **ReplicationDestination** の **.status.rsync.address** フィールドからコピーしたホストアドレスに置き換えます。

   ストレージドライバーがクローン作成をサポートする場合は、**copyMethod** の値に **Clone** を使用すると、レプリケーションのより効率的なプロセスになる可能性があります。

   storageClassName および **volumeSnapshotClassName** はオプションのパラメーターです。ご使用の環境のデフォルトとは異なるストレージクラスおよびボリュームスナップショットクラス名を使用している場合は、それらの値を指定してください。

   永続ボリュームの同期方法を設定できるようになりました。

   b. ソースクラスターで、以下のコマンドを入力して **ReplicationSource** オブジェクトの **address** および **sshKeys** の値を、宛先クラスターから書き留めた値に置き換えて **replication_source.yaml** ファイルを変更します。

```
sed -i "s/<my.host.com>/$ADDRESS/g" replication_source.yaml
sed -i "s/<mysshkeys>/$SSHKEYS/g" replication_source.yaml
oc create -n <source> -f replication_source.yaml
```

   my.host.com は、設定時に **ReplicationDestination** の **.status.rsync.address** フィールドからコピーしたホストアドレスに置き換えます。

   mysshkeys は、設定時に **ReplicationDestination** の **.status.rsync.sshKeys** フィールド

myssshkeys は、設定時に **ReplicationDestination** の **.status.rsync.sshKeys** フィールドからコピーしたキーに置き換えます。

source を、ソース が置かれている永続ボリューム要求の名前に置き換えます。

注記: 複製する永続ボリューム要求と同じ namespace にファイルを作成する必要があります。

c. **ReplicationSource** オブジェクトで以下のコマンドを実行して、レプリケーションが完了したことを確認します。

```
oc describe ReplicationSource -n <source-ns> <source>
```

source-ns をソースが置かれている Persistent Volume Claim の namespace に置き換えます。

source は、レプリケーションソースのカスタムリソースの名前に置き換えます。

レプリケーションが成功した場合、出力は次の例のようになります。

```
Status:
  Conditions:
    Last Transition Time:  2021-10-14T20:48:00Z
    Message:               Synchronization in-progress
    Reason:                SyncInProgress
    Status:                True
    Type:                  Synchronizing
    Last Transition Time:  2021-10-14T20:41:41Z
    Message:               Reconcile complete
    Reason:                ReconcileComplete
    Status:                True
    Type:                  Reconciled
  Last Sync Duration:      5m20.764642395s
  Last Sync Time:          2021-10-14T20:47:01Z
  Next Sync Time:          2021-10-14T20:48:00Z
```

Last Sync Time に時間がリストされていない場合は、レプリケーションが完了していません。

元の永続ボリュームのレプリカがあります。

1.2.1.5. restic バックアップの設定

Restic ベースのバックアップは、永続ボリュームの Restic ベースのバックアップコピーを、**restic-config.yaml** シークレットファイルで指定された場所にコピーします。Restic バックアップは、クラスター間でデータを同期しませんが、データをバックアップします。

次の手順を実行して、restic ベースのバックアップを設定します。

1. 次の YAML コンテンツのようなシークレットを作成して、バックアップイメージが保存されるリポジトリーを指定します。

```
apiVersion: v1
kind: Secret
metadata:
  name: restic-config
```

```
type: Opaque
stringData:
  RESTIC_REPOSITORY: <my-restic-repository>
  RESTIC_PASSWORD: <my-restic-password>
  AWS_ACCESS_KEY_ID: access
  AWS_SECRET_ACCESS_KEY: password
```

my-restic-repository は、バックアップファイルを保存する S3 バケットリポジトリーの場所に置き換えます。

my-restic-password は、リポジトリーへのアクセスに必要な暗号化キーに置き換えます。

必要に応じて、**アクセス** と **パスワード** は、プロバイダーのクレデンシャルに置き換えます。

新しいリポジトリーを準備する必要がある場合の手順は、新しいリポジトリーの準備 を参照してください。この手順を使用する場合は、**restic init** コマンドを実行してリポジトリーを初期化する必要がある手順をスキップしてください。VolSync は、最初のバックアップ中にリポジトリーを自動的に初期化します。

重要: 複数の永続ボリューム要求を同じ S3 バケットにバックアップする場合には、バケットへのパスは永続ボリュームクレームごとに一意である必要があります。各永続ボリュームクレームは個別の **ReplicationSource** でバックアップされるので、個別の restic-config シークレットが必要です。

同じ S3 バケットを共有することで、各 **ReplicationSource** は S3 バケット全体への書き込みアクセスが割り当てられます。

2. 次の YAML コンテンツに似た **ReplicationSource** オブジェクトを作成して、バックアップポリシーを設定します。

```
apiVersion: volsync.backube/v1alpha1
kind: ReplicationSource
metadata:
  name: mydata-backup
spec:
  sourcePVC: <source>
  trigger:
    schedule: "*/30 * * * *" #\*
  restic:
    pruneIntervalDays: 14
    repository: <restic-config>
    retain:
      hourly: 6
      daily: 5
      weekly: 4
      monthly: 2
      yearly: 1
    copyMethod: Clone
  # The StorageClass to use when creating the PiT copy (same as source PVC if omitted)
  #storageClassName: my-sc-name
  # The VSC to use if the copy method is Snapshot (default if omitted)
  #volumeSnapshotClassName: my-vsc-name
```

source は、バックアップしている永続ボリュームクレームに置き換えます。

schedule の値は、バックアップを実行する頻度に置き換えます。この例では、30 分ごとにスケジュールが指定されています。スケジュールの設定の詳細は、Synchronization のスケジュール を参照してください。

PruneIntervalDays の値は、インスタンスで次にデータの圧縮するまでの経過時間 (日数) に置き換えて、スペースを節約します。プルーニング操作は、実行中に大量の I/O トラフィックを生成する可能性があります。

restic-config は、ステップ 1 で作成したシークレットの名前に置き換えます。

retain の値は、バックアップしたイメージの保持ポリシーに設定します。

ベストプラクティス: **CopyMethod** の値に **Clone** を使用して、特定の時点のイメージが確実に保存されるようにします。

注記: デフォルトでは、restic ムーバーは root 権限なしで実行されます。restic ムーバーを root として実行する場合は、次のコマンドを実行して、昇格された権限のアノテーションを namespace に追加します。

```
oc annotate namespace <namespace> volsync.backube/privileged-movers=true
```

<namespace> を namespace の名前に置き換えます。

1.2.1.5.1. restic バックアップの復元

コピーされたデータを restic バックアップから新しい永続ボリューム要求に復元できます。**ベストプラクティス:** バックアップ 1 つだけを新しい永続ボリューム要求に復元します。Restic バックアップを復元するには、次の手順を実行します。

1. 次の例のように、新しいデータを含む新しい永続ボリュームクレームを作成します。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <pvc-name>
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

pvc-name は、新しい永続ボリュームクレームの名前に置き換えます。

2. 次の例のような **ReplicationDestination** カスタムリソースを作成して、データの復元先を指定します。

```
apiVersion: volsync.backube/v1alpha1
kind: ReplicationDestination
metadata:
  name: <destination>
spec:
  trigger:
    manual: restore-once
  restic:
```

```
    repository: <restic-repo>
    destinationPVC: <pvc-name>
    copyMethod: Direct
```

destination は、宛先 CR の名前に置き換えます。

restic-repo は、ソースが保存されているリポジトリーへのパスに置き換えます。

pvc-name は、データを復元する新しい永続ボリュームクレームの名前に置き換えます。これには、新しいボリューム要求をプロビジョニングするのではなく、既存の永続ボリューム要求を使用してください。

復元プロセスは 1 回だけ完了する必要があります。この例では、最新のバックアップを復元します。復元オプションの詳細は、VolSync ドキュメントの Restore options を参照してください。

1.2.1.6. Rclone レプリケーションの設定

Rclone バックアップは、Rclone を使用して AWS S3 などの中間オブジェクトストレージの場所を介して単一の永続ボリュームを複数の場所にコピーします。複数の場所にデータを配布する場合に役立ちます。

次の手順を実行して、Rclone レプリケーションを設定します。

1. 次の例のような **ReplicationSource** カスタムリソースを作成します。

```
apiVersion: volsync.backube/v1alpha1
kind: ReplicationSource
metadata:
  name: <source>
  namespace: <source-ns>
spec:
  sourcePVC: <source-pvc>
  trigger:
    schedule: "*/6 * * * *" #\*
  rclone:
    rcloneConfigSection: <intermediate-s3-bucket>
    rcloneDestPath: <destination-bucket>
    rcloneConfig: <rclone-secret>
    copyMethod: Snapshot
    storageClassName: <my-sc-name>
    volumeSnapshotClassName: <my-vsc>
```

source-pvc は、レプリケーションソースのカスタムリソースの名前に置き換えます。

source-ns をソースが置かれている Persistent Volume Claim の namespace に置き換えます。

source は、レプリケートしている永続ボリュームクレームに置き換えます。

スケジュール の値は、レプリケーションを実行する頻度に置き換えます。この例では、6 分ごとにスケジュールが指定されています。この値は引用符で囲む必要があります。詳しくは Synchronization のスケジュール を参照してください。

Intermediate-s3-bucket は、Rclone 設定ファイルの設定セクションへのパスに置き換えます。

destination-bucket は、レプリケートされたファイルをコピーするオブジェクトバケットへのパスに置き換えます。

rclone-secret は、Rclone 設定情報を含むシークレットの名前に置き換えます。

copyMethod の値は **Clone**、**Direct**、または **Snapshot** として設定します。この値は、ある特定の時点でのコピーを生成するかどうか、生成する場合は、生成方法を指定します。

my-sc-name は、ポイントインタイムコピーに使用するストレージクラスの名前に置き換えます。指定しない場合、ソースボリュームのストレージクラスが使用されます。

スナップショット を **copyMethod** として指定した場合は **my-vsc** を使用する **VolumeSnapshotClass** の名前に置き換えます。これは、他のタイプの **copyMethod** には必要ありません。

2. 次の例のような **ReplicationDestination** カスタムリソースを作成します。

```
apiVersion: volsync.backube/v1alpha1
kind: ReplicationDestination
metadata:
  name: database-destination
  namespace: dest
spec:
  trigger:
    schedule: "3,9,15,21,27,33,39,45,51,57 * * * *" #/*
  rclone:
    rcloneConfigSection: <intermediate-s3-bucket>
    rcloneDestPath: <destination-bucket>
    rcloneConfig: <rclone-secret>
    copyMethod: Snapshot
    accessModes: [ReadWriteOnce]
    capacity: 10Gi
    storageClassName: <my-sc>
    volumeSnapshotClassName: <my-vsc>
```

スケジュール の値は、レプリケーションを宛先に移動する頻度に置き換えます。移動元と宛先のスケジュールをオフセットして、データが宛先からプルされる前に複製を完了できるようにする必要があります。この例では、オフセットは 3 分で、6 分間隔でスケジュールされています。この値は引用符で囲む必要があります。スケジュールの詳細は、同期のスケジュール を参照してください。

Intermediate-s3-bucket は、Rclone 設定ファイルの設定セクションへのパスに置き換えます。

destination-bucket は、レプリケートされたファイルをコピーするオブジェクトバケットへのパスに置き換えます。

rclone-secret は、Rclone 設定情報を含むシークレットの名前に置き換えます。

copyMethod の値は **Clone**、**Direct**、または **Snapshot** として設定します。この値は、ある特定の時点でのコピーを生成するかどうか、生成する場合は、生成方法を指定します。

accessModes の値は、永続ボリュームクレームのアクセスモードを指定します。有効な値は **ReadWriteOnce** または **ReadWriteMany** です。

容量 は宛先ボリュームのサイズを指定します。このサイズは、着信データを格納するのに十分な大きさに指定します。

my-sc は、特定の時点のコピーの宛先として使用するストレージクラスの名前に置き換えます。指定しない場合、システムストレージクラスが使用されます。

スナップショット を **copyMethod** として指定した場合は **my-vsc** を使用する **VolumeSnapshotClass** の名前に置き換えます。これは、他のタイプの **copyMethod** には必要ありません。含まれていない場合は、システムのデフォルトの **VolumeSnapshotClass** が使用されます。

注記: デフォルトでは、rclone ムーバーは root 権限なしで実行されます。rclone ムーバーを root として実行する場合は、次のコマンドを実行して、昇格された権限のアノテーションを namespace に追加します。

```
oc annotate namespace <namespace> volsync.backube/privileged-movers=true
```

<namespace> を namespace の名前に置き換えます。

1.2.1.7. 関連情報

詳細については、以下のトピックを参照してください。

- Rsync-TLS レプリケーション用の独自のシークレットを作成する 方法は、Rsync-TLS レプリケーション用のシークレットの作成を参照してください。
- Rsync の追加情報については、VolSync ドキュメントの Usage を参照してください。
- リスティックオプションの詳細は、VolSync ドキュメントの バックアップオプション を参照してください。
- Installing VolSync on the managed clusters に戻る

1.2.2. 複製されたイメージを使用可能な永続的なボリュームクレームに変換

データを復元するには、レプリケートされたイメージを永続ボリューム要求に変換する必要がある場合があります。

VolumeSnapshot を使用して **ReplicationDestination** の場所から永続ボリューム要求を複製または復元すると、**VolumeSnapshot** が作成されます。**VolumeSnapshot** には、最後に成功した同期からの **latestImage** が含まれます。イメージのコピーは、使用する前に永続的なボリュームクレームに変換する必要があります。VolSync **ReplicationDestination** ボリュームポピュレーターを使用すると、イメージのコピーを使用可能な永続ボリュームクレームに変換できます。

1. 永続ボリューム要求を復元する **ReplicationDestination** を参照する **dataSourceRef** で永続ボリューム要求を作成します。この永続ボリューム要求には、**ReplicationDestination** カスタムリソース定義の **status.latestImage** 設定で指定された **VolumeSnapshot** の内容が設定されます。
   次の YAML コンテンツは、使用される可能性のある永続ボリューム要求のサンプルを示しています。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: <pvc-name>
  namespace: <destination-ns>
spec:
  accessModes:
    - ReadWriteOnce
  dataSourceRef:
    kind: ReplicationDestination
```

```
    apiGroup: volsync.backube
    name: <replicationdestination_to_replace>
  resources:
    requests:
      storage: 2Gi
```

pvc-name は、新規の永続ボリューム要求 (PVC) の名前に置き換えます。

destination-ns は、永続ボリューム要求および **ReplicationDestination** が置かれている namespace に置き換えます。

replicationdestination_to_replace は **ReplicationDestination** 名に置き換えます。

ベストプラクティス: 値が少なくとも最初のソース永続ボリュームクレームと同じサイズである場合は、**resources.requests.storage** を異なる値で更新できます。

2. 次のコマンドを入力して、永続ボリュームクレームが環境で実行されていることを確認します。

```
$ kubectl get pvc -n <destination-ns>
```

注記:

latestImage が存在しない場合、永続ボリューム要求は **ReplicationDestination** が完了し、スナップショットが利用可能になるまで保留状態のままになります。**ReplicationDestination** と、**ReplicationDestination** を使用する永続ボリュームコントローラーを同時に作成できます。永続ボリューム要求は、**ReplicationDestination** がレプリケーションを完了し、スナップショットが使用可能になった後にのみ、ボリューム作成プロセスを開始します。スナップショットは、**.status.latestImage** にあります。

さらに、使用されているストレージクラスの **volumeBindingMode** 値が **WaitForFirstConsumer** である場合、ボリュームポピュレーターは、永続ボリューム要求のコンシューマーが存在するまで待機してから、永続ボリューム要求が読み込まれます。永続ボリューム要求をマウントする Pod など、コンシューマーにアクセス権が必要な場合、そのボリュームにはデータが入力されます。VolSync ボリュームポピュレーターコントローラーは、**ReplicationDestination** の **latestImage** を使用します。**latestImage** は、永続ボリューム制御の作成後にレプリケーションが完了するたびに更新されます。

1.2.3. 同期のスケジューリング

レプリケーションの開始方法を決定するときは、常に実行する、スケジュールどおりに実行する、または手動で実行するという 3 つのオプションから選択します。レプリケーションのスケジュールは、よく選択されるオプションです。

スケジュール オプションは、スケジュール時にレプリケーションを実行します。スケジュールは **cronspec** で定義されるため、スケジュールを時間間隔または特定の時間として設定できます。スケジュールの値の順序は次のとおりです。

"minute (0-59) hour (0-23) day-of-month (1-31) month (1-12) day-of-week (0-6)"

レプリケーションはスケジュールされた時間に開始されます。このレプリケーションオプションの設定は、以下の内容のようになります。

```
spec:
  trigger:
    schedule: "*/6 * * * *"
```

これらの方法のいずれかを有効にしたら、設定した方法に従って同期スケジュールが実行されます。

追加情報およびオプションについては、VolSync のドキュメントを参照してください。

1.2.4. VolSync の詳細設定

永続ボリュームをレプリケートするときに、独自のシークレットを作成するなど、VolSync をさらに設定できます。

1.2.4.1. Rsync-TLS レプリケーション用のシークレットの作成

送信元と宛先は、TLS 接続の共有キーにアクセスできる必要があります。キーの場所は **keySecret** フィールドで確認できます。**.spec.rsyncTLS.keySecret** にシークレット名を指定しない場合、シークレット名は自動的に生成され、**.status.rsyncTLS.keySecret** に追加されます。

独自のシークレットを作成するには、次の手順を実行します。

1. シークレットには次の形式を使用します: **<id>:<at_least_32_hex_digits>**
   次の例を参照してください: **1:23b7395fafc3e842bd8ac0fe142e6ad1**

2. 前の例に対応する次の **Secret.yaml** の例を参照してください。

```
apiVersion: v1
data:
  # echo -n 1:23b7395fafc3e842bd8ac0fe142e6ad1 | base64
  psk.txt: MToyM2I3Mzk1ZmFmYzNlODQyYmQ4YWMwZmUxNDJlNmFkMQ==
kind: Secret
metadata:
  name: tls-key-secret
type: Opaque
```