Red Hat Advanced Cluster Management for Kubernetes 2.10

业务连续性

业务连续性

Last Updated: 2024-07-01

Red Hat Advanced Cluster Management for Kubernetes 2.10 业务连续性

业务连续性

法律通告

摘要

了解更多有关恢复集群、灾难恢复等的信息。

目录

第 1 章 业务连续性

有关灾难恢复解决方案和 hub 集群以及受管集群，请参阅以下主题。

- 备份和恢复
  - 备份和恢复 Operator 架构
  - 配置主动被动 hub 集群
  - 安装备份和恢复 Operator
  - 调度和恢复备份
- 使用 VolSync 复制持久性卷
  - 使用 VolSync 复制持久性卷
  - 将复制镜像转换为可用的持久性卷声明
  - 调度同步

1.1. 备份和恢复

集群备份和恢复 Operator 在 hub 集群上运行，并为 Red Hat Advanced Cluster Management for Kubernetes hub 集群失败提供灾难恢复解决方案。当 hub 集群失败时，一些功能（如基于策略的警报或集群更新）会停止工作，即使所有受管集群仍可以正常工作。当 hub 集群不可用时，您需要一个恢复计划来决定是否可以恢复，或者是否需要从新部署的 hub 集群中恢复数据。

备份和恢复组件使用一个策略发送警报，以在 hub 集群不可用时通知管理员，并可能需要进行恢复操作。如果备份解决方案无法正常工作，同样的策略会提醒管理员并报告与备份数据相关的问题，即使主 hub 集群处于活跃状态并在管理集群。

集群备份和恢复 Operator 依赖于 OADP Operator 安装 Velero，并从 hub 集群创建到存储数据的备份存储位置的连接。Velero 是运行备份和恢复操作的组件。集群备份和恢复 Operator 解决方案为所有 Red Hat Advanced Cluster Management hub 集群资源（包括受管集群、应用程序和策略）提供备份和恢复支持。

集群备份和恢复 Operator 支持备份扩展 hub 集群安装的任何第三方资源。使用这个备份解决方案，您可以定义基于 cron 的备份计划，这些计划在指定时间段内运行。当 hub 集群停机时，可以部署新的 hub 集群，并将备份的数据移到新的 hub 集群中。

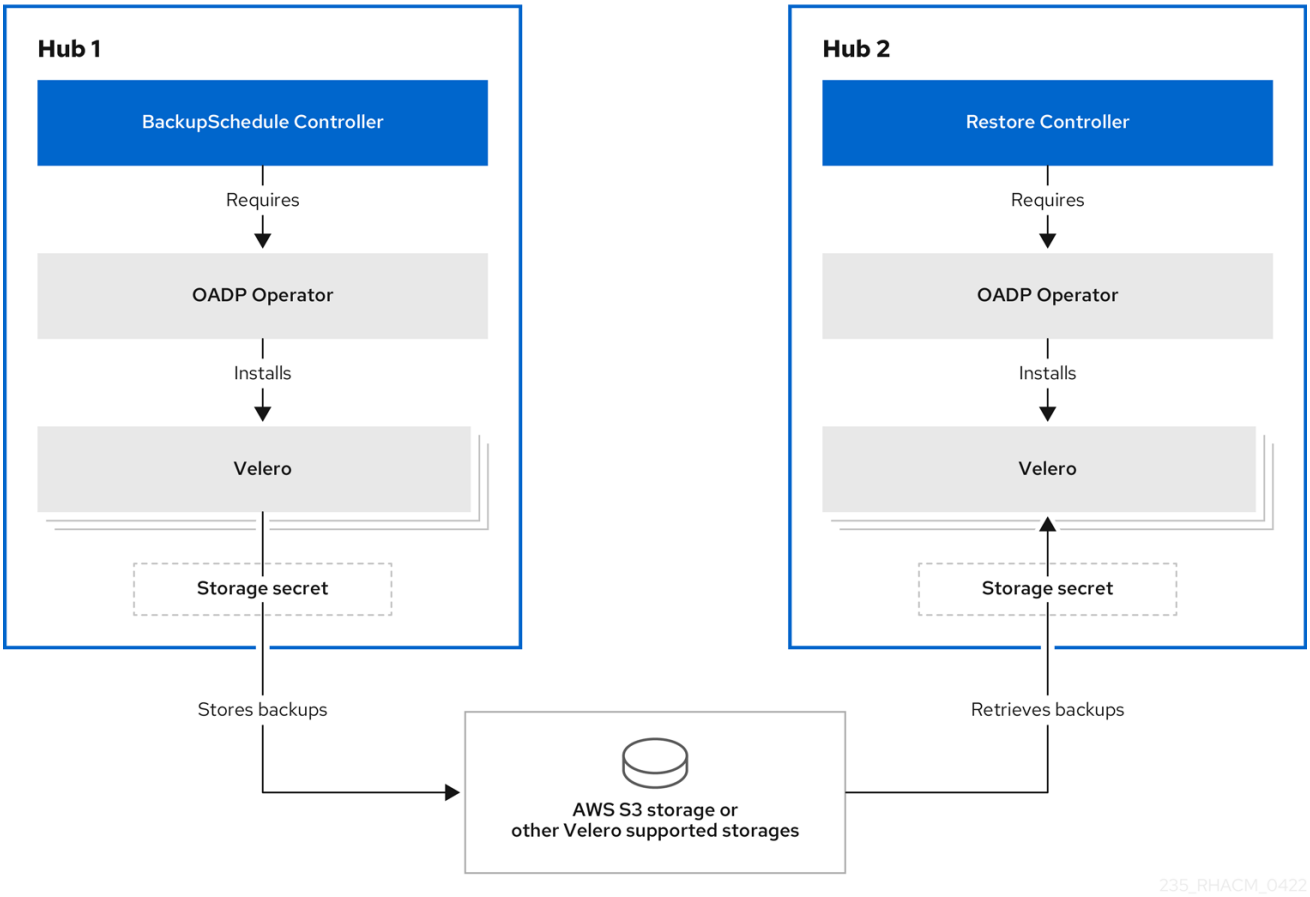
继续阅读以下主题以了解更多有关备份和恢复 Operator 的信息：

- 备份和恢复 Operator 架构
- 配置主动被动 hub 集群
- 安装备份和恢复 Operator
- 调度和恢复备份
- 恢复备份
- 验证备份或恢复配置
- 使用受管服务帐户自动连接集群

- 备份和恢复高级配置

1.1.1. 备份和恢复 Operator 架构

Operator 定义 **BackupSchedule.cluster.open-cluster-management.io** 资源，用于设置 Red Hat Advanced Cluster Management 备份计划，以及 **restore.cluster.open-cluster-management.io** 资源，该资源用于处理和恢复这些备份。Operator 会创建对应的 Velero 资源，并定义备份远程集群和需要恢复的任何其他 hub 集群资源所需的选项。查看以下示意图：



1.1.1.1. 备份的资源

集群备份和恢复 Operator 解决方案为所有 hub 集群资源（如受管集群、应用程序和策略）提供备份和恢复支持。您可以使用解决方案备份任何扩展基本 hub 集群安装的第三方资源。使用这个备份解决方案，您可以定义一个基于 cron 的备份调度，该调度在指定时间段内运行，并持续备份 hub 集群内容的最新版本。

当 hub 集群需要替换或处于灾难情况下，当 hub 集群停机时，可以部署新的 hub 集群并备份数据被移到新的 hub 集群中。

查看以下用于识别备份数据的集群备份和恢复过程的排序列表：

- 排除 **MultiClusterHub** 命名空间中的所有资源。这是为了避免备份链接到当前 hub 集群身份的安装资源，不应该备份。
- 备份 **.open-cluster-management.io** 和 **.hive.openshift.io** 后缀的 API 版本的所有资源。这些后缀表示所有 Red Hat Advanced Cluster Management 资源都已备份。

- 备份 中的所有资源： **argoproj.io**、**app.k8s.io**、**core.observatorium.io**、**hive.openshift.io**。这些资源在 **acm-resources-schedule** 备份中备份，但 **agent-install.openshift.io** API 组中的资源除外。这些资源在 **acm-managed-clusters-schedule** 备份中备份。

- 从以下 API 组中排除所有资源： **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**.

- 排除作为包含 API 组一部分的所有资源，但不需要或被所有者资源重新创建，这些资源也会被备份：
clustermanagementaddon,**observabilityaddon**,**applicationmanager**,**certpolicycontroller**,**iampolicycontroller**,**policycontroller**,**searchcollector**,**workmanager**, backupschedule
,**backupschedule**,**restore**,**clusterclaim.cluster.open-cluster-management.io**。

- 使用以下标签之一备份 secret 和 ConfigMap：**cluster.open-cluster-management.io/type**, **hive.openshift.io/secret-type**, **cluster.open-cluster-management.io/backup**。

- 对于您要备份的任何其他资源，使用 **cluster.open-cluster-management.io/backup** 标签，且不包含在前面提到的条件中。请参见以下示例：

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

注： hive.openshift.io.ClusterDeployment 资源使用的 Secret 需要备份，并仅在使用控制台创建集群时使用 **cluster.open-cluster-management.io/backup** 标签自动标注。如果使用 GitOps 部署 Hive 集群，您必须手动将 **cluster.open-cluster-management.io/backup** 标签添加到 **ClusterDeployment** 资源使用的 secret 中。带有 cluster. **open-cluster-management.io/backup: cluster-activation 标签的 secret** 和配置映射资源会在集群激活时恢复。

- 排除您不想备份的特定资源。请参阅以下从备份过程中排除 Velero 资源的示例：

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

1.1.1.2. 由 Red Hat Advanced Cluster Management 调度创建的备份文件

您可以使用 Red Hat Advanced Cluster Management 调度来备份 hub 资源，这些资源会根据资源类型或标签注解在单独的备份文件中分组。

BackupSchedule.cluster.open-cluster-management.io 资源会创建四个 **schedule.velero.io** 资源。这些 **schedule.velero.io** 资源生成备份文件，这些文件也称为资源。

要查看调度的备份文件列表，请运行以下命令： **oc get schedules -A | grep acm**。

调度的备份文件为 **backup.velero.io**。查看下表查看这些调度的备份文件的描述：

表 1.1. 调度的备份表

调度的备份	描述
凭证备份	存储以下内容： Hive 凭证、Red Hat Advanced Cluster Management、用户创建的凭证和 **ConfigMap**。此备份文件的名称是 **acm-credentials-schedule-<timestamp>**。
资源备份	包括一个备份，用于 Red Hat Advanced Cluster Management 资源，**acm-resources-schedule-<timestamp> 备份**，另一个用于通用资源 **acm-resources-generic-schedule-<timestamp>**。使用备份标签 **cluster.open-cluster-management.io/backup** 注解的任何资源都存储在 backup, **acm-resources-generic-schedule-backup** 下。例外是存储在 backup**acm-credentials-schedule-<timestamp** > 下的 **Secret** 或 **ConfigMap** 资源。
受管集群备份	仅包含激活受管集群连接到 hub 集群的资源，其中恢复备份。此备份文件的名称是 **acm-managed-clusters-schedule-<timestamp>**。

1.1.1.3. 在受管集群激活时恢复的资源

当您将 **cluster.open-cluster-management.io/backup** 标签添加到资源时，资源会在 **acm-resources-generic-schedule** 备份中自动备份。如果需要恢复任何资源，则必须将标签值设置为 **cluster-activation**，仅在受管集群移到新的 hub 集群后，并在恢复的资源中使用 **veleroManagedClustersBackupName:latest**。这样可确保资源不会被恢复，除非受管集群激活被调用。查看以下示例：

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

注： 对于任何受管集群命名空间或其中的任何资源，您必须在集群激活步骤中恢复一个。因此，如果您需要添加到受管集群命名空间中创建的备份资源，则对 **cluster.open-cluster-management.io/backup** 标签使用 **cluster-activation** 值。要了解恢复过程，请查看以下信息：

- 如果恢复命名空间，则 **managedcluster-import-controller** 会删除命名空间。
- 如果恢复 **managedCluster** 自定义资源，则 **cluster-manager-registration-controller** 会创建命名空间。

除了使用 **cluster.open-cluster-management.io/backup: cluster-activation** 标签并由 **acm-resources-generic-schedule** 备份存储的激活数据资源外，集群备份和恢复 Operator 还默认在激活集合中包括一些资源。以下资源由 **acm-managed-clusters-schedule** 备份备份：

- **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. 配置主动-被动 hub 集群

了解如何配置主动 - 被动 hub 集群配置，其中初始 hub 集群会备份数据，并在活跃集群不可用时控制受管集群。

1.1.2.1. 主动-被动配置

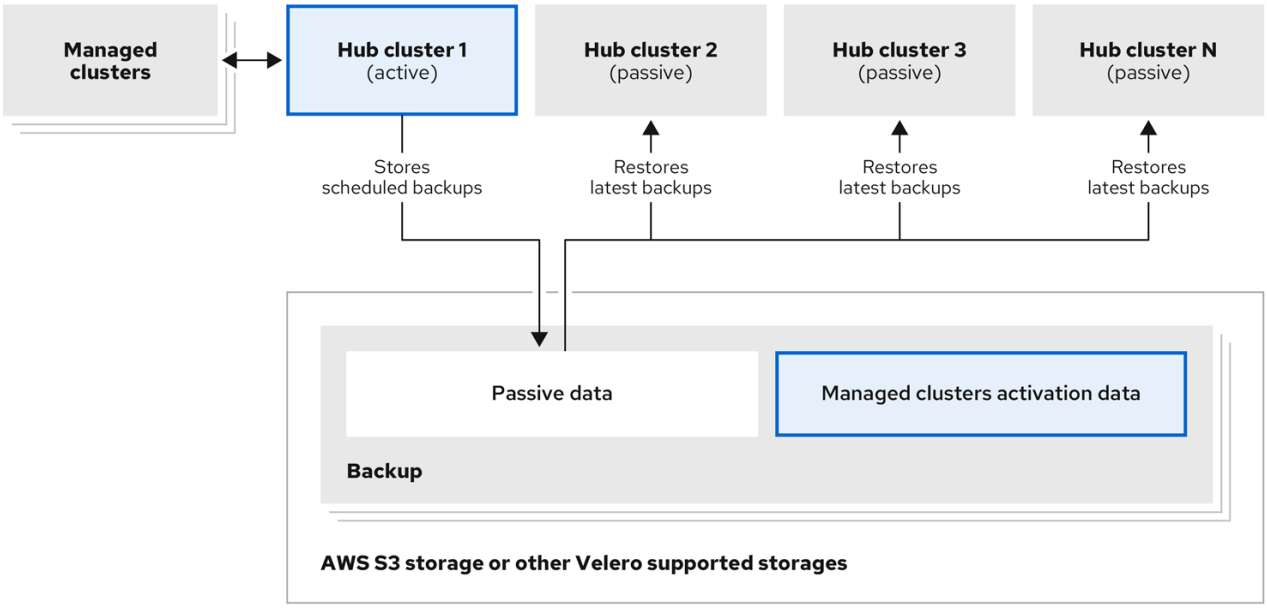
在主动-被动配置中，有一个主动 hub 集群和被动 hub 集群。一个活跃 hub 集群也被视为主 hub 集群，它使用 **BackupSchedule.cluster.open-cluster-management.io** 资源以定义的时间间隔管理集群并备份资源。

注： 要备份主 hub 集群数据，您不需要 **主动-被动配置**。您只需备份和存储 hub 集群数据。这样，如果出现问题或失败，您可以部署新的 hub 集群，并在这个新 hub 集群中恢复主 hub 集群数据。要缩短恢复主 hub 集群数据的时间，您可以使用 **主动 - 被动配置**，但这不是必须的。

被动 hub 集群会持续检索最新的备份并恢复被动数据。当有新的备份数据时，被动 hub 使用 **Restore.cluster.open-cluster-management.io** 资源从主 hub 集群恢复被动数据。这些 hub 集群处于备用状态，当主 hub 集群失败时会成为主 hub 集群。

主动和被动 hub 集群连接到相同的存储位置，主 hub 集群备份 被动 hub 集群的数据，以访问主 hub 集群。有关如何设置此自动恢复配置的详情，请参阅 *在检查备份时恢复被动资源*。

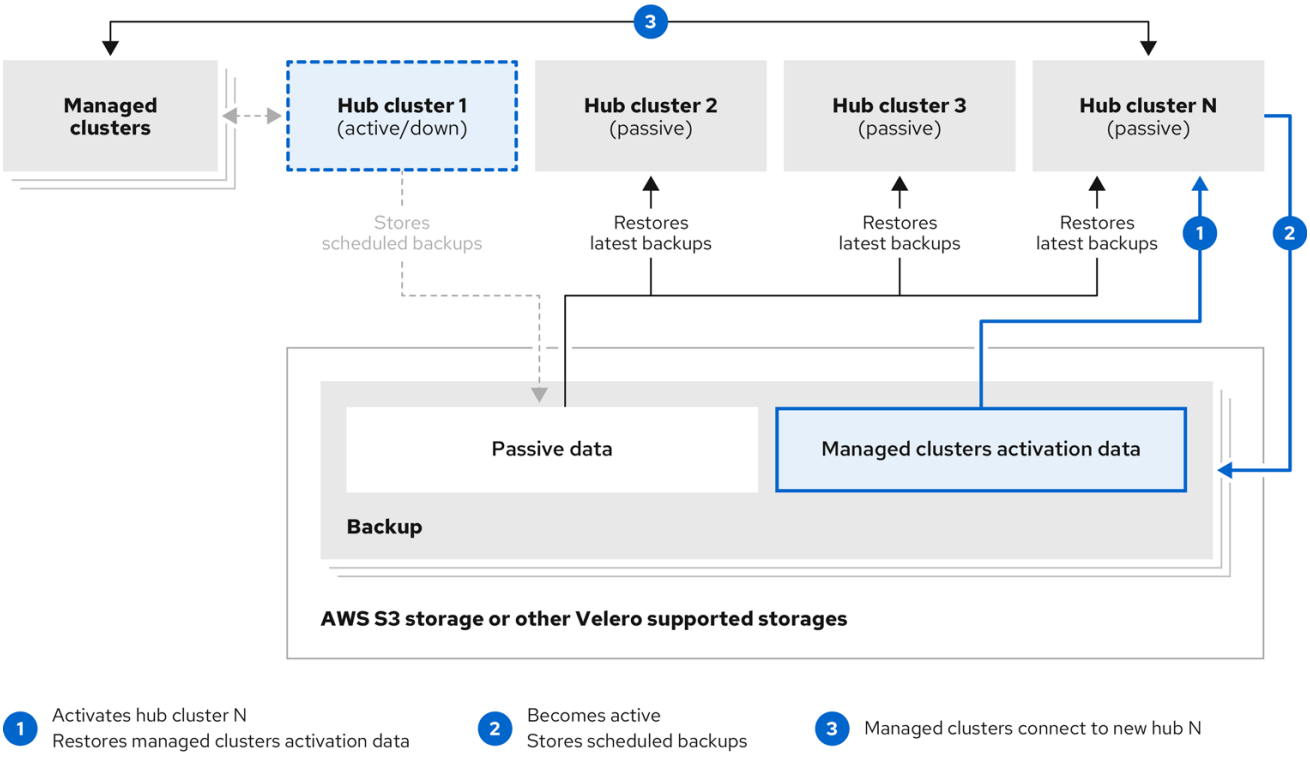
在以下图中，活跃 hub 集群会管理本地集群并定期备份 hub 集群数据：



被动 hub 集群恢复这个数据，但受管集群激活数据除外，后者将受管集群移到 passive hub 集群。被动 hub 集群可以持续恢复被动数据。被动 hub 集群可以将被动数据恢复为一次性操作。如需了解更多详细信息，请参阅 *恢复被动资源*。

1.1.2.2. 灾难恢复

当主 hub 集群失败时，管理员选择使用被动 hub 集群来接管受管集群。在以下镜像中，管理员决定将 *Hub 集群 N* 用作新的主 hub 集群：



hub 集群 N 恢复受管集群激活数据。此时，受管集群与 *Hub 集群 N* 连接。管理员通过在新的主 hub 集群（*Hub 集群 N*）上激活备份，方法是创建一个 **BackupSchedule.cluster.open-cluster-management.io** 资源，并将备份存储在与初始主 hub 集群相同的存储位置。

所有其他被动 hub 集群现在使用由新主 hub 集群创建的备份数据恢复被动数据。*Hub N* 现在是主 hub 集群，管理集群和备份数据。

备注：

- 上图中的进程 1 不是自动的，因为管理员必须决定主 hub 集群是否已失败且需要替换，或者 hub 集群和受管集群之间是否有网络通信错误。管理员还决定哪个被动 hub 集群成为主 hub 集群。当备份策略报告备份错误时，策略与 :aap: 作业集成可帮助您自动执行此步骤。

- 上图中的进程 2 是手动的。如果管理员没有从新的主 hub 集群创建备份，管理员会收到使用作为 cron 作业主动运行的备份通知的备份。

1.1.2.3. 其他资源

- 请参阅在检查备份时恢复被动资源。

- 请参阅恢复被动资源。

1.1.3. 安装备份和恢复 Operator

集群备份和恢复 Operator 不会被自动安装。继续阅读以了解如何安装和启用 Operator。

备注：

- 自定义资源定义的有效范围是集群范围内的，因此不能在同一集群中安装两个 OADP 或 Velero 版本。如果您有两个不同的版本，则一个版本会使用错误的自定义资源定义来运行。

- 如果您没有在 **MultiClusterHub** 资源中启用集群备份和恢复 Operator，则 OADP Operator 和 Velero 自定义资源定义仍然安装在 hub 集群中。**MultiClusterHub** 资源将 OADP 和 Velero 自定义资源定义与 OADP Operator 使用的版本进行协调，该定义会在启用集群备份和恢复 Operator 时安装。因此，您无法在 hub 集群上安装 OADP 或 Velero 的另一个版本，除非您的版本使用与启用备份和恢复 Operator 时安装的 OADP Operator 相同的自定义资源定义。

- 备份组件可以与组件命名空间中安装的 OADP Operator 一起工作。

- 在使用备份和恢复 Operator 之前，您必须设置 hub 集群。

重要：

如果手动安装 OADP operator，则 OADP operator 和 Velereo 的自定义资源定义版本必须完全匹配。如果这些版本与另一个版本不匹配，您可能会遇到问题。如果您之前已在与备份组件命名空间不同的命名空间中安装并使用了 OADP Operator，请卸载这个版本。

Velero 在 Red Hat Advanced Cluster Management for Kubernetes hub 集群中使用 OADP Operator 安装。它用于备份和恢复 Red Hat Advanced Cluster Management hub 集群资源。

有关 Velero 支持的存储供应商列表，请参阅关于安装 OADP。

要安装并启用 Operator，您必须完成以下任务：

- 为备份和恢复 Operator 设置 hub 集群

- 启用备份和恢复 Operator

1.1.3.1. 为备份和恢复 Operator 设置 hub 集群

要使用备份和恢复 Operator，您必须设置 hub 集群。

1.1.3.1.1. 创建存储位置 secret

要创建存储位置 secret，请完成以下步骤：

1. 完成为保存备份的云存储 创建默认 Secret 的步骤。
2. 在 OADP Operator 命名空间中创建 secret 资源，它位于备份组件命名空间中。

1.1.3.1.2. 启用备份 Operator

要为主动和被动 hub 集群启用备份 Operator，请完成以下步骤：

1. 在 Red Hat OpenShift Container Platform 集群中，安装 Red Hat Advanced Cluster Management for Kubernetes operator 版本 2.10.x。安装 Red Hat Advanced Cluster Management 时会自动创建 **MultiClusterHub** 资源，并显示以下状态： **Running**。
2. 手动安装集群备份和恢复 Operator。启用集群备份和恢复 Operator (**cluster-backup**)。通过将 **cluster-backup** 参数设置为 **true** 来编辑 **MultiClusterHub** 资源。这会在带有备份组件的同一命名空间中安装 OADP operator。
3. 在被动 hub 集群上运行恢复操作前，请完成以下步骤：
   a. 手动配置 hub 集群，并在活跃的 hub 集群以及与活跃 hub 集群相同的命名空间中安装所有 Operator。
   b. 验证是否安装了其他 Operator，如：Ansible Automation Platform、Red Hat OpenShift Container Platform GitOps 或证书管理器。通过验证，您可以确保配置了与初始 hub 集群相同的新 hub 集群。
   c. 在安装备份和恢复 Operator 以及在上一个 hub 集群上配置的任何 Operator 时，请确保 passive hub 集群使用与初始 hub 集群相同的命名空间名称。
4. 在被动 hub 集群上创建 **DataProtectionApplication 资源**。
5. 将被动 hub 集群连接到初始 hub 集群备份数据的相同存储位置。

1.1.3.1.3. 创建 DataProtectionApplication 资源

要为主动和被动 hub 集群创建 **DataProtectionApplication** 资源实例，请完成以下步骤：

1. 在 Red Hat OpenShift Container Platform 控制台中选择 **Operators** > **Installed Operators**。
2. 在 DataProtectionApplication 下点 **Create instance**。
3. 使用 {ocp-short）控制台或使用 **DataProtectionApplication** 示例中所述的 YAML 文件选择配置来创建 Velero 实例。
4. 将 **DataProtectionApplication** namespace 设置为 **open-cluster-management-backup**。
5. 为 **DataProtectionApplication** 资源正确设置规格(**spec:**)值。然后点**创建**。
   如果您打算使用默认的备份存储位置，请在 **backupStorageLocations** 部分中设置以下值 **default: true**。查看以下 **DataProtectionApplication** 资源示例：

```
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. 在断开连接的环境中启用备份和恢复组件

要在断开连接的环境中使用 Red Hat OpenShift Container Platform 启用备份和恢复组件，请完成以下步骤：

1. 使用 follwing 注解更新 **MultiClusterHub** 资源，以覆盖安装 OADP Operator 的源。在 **MultiClusterHub** 资源上启用 **cluster-backup** 组件前创建注解：

```
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** 是一个自定义名称，代表您定义并用来在断开连接的环境中访问 Red Hat OpenShift Operator 的 **CatalogSource** 资源的名称。运行以下命令来检索 **catalogsource** ：

```
oc get catalogsource -A
```

输出可能类似以下：

```
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. 启用备份和恢复 Operator

当第一次创建 **MultiClusterHub** 资源时，可以启用集群备份和恢复 Operator。**cluster-backup** 参数设为 **true**。启用 Operator 后，会安装 operator 资源。

如果已创建了 **MultiClusterHub** 资源，您可以通过编辑 **MultiClusterHub** 资源来安装或卸载集群备份 Operator。如果要卸载集群备份 Operator，将 **cluster-backup** 设置为 **false**。

启用备份和恢复 Operator 时，**MultiClusterHub** 资源可能类似以下 YAML 文件：

```
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. 其他资源

- 请参阅 Velero。
- 如需支持的 Velero 存储供应商列表，请参阅 OpenShift Container Platform 文档中的 AWS S3 兼容备份存储供应商。
- 了解有关 *DataProtectionApplication* 资源的更多信息。

1.1.4. 调度和恢复备份

完成以下步骤以调度和恢复备份：

1. 使用备份和恢复 Operator **backupschedule.cluster.open-cluster-management.io** 创建备份调度，并使用 **restore.cluster.open-cluster-management.io** 资源来恢复备份。
2. 运行以下命令来创建 **backupschedule.cluster.open-cluster-management.io** 资源：

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

   您的 **cluster_v1beta1_backupschedule.yaml** 资源可能类似以下文件：

   ```
   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) 每 2 小时创建备份

   (2) **可选：**在 120h 后删除调度的备份。如果没有指定，则使用最大 Velero 默认值 720h。

   查看 **backupschedule.cluster.open-cluster-management.io spec** 属性的描述：

   - **veleroSchedule** 是必需属性，定义用于调度备份的 cron 作业。
   - **veleroTtl** 是可选属性，定义调度的备份资源的过期时间。如果没有指定，则使用 Velero 设置的最大默认值，即 **720h**。
3. 检查 **backupschedule.cluster.open-cluster-management.io** 资源的状态，这会显示三个 **schedule.velero.io** 资源的定义。运行以下命令:

   ```
   oc get BackupSchedule -n open-cluster-management-backup
   ```
4. 提醒，恢复操作在不同的 hub 集群上运行，用于恢复场景。要启动恢复操作，请在要恢复备份的 hub 集群中创建一个 **restore.cluster.open-cluster-management.io** 资源。

注：当您在新的 hub 集群上恢复备份时，请确保关闭创建备份的以前的 hub 集群。如果正在运行，则旧的 hub 集群会在受管集群协调发现受管集群不再可用时立即重新导入受管集群。

您可以使用集群备份和恢复 Operator，**backupschedule.cluster.open-cluster-management.io** 和 **restore.cluster.open-cluster-management.io** 资源来创建备份或恢复资源。请参阅 *cluster-backup-operator* 示例。

5. 运行以下命令来创建 **restore.cluster.open-cluster-management.io** 资源：

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

您的资源可能类似以下文件：

```
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. 运行以下命令来查看 Velero **Restore** 资源：

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

7. 运行以下命令，查看 Red Hat Advanced Cluster Management **Restore** 事件：

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

有关 **Restore** YAML 资源的参数和示例的描述，请参阅 *恢复备份*部分。

1.1.4.1. 扩展备份数据

您可以通过在资源中添加 **cluster.open-cluster-management.io/backup** 标签来备份集群备份和恢复的第三方资源。标签的值可以是任意字符串，包括空字符串。使用一个可以帮助您识别要备份的组件的值。例如，如果组件是由 IDP 解决方案提供，请使用 **cluster.open-cluster-management.io/backup: idp** 标签。

注： 如果您希望在受管集群激活资源时恢复资源，请使用 **cluster.open-cluster-management.io/backup** 标签的 **cluster-activation** 值。恢复受管集群激活资源会导致受管集群活跃由 hub 集群（在启动恢复的位置）主动管理。

1.1.4.2. 调度集群备份

在创建 **backupschedule.cluster.open-cluster-management.io** 资源时，会激活备份调度。查看以下 **backupschedule.cluster.open-cluster-management.io** 示例：

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name: schedule-acm
```

```
  namespace: open-cluster-management-backup
spec:
  veleroSchedule: 0 */2 * * *
  veleroTtl: 120h
```

创建 **backupschedule.cluster.open-cluster-management.io** 资源后，运行以下命令来获取调度的集群备份的状态：

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

backupschedule.cluster.open-cluster-management.io 资源会创建六个 **schedule.velero.io** 资源，用于生成备份。运行以下命令查看调度的备份列表：

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

资源在组中单独备份，如下表所示：

表 1.2. 资源组表

资源	描述
凭证备份	存储 Hive 凭证、Red Hat Advanced Cluster Management 和用户创建的凭证和 ConfigMap 的备份文件。
资源备份	包括一个 Red Hat Advanced Cluster Management 资源的备份，另一个用于通用资源。这些资源使用 **cluster.open-cluster-management.io/backup** 标签。
受管集群备份	仅包含激活受管集群连接到 hub 集群的资源，其中恢复备份。

注 ：*资源备份*文件包含特定于受管集群的资源，但不包含将受管集群连接到 hub 集群的资源子集。连接受管集群的资源称为激活资源，并包含在受管集群备份中。当您只在新 hub 集群中为 *凭证*和 *资源*备份中恢复备份时，新的 hub 集群将所有使用 Hive API 创建的受管集群处于分离状态。只有在被动 hub 集群上恢复激活数据时，才会使用导入操作在主 hub 集群上导入的受管集群。受管集群仍然连接到创建备份文件的原始 hub 集群。

当恢复激活数据时，只有使用 Hive API 创建的受管集群才会与新的 hub 集群自动连接。所有其他受管集群都处于 *Pending* 状态。您需要手动将它们重新关联到新集群。

1.1.4.2.1. 避免备份冲突

如果 hub 集群从被动 hub 集群变为主 hub 集群，或者不同的受管集群在同一存储位置备份数据，则可能会出现备份冲突。

因此，最新的备份由 hub 集群生成，该集群不再设置为主 hub 集群。此 hub 集群仍然会生成备份，因为 **BackupSchedule.cluster.open-cluster-management.io** 资源仍然被启用。

请参阅以下列表以了解可能导致备份冲突的两个场景：

1. 主 hub 集群意外失败，这由以下条件导致：

- 从主 hub 集群到 Hub1 的通信会失败。
- Hub1 备份数据在辅助 hub 集群上恢复，名为 Hub2。
- 管理员会在 Hub2 上创建 **BackupSchedule.cluster.open-cluster-management.io** 资源，它是主 hub 集群，并为通用存储位置生成备份数据。
- Hub1 意外开始再次工作。
  因为 **BackupSchedule.cluster.open-cluster-management.io** 资源仍然在 Hub1 上被启用，所以 Hub1 将备份恢复到与 Hub2 相同的存储位置。现在，两个 hub 集群都在同一个存储位置写入备份数据。任何从此存储位置恢复最新的备份的 hub 集群都可以使用 Hub1 数据，而不是 Hub2 数据。

2. 管理员通过使 Hub2 成为主 hub 集群来测试灾难场景，这由以下条件导致：
   - Hub1 已停止。
   - Hub1 备份数据在 Hub2 上恢复。
   - 管理员会在 Hub2 上创建 **BackupSchedule.cluster.open-cluster-management.io** 资源，它是主 hub 集群，并为通用存储位置生成备份数据。
   - 灾难测试完成后，管理员将恢复到之前的状态，并再次使 Hub1 主 hub 集群。
   - Hub1 在 Hub2 仍然处于活跃状态时启动。
     由于 **BackupSchedule.cluster.open-cluster-management.io** 资源仍然在 Hub2 上被启用，所以它会在损坏备份数据的同一存储位置写入备份。任何从这个位置恢复最新的备份的 hub 集群都可以使用 Hub2 数据，而不是 Hub1 数据。在这种情况下，在启动 Hub1 前，首先停止 Hub2 或删除 **BackupSchedule.cluster.open-cluster-management.io** 资源，然后解决备份冲突问题。

为了避免和报告备份冲突，**BackupSchedule.cluster.open-cluster-management.io** 资源有一个 **BackupCollision** 状态。如果当前 hub 集群生成了存储位置的最新备份，控制器会定期检查。如果没有，不同的 hub 集群最近将备份数据写入存储位置，表示 hub 集群与不同的 hub 集群共存。

在这种情况下，当前 hub 集群 **BackupSchedule.cluster.open-cluster-management.io** 资源状态被设置为 **BackupCollision**，此资源创建的 **Schedule.velero.io** 资源将被删除以避免数据崩溃。**BackupCollision** 由备份策略报告。管理员在从无效 hub 集群中删除 **BackupSchedule.cluster.open-cluster-management.io** 资源并在有效的主 hub 集群上创建新的 **BackupSchedule.cluster.open-cluster-management.io** 资源前，管理员会验证哪个 hub 集群写入存储位置，以恢复备份。

运行以下命令检查是否有备份冲突：

```
oc get backupschedule -A
```

如果存在备份冲突，输出可能类似以下示例：

```
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. 其他资源

- 请参阅 *cluster-backup-operator* 示例。
- 有关 **Restore** YAML 资源的参数和示例的描述，请参阅 恢复备份 部分。
- 返回到 调度和恢复备份

1.1.5. 恢复备份

在进行一般的恢复时，运行备份的 hub 集群变得不可用，备份的数据需要移到一个新的 hub 集群。这可以通过在新的 hub 集群上运行集群恢复操作来完成。在这种情况下，恢复操作会在创建备份的不同 hub 集群中运行。

有些情况下，您要在收集备份的同一 hub 集群中恢复数据，以便恢复来自以前快照的数据。在这种情况下，恢复和备份操作都在同一 hub 集群中运行。

在 hub 集群中创建 **restore.cluster.open-cluster-management.io** 资源后，您可以运行以下命令来获取恢复操作的状态：

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

您还应能够验证是否已创建备份文件中包含的已备份资源。

注：restore.cluster.open-cluster-management.io 资源运行一次，除非您使用 **syncRestoreWithNewBackups** 选项并将其设置为 **true**，如 Restore passive resources 部分所述。如果要在恢复操作完成后再次运行相同的恢复操作，您必须创建一个具有相同 **spec** 选项的新 **restore.cluster.open-cluster-management.io** 资源。

restore 操作用于恢复备份操作创建的所有三种备份类型。您可以选择只安装特定类型的备份，如只有受管集群、只有用户凭证或只安装 hub 集群资源。

恢复定义以下三个必要的 **spec** 属性，其中为备份文件类型定义了恢复逻辑：

- **veleroManagedClustersBackupName** 用于定义受管集群激活资源的恢复选项。
- **veleroCredentialsBackupName** 用于为用户凭证定义 restore 选项。
- **veleroResourcesBackupName** 用于定义 hub 集群资源的 restore 选项（**Applications**、**Policy** 及其他 hub 集群资源，如受管集群被动数据）。
  前面提到的属性的有效选项有以下值：
  - **latest** - 此属性恢复此类型的备份文件。
  - **skip** - 此属性不会尝试使用当前恢复操作恢复这种类型的备份。
  - **<backup_name>** - 此属性按名称恢复指向它的指定的备份。

由 **restore.cluster.open-cluster-management.io** 创建的 **restore.velero.io** 资源的名称遵循以下模版规则 **<restore.cluster.open-cluster-management.io name>-<velero-backup-resource-name>**.查看以下描述：

- **restore.cluster.open-cluster-management.io 名称** 是当前 **restore.cluster.open-cluster-management.io** 资源的名称，该资源用于启动恢复。

- **velero-backup-resource-name** 是 Velero 备份文件的名称，用于恢复数据。例如，名为 **restore-acm** 的 **restore.cluster.open-cluster-management.io** 资源创建 **restore.velero.io** 恢复资源。查看以下格式示例：
  - **restore-acm-acm-managed-clusters-schedule-20210902205438** 可用于恢复受管集群激活数据备份。在本例中，用于恢复资源的 **backup.velero.io** 备份名称为 **acm-managed-clusters-schedule-20210902205438**。
  - **restore-acm-acm-credentials-schedule-20210902206789** 用于恢复凭据备份。在本例中，用于恢复资源的 **backup.velero.io** 备份名称为 **acm-managed-clusters-schedule-20210902206789**。
  - **restore-acm-acm-resources-schedule-20210902201234** 用于恢复应用程序、策略和其他 hub 集群资源，如受管集群被动数据备份。在这个示例中，用于恢复资源的 **backup.velero.io** 备份名称为 **acm-managed-clusters-schedule-20210902201234**。

注：如果 **skip** 用于备份类型，则不会创建 **restore.velero.io**。

查看以下集群 **Restore** 资源的 YAML 示例。在这个示例中，使用最新可用的备份文件恢复所有三种备份文件：

```
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
```

Note:当来自受管集群备份的 **acm-managed-clusters** 备份被恢复到另外一个 hub 集群时，只有 Hive API 创建的受管集群会自动连接到新的 hub 集群。所有其他受管集群都处于 **Pending Import** 状态，且必须重新导入到新的 hub 集群中。如需更多信息，请参阅恢复导入的受管集群。

1.1.5.1. 将数据恢复回初始主 hub

当您需要恢复集群中的备份数据时，请创建一个新集群。在 hub 集群恢复操作中，如果这些资源不是要恢复的备份数据的一部分，您可以配置 hub 集群备份恢复来清理现有资源。恢复会清理之前备份创建的资源，但不会清理用户资源。因此，在此 hub 集群上用户创建的资源不会被清理，因此此 hub 集群上的数据不会反映恢复的资源可用数据。

灾难恢复测试是一个示例，您可以使用现有的 hub 集群。在恢复测试中，您只测试 hub 备份场景。在这种情况下，初始主 hub 集群不会创建新资源。相反，备份数据已从主 hub 集群临时改为 passive hub 集群。

1.1.5.2. 准备新的 hub 集群

在新 hub 集群上运行恢复操作前，您需要手动配置 hub 集群，并在初始 hub 集群上安装相同的 Operator。您必须在与初始 hub 集群相同的命名空间中安装 Red Hat Advanced Cluster Management Operator，创建 *DataProtectionApplication* 资源，然后连接到之前备份了数据的同一存储位置。

对 Red Hat Advanced Cluster Management Operator 创建的 **MultiClusterHub** 资源使用与初始 hub 集群上的配置相同，包括对 **MultiClusterEngine** 资源的任何更改。

例如，如果初始 hub 集群安装了任何其他 Operator，如 Ansible Automation Platform Red Hat

例如，如果初始 hub 集群安装了任何其他 Operator，如 Ansible Automation Platform、Red Hat OpenShift GitOps、**cert-manager**，则必须在运行恢复操作前安装它们。这样可确保配置新的 hub 集群与初始 hub 集群相同。

1.1.5.3. 在恢复后清理 hub 集群

如果使用当前恢复的备份更改了，Velero 会更新现有资源。Velero 不会清理 delta 资源，这些资源是由以前的恢复创建的资源，而不是当前恢复的备份的一部分。这限制了在新 hub 集群上恢复 hub 集群数据时可以使用的情况。除非恢复只应用一次，否则您无法可靠地使用新的 hub 集群作为被动配置。hub 集群中的数据没有反映恢复的资源可用的数据。

为解决这个限制，当创建 **Restore.cluster.open-cluster-management.io** 资源时，备份 Operator 会运行一个清理 hub 集群的 post 恢复操作。该操作会删除之前不是由 Red Hat Advanced Cluster Management 恢复（不是当前恢复的备份的一部分）创建的资源。

后恢复清理使用 **cleanupBeforeRestore** 属性来识别要清理的对象子集。您可以使用以下选项进行后恢复清理：

- **None**: 不需要清理，只开始 Velero 恢复。在新 hub 集群中使用 **None**。
- **CleanupRestored** ：清理之前由以前 Red Hat Advanced Cluster Management 恢复创建的所有资源，这些资源不是当前恢复的备份的一部分。
- **CleanupAll** ：清理 hub 集群上可能成为 Red Hat Advanced Cluster Management 备份的一部分的所有资源，即使它们没有因为恢复操作而创建。这在恢复操作启动前在 hub 集群上创建额外内容时使用。
  最佳实践：避免使用 **CleanupAll** 选项。仅将它用作最后的手段，且需要非常小心。除了之前恢复的备份创建的资源外，**CleanupAll** 还会清理用户创建的 hub 集群上的资源。反之，使用 **CleanupRestored** 选项来防止在 hub 集群指定为灾难场景的被动候选时更新 hub 集群内容。使用干净的 hub 集群作为被动集群。

备注：

- 如果恢复的备份没有资源，Velero 会为 velero 恢复资源设置状态 **PartiallyFailed**。这意味着，如果任何创建的 **restore.velero.io** 资源没有恢复任何资源，则 **restore.cluster.open-cluster-management.io** 资源可能会处于 **PartiallyFailed** 状态。
- **restore.cluster.open-cluster-management.io** 资源会运行一次，除非您使用 **syncRestoreWithNewBackups:true** 来在新的备份可用时恢复被动数据。在这种情况下，请按照使用同步示例的恢复被动操作。请参阅在检查备份时恢复被动资源。完成恢复操作后，您想要在同一 hub 集群上运行另一个恢复操作，您必须创建一个新的 **restore.cluster.open-cluster-management.io** 资源。
- 虽然您可以创建多个 **restore.cluster.open-cluster-management.io** 资源，但在任何时间点上只能有一个。

1.1.5.4. 在检查备份时恢复被动资源

使用 **restore-passive-sync** 示例恢复被动数据，同时继续检查新备份是否可用并自动恢复它们。要自动恢复新的备份，您必须将 **syncRestoreWithNewBackups** 参数设置为 **true**。您还必须仅恢复最新的被动数据。您可以在本节末尾找到示例示例。

将 **VeleroResourcesBackupName** 和 **VeleroCredentialsBackupName** 参数设置为 **latest**，**VeleroManagedClustersBackupName** 参数为 **skip**。当将 **VeleroManagedClustersBackupName** 设置为 **latest** 后，受管集群会在新的 hub 集群中激活，现在是主 hub 集群。

当激活的受管集群变为主 hub 集群时，恢复资源被设置为 **Finished**，并且 **syncRestoreWithNewBackups** 会被忽略，即使设置为 **true**。

默认情况下，当 **syncRestoreWithNewBackups** 设为 **true** 时，控制程序会每 30 分钟检查新的备份。如果找到新的备份，它会恢复备份的资源。您可以通过更新 **restoreSyncInterval** 参数来更改检查的持续时间。

例如，请查看以下资源每 10 分钟检查备份：

```
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. 恢复被动资源

使用 **restore-acm-passive** 示例在被动配置中恢复 hub 集群资源。被动数据是备份数据，如 secret、ConfigMap、应用程序、策略以及所有受管集群自定义资源，它不在受管集群和 hub 集群之间激活连接。备份资源通过凭证备份和恢复资源在 hub 集群上恢复。

请参见以下示例：

```
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. 恢复激活资源

在恢复被动 hub 集群上的激活数据前，请关闭创建备份的以前的 hub 集群。如果主 hub 集群仍在运行，它会尝试根据在这个 hub 集群上运行的协调流程，尝试使用不再可用的受管集群。

当您希望 hub 集群管理集群时，请使用 **restore-acm-passive-activate** 示例。在这种情况下，假设其它数据已在使用被动资源的 hub 集群上恢复。

```
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
```

根据您如何恢复被动资源，您有一些选项来恢复激活资源：

- 如果您使用 **restore-acm-passive-sync cluster.open-cluster-management.io** 资源，如 *Restore passive resources while checking for backups to restore passive data* 部分所述，将 **veleroManagedClustersBackupName** 值更新为 **latest**。因此，受管集群资源和 **restore-acm-passive-sync** 资源会被恢复。
- 如果您将被动资源恢复为一个时间操作，或者还没有恢复任何资源，选择恢复所有资源，如 *恢复所有资源*部分所述。

1.1.5.7. 恢复受管集群激活数据

当使用 **cluster.open-cluster-management.io/backup: cluster-activation** 标签时，受管集群激活数据或其他激活数据资源由受管集群备份和 resource-generic 备份存储。当在新的 hub 集群上恢复激活数据时，受管集群由运行恢复的 hub 集群主动管理。请参阅 *调度和恢复备份* 以了解如何使用 Operator。

1.1.5.8. 恢复所有资源

如果要一次恢复所有数据，请使用 **restore-acm** 示例，并使 hub 集群在一个步骤中管理受管集群。在 hub 集群中创建 **restore.cluster.open-cluster-management.io** 资源后，运行以下命令获取恢复操作的状态：

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

您的示例可能类似以下资源：

```
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
```

在 hub 集群中，验证是否已创建备份文件中包含的备份资源。

1.1.5.9. 恢复导入的受管集群

只有使用 Hive API 与主 hub 集群连接的受管集群会自动连接到新的 hub 集群，其中恢复激活数据。这些集群由 Hive API 在主 hub 集群上创建，使用 **Clusters** 选项卡中的 **Create cluster** 按钮或通过 CLI 创建。当激活数据被恢复时，使用 **Import cluster** 按钮连接到初始 hub 集群的受管集群显示为 **Pending Import**，需要在新的 hub 集群中重新导入。

Hive 受管集群可以与新的 hub 集群连接，因为 Hive 将受管集群 **kubeconfig** 存储在 hub 集群上的受管集群命名空间中。这在新的 hub 集群上备份和恢复。然后，导入控制器使用恢复的配置更新受管集群上的 bootstrap **kubeconfig**，该配置仅适用于使用 Hive API 创建的受管集群。导入的集群不可用。

要在新 hub 集群上重新连接导入的集群，请在启动恢复操作后手动创建 **auto-import-secret** 资源。如需了解更多详细信息，请参阅 *使用自动导入 secret 导入集群*。

在受管集群命名空间中创建 **auto-import-secret** 资源，每个集群处于 **Pending Import** 状态。使用具有足够权限的 **kubeconfig** 或令牌，以便导入组件在新 hub 集群上启动自动导入。您必须使用令牌连接到受管集群，为每个受管集群具有访问权限。令牌必须具有 **klusterlet** 角色绑定或具有相同权限的角色。

1.1.5.10. 使用其他恢复示例

查看以下 Restore 部分，以查看恢复不同类型的备份文件的 YAML 示例。

- 恢复所有三种备份资源：

```
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
```

- 仅恢复受管集群资源：

```
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
```

- 使用 **acm-managed-clusters-schedule-20210902205438** 备份只为受管集群恢复资源：

```
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
```

备注:

  - **restore.cluster.open-cluster-management.io** 资源运行一次。恢复操作完成后，您可以选择在同一 hub 集群中运行另一个恢复操作。您必须创建新的 **restore.cluster.open-cluster-management.io** 资源才能运行新的恢复操作。

  - 您可以创建多个 **restore.cluster.open-cluster-management.io**，但在任何时候都只能运行一个。

1.1.5.11. 查看恢复事件

使用以下命令获取有关恢复事件的信息：

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

您的事件列表可能类似以下示例：

```
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. 其他资源

- 请参阅 *DataProtectionApplication*。
- 请参阅使用自动导入 secret 导入集群。
- 请参阅调度和恢复备份。

1.1.6. 使用受管服务帐户自动连接集群

备份控制器使用 Managed Service Account 组件自动将导入的集群连接到新的 hub 集群。Managed Service Account 创建一个令牌，为每个受管集群命名空间中的每个导入的集群备份令牌。该令牌使用 **klusterlet-bootstrap-kubeconfig ClusterRole** 绑定，允许自动导入操作使用令牌。**klusterlet-bootstrap-kubeconfig ClusterRole** 只能获取或更新 **bootstrap-hub-kubeconfig** secret。要了解更多有关受管服务帐户组件的信息，请参阅 *什么是受管服务帐户？*

当新 hub 集群上恢复激活数据时，恢复控制器会运行 post 恢复操作，并查找处于 **Pending Import** 状态的所有受管集群。如果找到了 Managed Service Account 生成的有效令牌，控制器会使用令牌创建一个 **auto-import-secret**。因此，导入组件会尝试重新连接受管集群。如果可以访问集群，则操作可以成功。

1.1.6.1. 启用自动导入

默认情况下，禁用使用 Managed Service Account 组件的自动导入功能。要启用自动导入功能，请完成以下步骤：

1. 通过在 **MultiClusterEngine** 资源中将 **managedserviceaccount enabled** 参数设置为 **true** 来启用 Managed Service Account 组件。请参见以下示例：

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

2. 通过将 **useManagedServiceAccount** 参数设置为 **true**，为 **BackupSchedule.cluster.open-cluster-management.io** 资源启用自动导入功能。请参见以下示例：

```
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
```

默认令牌有效期持续时间被设置为 **veleroTtl** 值的两倍，以提高令牌对整个生命周期存储令牌的所有备份的几率。在某些情况下，您可能需要通过为可选的 **managedServiceAccountTTL** 属性设置值来控制令牌的有效时长。

如果您需要为生成的令牌更新默认令牌过期时间，请谨慎使用 **managedServiceAccountTTL**。从默认值更改令牌过期时间可能会导致生成带有令牌集的备份，在备份生命周期中过期。因此，导入功能不适用于受管集群。

重要 ：除非需要控制令牌的有效时长，否则不要使用 **managedServiceAccountTTL**。

有关使用 **managedServiceAccountTTL** 属性的示例：

```
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
```

启用自动导入功能后，备份组件通过创建以下内容开始处理导入的受管集群：

- 名为 **managed-serviceaccount** 的 **ManagedServiceAddon**。
- 名为 **auto-import-account** 的 **ManagedServiceAccount**。

- 每个 **ManagedServiceAccount** 的 **ManifestWork**，用于为受管集群上的 **ManagedServiceAccount** 令牌设置 **klusterlet-bootstrap-kubeconfig RoleBinding**。

只有在创建 Managed Service Account 时可以访问受管集群时，才会创建令牌，否则在受管集群可用后它会被创建。

1.1.6.2. 自动导入注意事项

以下场景可能会阻止受管集群在移至新的 hub 集群时被自动导入：

- 在没有 **ManagedServiceAccount** 令牌的情况下运行 hub 备份时，例如在受管集群无法访问时创建 **ManagedServiceAccount** 资源时，备份不包含自动导入受管集群的令牌。
- 如果 **auto-import-account** secret 令牌有效并且已备份，但在备份的可用令牌已过期时运行恢复操作，则自动导入操作会失败。**restore.cluster.open-cluster-management.io** 资源会为每个受管集群报告无效的令牌问题。
- 因此在恢复时创建的 **auto-import-secret** 使用 **ManagedServiceAccount** 令牌连接到受管集群，受管集群还必须提供 kube **apiserver** 信息。**apiserver** 必须在 **ManagedCluster** 资源中设置。请参见以下示例：

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: managed-cluster-name
spec:
  hubAcceptsClient: true
  leaseDurationSeconds: 60
  managedClusterClientConfigs:
    url: <apiserver>
```

  当在 hub 集群中导入集群时，只会在 OpenShift Container Platform 集群中自动设置 **apiserver**。您必须在其他类型的受管集群中手动设置 **apiserver**，如 EKS 集群，否则自动导入功能会忽略集群。因此，当您将其移到恢复 hub 集群时，集群会处于 **Pending Import** 状态。

- 如果备份调度在 **ManagedServiceAccount** secret 上设置了备份标签前运行，则 **ManagedServiceAccount** secret 可能不包含在备份中。**ManagedServiceAccount** secret 在创建时没有设置集群 **open-cluster-management.io/backup** 标签。因此，备份控制器定期在受管集群命名空间下搜索 **ManagedServiceAccount** secret，并在未找到时添加备份标签。

1.1.6.3. 禁用自动导入

您可以通过在 **BackupSchedule** 资源中将 **useManagedServiceAccount** 参数设置为 **false** 来禁用自动导入集群功能。请参见以下示例：

```
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
```

默认值为 **false**。将值设为 **false** 后，备份 Operator 会删除所有创建的资源，包括 **ManagedServiceAddon**、**ManagedServiceAccount** 和 **ManifestWork**。删除资源会删除 hub 集群和受管集群中的自动导入令牌。

1.1.6.4. 其他资源

- 请参阅什么是受管服务帐户？ 以了解更多有关受管服务帐户组件的信息。
- 使用受管服务帐户返回到自动连接集群。

1.1.7. 验证备份或恢复配置

当您在 **MultiClusterHub** 资源中将 **cluster-backup** 选项设置为 **true** 时，多集群引擎 operator 安装集群备份和恢复名为 **cluster-backup-chart** 的 operator Helm chart。然后，这个 chart 安装 **backup-restore-enabled** 和 **backup-restore-auto-import** 策略。使用这些策略查看有关备份和恢复组件的问题的信息。

注： hub 集群通过使用 **local-cluster** 受管集群自行导入和管理。如果您在 **MultiClusterHub** 资源中设置 **disableHubSelfManagement=true** 来禁用此功能，则不会将 **backup-restore-enabled** 策略放在 hub 集群中，策略模板不会生成任何报告。

如果集群 hub 由全局 hub 集群管理，或者在受管集群中安装它，则禁用 **disableHubSelfManagement=true** 设置。在本例中，您可以启用 **backup-restore-enabled** 策略。通过在代表受管 hub 集群的 **ManagedCluster** 资源中设置 **is-hub=true** 标签来启用策略。

backup-restore-enabled 策略包括一组用于检查以下限制的模板：

- **OADP 频道验证**
  - 当您在 **MultiClusterHub** 中启用备份组件时，集群备份和恢复 Operator Helm Chart 将安装 OADP operator。**OADP-channel** 模板检查安装的 Red Hat OADP Operator 版本是否与 Red Hat Advanced Cluster Management 集群备份和恢复 Operator 设置的版本匹配。
  - 如果在 hub 集群上找到一个安装的 Red Hat OADP Operator，则模板会显示违反情况，但 Red Hat OADP Operator 与 Red Hat Advanced Cluster Management 集群备份和恢复 Operator Helm chart 安装的版本不匹配。违反情况显示集群中 OADP Operator 的错误版本。由于 OADP Operator 和 Velero 自定义资源定义(CRD) **是集群范围的**，因此不能在同一集群中安装多个版本。相反，您必须只安装正确的版本。
  - 在以下示例中，备份和恢复 Operator 可使用错误的 CRD 运行会导致行为错误：
    - Red Hat Advanced Cluster Management 有多个安装的 OADP 版本。
    - 如果 **MultiClusterHub** 安装的 OADP 版本已被卸载，您可以手动安装不同的版本。
- **Pod 验证**
  以下模板检查备份组件和依赖项的 pod 状态：
  - **acm-backup-pod-running** 模板检查备份和恢复 operator pod 是否在运行。
  - **oadp-pod-running** 模板检查 OADP operator pod 是否在运行。
  - **velero-pod-running** 模板检查 Velero pod 是否在运行。
- **数据保护应用程序验证**
  - **data-protection-application-available** 模板检查是否创建了 **DataProtectionApplication.oadp.openshift.io** 资源。这个 OADP 资源设置 Velero 配置

DataProtectionApplication.oadp.openshift.io 资源。这个 OADP 资源设置 Velero 配置。

- **备份存储验证**
  - **backup-storage-location-available** 模板检查 **BackupStorageLocation.velero.io** 资源是否已创建以及状态值是否为 **Available**。这意味着与备份存储的连接有效。
- BackupSchedule **冲突验证**
  - 如果当前 hub 集群上存在 **BackupSchedule.cluster.open-cluster-management.io**，则 **acm-backup-clusters-collision-report** 模板会验证状态不是 **BackupCollision**。这会在将备份数据写入存储位置时，验证当前 hub 集群与其它 hub 集群不冲突。
    有关 **BackupCollision** 的定义，请参阅 避免备份冲突。
- BackupSchedule **和恢复状态验证**
  - **acm-backup-phase-validation** 模板检查当前集群中是否存在 **BackupSchedule.cluster.open-cluster-management.io**，则检查状态为 **Failed** 或 **Empty** 状态。这样可确保如果此集群是主 hub 集群，并正在生成备份，则 **BackupSchedule.cluster.open-cluster-management.io** 状态是健康。
  - 如果当前集群中存在 **Restore.cluster.open-cluster-management.io**，则相同的模板会检查当前集群中没有处于 **Failed** 或 **Empty** 状态的状态。这样可确保如果这个集群是二级 hub 集群，且被恢复备份，**Restore.cluster.open-cluster-management.io** 状态是健康。
- **备份存在验证**
  - **acm-managed-clusters-schedule-backups-available** 模板检查 **Backup.velero.io** 资源是否位于 **BackupStorageLocation.velero.io** 指定的位置上，以及备份是否由 **BackupSchedule.cluster.open-cluster-management.io** 资源创建。这验证了备份已至少运行一次，使用备份和恢复 Operator。
- **备份完成**
  - **acm-backup-in-progress-report** 模板检查 **Backup.velero.io** 资源是否处于 **InProgress** 状态。这个验证会被添加，因为带有大量资源，velero pod 会作为备份运行重启，备份会停留在不继续完成状态。在正常备份过程中，备份资源会在某一时间点进行，但不会被卡住并在完成运行。正常情况下，在调度运行时报告 **acm-backup-in-progress-report** 模板会在调度运行时报告警告并备份正在进行。
- **主动作为 cron 作业运行的备份**
  - **BackupSchedule.cluster.open-cluster-management.io** 主动运行并在存储位置保存新的备份。此验证通过 **backup-schedule-cron-enabled** 策略模板来完成。模板检查是否有带有 **velero.io/schedule-name: acm-validation-policy-schedule** 标签的 **Backup.velero.io**。
  - **acm-validation-policy-schedule** 备份设置为在为备份 cron 调度设定时间后过期。如果没有创建备份的 cron 作业，旧的 **acm-validation-policy-schedule** 备份将被删除，因为它过期且没有创建新的备份。因此，如果在任何时间点上没有 **acm-validation-policy-schedule backups**，这代表没有活跃的 cron 作业生成备份。
  - 此策略旨在帮助在 hub 集群活跃并生成或恢复备份时通知 hub 集群管理员。

backup-restore-auto-import 策略包括一组检查以下限制的模板：

- **自动导入 secret 验证**
  - **auto-import-account-secret** 模板检查在 **local-cluster** 以外的受管集群命名空间中是否创建了 **ManagedServiceAccount** secret。备份控制器定期扫描导入的受管集群。发现受管集群

后，备份控制器会在受管集群命名空间中创建 **ManagedServiceAccount** 资源。此过程在受管集群中启动令牌创建。但是，如果此操作时无法访问受管集群，则 **ManagedServiceAccount** 无法创建令牌。例如，如果受管集群是休眠状态，则无法创建令牌。因此，如果在这个期间执行 hub 备份，则备份缺少一个令牌来自动导入受管集群。

- **自动导入备份标签验证**
  - **auto-import-backup-label** 模板验证 **local-cluster** 以外的受管集群命名空间中的 **ManagedServiceAccount** secret 是否存在。如果模板找到 **ManagedServiceAccount** secret，则模板在 secret 上强制执行 **cluster.open-cluster-management.io/backup** 标签。该标签对于在 Red Hat Advanced Cluster Management 备份中包含 **ManagedServiceAccount** secret 至关重要。

1.1.7.1. 使用服务器端加密保护数据

服务器端加密是应用程序或服务的数据加密，在存储位置接收数据。在传输过程中，备份机制本身不会加密数据（因为它会前往备份存储位置），或其他人（同时存储在备份存储位置的磁盘上）。它依赖于对象和快照系统中的原生机制。

最佳实践 ：使用可用的备份存储服务器端加密数据。备份包含资源，如在 hub 集群外存储需要加密的凭证和配置文件。

您可以使用 **serverSideEncryption** 和 **kmsKeyId** 参数为存储在 Amazon S3 中的备份启用加密。如需了解更多详细信息，请参阅 *备份存储位置 YAML*。以下示例指定在设置 **DataProtectionApplication** 资源时的 AWS KMS 密钥 ID：

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

请参阅 *Velero 支持的存储供应商*，以找出其它存储供应商的所有可配置参数。

1.1.7.2. 其他资源

- 请参阅备份存储位置 YAML。
- 请参阅 Velero 支持的存储供应商。
- 返回到验证您的备份或恢复配置。

1.1.8. 备份和恢复高级配置

您可以通过查看以下部分来进一步配置备份和恢复：

1.1.8.1. 资源请求和限值自定义

最初安装 Velero 时，Velero pod 会被设置为默认 CPU 和内存限值，如下例所示：

```
resources:
 limits:
   cpu: "1"
```

```
    memory: 256Mi
  requests:
    cpu: 500m
    memory: 128Mi
```

以上示例中的限制在某些情况下可以正常工作，但可能会在集群备份大量资源时进行更新。例如，当备份在管理 2000 集群的 hub 集群上运行时，Velero pod 会因为内存不足错误(OOM)崩溃。对于这种情况，以下配置允许备份完成：

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

要更新 Velero pod 资源的限制和请求，您需要更新 **DataProtectionApplication** 资源，并为 Velero pod 插入 **resourceAllocation** 模板。查看以下示例：

```
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. 其他资源

- 请参阅 Red Hat OpenShift Container Platform 文档中的 默认 Velero 云供应商插件 部分，以了解更多有关 **DataProtectionApplication** 参数的信息。
- 如需有关备份和恢复 CPU 和内存要求的更多详情，请参阅 OpenShift Container Platform 文档中的配置主题的 CPU 和内存要求。

1.2. VOLSYNC 持久性卷复制服务

VolSync 是一种 Kubernetes 操作器，支持异步复制集群中的持久性卷，或者在集群中使用存储类型不兼容进行复制的集群间复制。它使用容器存储接口(CSI)来克服兼容性限制。在您的环境中部署 VolSync Operator 后，您可以使用它来创建和维护持久数据的副本。VolSync 只能在版本 4.13 或更高版本的 Red Hat OpenShift Container Platform 集群上复制持久性卷声明。

重要： VolSync 只支持复制带有 **volumeMode** 的 **Filesystem** 的持久性卷声明。如果您没有选择 **volumeMode**，则默认为 **Filesystem**。

- 使用 VolSync 复制持久性卷
  - 在受管集群上安装 VolSync
  - 配置 Rsync-TLS 复制
  - 配置 Rsync 复制
  - 配置剩余的备份
  - 配置 Rclone 复制
- 将复制镜像转换为可用的持久性卷声明
- 调度同步

1.2.1. 使用 VolSync 复制持久性卷

您可以使用三种支持的方法来复制带有 VolSync 的持久性卷，这取决于您拥有的同步位置数量：rsync, rsync-tls, restic 或 Rclone。

1.2.1.1. 先决条件

在集群上安装 VolSync 前，您必须满足以下要求：

- 配置了运行 Red Hat Advanced Cluster Management 版本 2.10 或更高版本的 hub 集群的 Red Hat OpenShift Container Platform 环境
- 至少配置两个由同一 Red Hat Advanced Cluster Management hub 集群管理的集群
- 使用 VolSync 配置的集群之间的网络连接。如果集群不在同一网络中，您可以配置 Submariner multicluster networking 和 service discovery，并使用 **ServiceType** 的 **ClusterIP** 值来联网集群，或使用带有 **LoadBalancer** 值的 **ServiceType** 的负载均衡器。
- 您用于源持久性卷的存储驱动程序必须与 CSI 兼容，并能够支持快照。

1.2.1.2. 在受管集群上安装 VolSync

要启用 VolSync 将一个集群上的持久性卷声明复制到另一个集群的持久性卷声明，您必须在源和目标受管集群中安装 VolSync。

VolSync 不创建自己的命名空间，因此它与其他 OpenShift Container Platform all-namespace operator 相同的命名空间中。对 VolSync 的 Operator 设置所做的任何更改也会影响同一命名空间中的其他 Operator，例如，如果您更改为手动批准频道更新。

您可以使用两种方式之一在环境中的两个集群中安装 VolSync。您可以为 hub 集群中的每个受管集群添加标签，也可以手动创建并应用 **ManagedClusterAddOn**，如以下部分所述：

1.2.1.2.1. 使用标签安装 VolSync

通过添加标签在受管集群中安装 VolSync。

- 从 Red Hat Advanced Cluster Management 控制台完成以下步骤：

1. 从 hub 集群控制台的 **Clusters** 页面中选择其中一个受管集群来查看其详情。
2. 在 **Labels** 字段中，添加以下标签：

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

   VolSync 服务 pod 已安装在受管集群上。
3. 为其他受管集群添加相同的标签。
4. 在每个受管集群中运行以下命令，以确认已安装了 VolSync Operator：

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

   安装 VolSync 时，会列出该 Operator。

- 使用命令行界面完成以下步骤：
  1. 在 hub 集群中启动一个命令行会话。
  2. 输入以下命令为第一个集群添加标签：

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

     将 **managed-cluster-1** 替换为其中一个受管集群的名称。
  3. 输入以下命令在第二个集群中添加标签：

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

     将 **managed-cluster-2** 替换为其他受管集群的名称。

     应该在每个对应受管集群的命名空间中自动创建一个 **ManagedClusterAddOn** 资源。

1.2.1.2.2. 使用 ManagedClusterAddOn 安装 VolSync

要通过手动添加 **ManagedClusterAddOn** 在受管集群中安装 VolSync，请完成以下步骤：

1. 在 hub 集群中，创建一个名为 **volsync-mcao.yaml** 的 YAML 文件，其中包含类似以下示例的内容：

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

   将 **managed-cluster-1-namespace** 替换为其中一个受管集群的命名空间。此命名空间与受管集群的名称相同。

   注： 名称必须是 **volsync**。

2. 输入类似以下示例的命令，将该文件应用到您的配置中：

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

3. 为其他受管集群重复上述步骤。
应该在每个对应受管集群的命名空间中自动创建一个 **ManagedClusterAddOn** 资源。

1.2.1.2.3. 更新 VolSync *ManagedClusterAddOn*

根据您使用的 Red Hat Advanced Cluster Management 版本，您可能需要更新 VolSync 版本。要更新 VolSync **ManagedClusterAddOn** 资源，请完成以下步骤：

1. 在 **ManagedClusterAddOn** 资源中添加以下注解：

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

2. 定义您要从中部署 VolySync 的 **operator-subscription-channel**。

3. 进入 **ManagedClusterAddOn** 资源并验证您已更新了 Volsync 版本，并确认包括了您选择的 **operator-subscription-channel**。

1.2.1.3. 配置 Rsync-TLS 复制

您可以使用 Rsync-TLS 复制创建持久性卷的 1:1 异步复制。您可以使用基于 Rsync-TLS 的复制进行灾难恢复，或者将数据发送到远程站点。使用 Rsync-TLS 时，Volis 在 stunnel 提供的 TLS 保护隧道中使用 Rsync 来同步数据。如需更多信息，请参阅 stunnel 文档。

以下示例演示了如何使用 Rsync-TLS 方法配置。有关 Rsync-TLS 的更多信息，请参阅 VolSync 文档中的使用情况。

1.2.1.3.1. 在受管集群中配置 Rsync-TLS 复制

对于基于 Rsync-TLS 的复制，请在源和目标集群上配置自定义资源。自定义资源使用 **address** 值将源连接到目的地，以及 stunnel 提供的 TLS 保护隧道，以确保传输的数据安全。

参阅以下信息和示例，将 Rsync-TLS 复制的配置从使用 **source-ns** 命名空间中的 **source** 集群中的持久性卷声明，改为使用 **destination-ns** 命名空间中的 **destination** 集群上的持久性卷声明。在需要时替换值：

1. 配置您的目标集群。

   a. 在目标集群中运行以下命令以创建命名空间：

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

   将 **destination-ns** 替换为复制目的地所在的命名空间。

   b. 创建名为 **replication_destination** 的新 YAML 文件，并复制以下内容：

   ```
   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. 在本例中，使用 **LoadBalancer** 的 **ServiceType** 值。负载均衡器服务由源集群创建，以便您的源集群可以将信息传送到不同的目标受管集群。如果您的源和目标位于同一集群中，或者配置了 Submariner 网络服务，则可以使用 **ClusterIP** 作为服务类型。记录下在配置源集群时引用的 secret 的地址和名称。确保 **capacity** 值与正在复制的持久性卷声明的容量匹配。
2. 确保 **capacity** 值与正在复制的持久性卷声明的容量匹配。

可选：如果您使用与环境默认值不同的存储类和卷快照类名称，请指定 **storageClassName** 和 **volumeSnapshotClassName** 参数的值。

c. 在目标集群中运行以下命令以创建 **replicationdestination** 资源：

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

将 **destination-ns** 替换为目的地所在的命名空间的名称。

创建 **replicationdestination** 资源后，以下参数和值会添加到资源中：

参数	值
.status.rsyncTLS.address	用于连接的目标集群的 IP 地址，用于启用源和目标集群进行通信。
.status.rsyncTLS.keySecret	包含与源集群验证连接的 TLS 密钥的 secret 名称。

d. 运行以下命令复制 **.status.rsyncTLS.address** 的值，以便在源集群中使用：将 **destination** 替换为复制目标自定义资源的名称。将 **destination-ns** 替换为目的地所在的命名空间的名称。

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

输出结果类似如下，这适用于 Amazon Web Services 环境：

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

e. 运行以下命令来复制 secret 的名称：

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

将 **destination** 替换为复制目标自定义资源的名称。

将 **destination-ns** 替换为目的地所在的命名空间的名称。

在配置源时，您必须在源集群中输入它。输出应该是 SSH 密钥 secret 文件的名称，该文件可能类似以下名称：

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

f. 通过针对目标集群输入以下命令来从目标集群复制密钥 secret：

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

将 **destination-ns** 替换为复制目的地所在的命名空间。

g. 输入以下命令在 **vi** 编辑器中打开 secret 文件：

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

h. 在目标集群的 open secret 文件中进行以下更改：

- 将命名空间更改为源集群的命名空间。本例中是 **source-ns**。
- 删除所有者引用(**.metadata.ownerReferences**)。

i. 在源集群中，在源集群中输入以下命令来创建 secret 文件：

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

2. 找到您要复制的源持久性卷声明。
注： 源持久性卷声明必须位于 CSI 存储类中。

3. 创建 **ReplicationSource** 项。

a. 在源集群中创建一个名为 **replication_source** 的新 YAML 文件，并复制以下内容：

```
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. 将 **source** 替换为复制源自定义资源的名称。有关如何替换此功能的说明，请参阅此流程的第 *3-vi* 步。
2. 将 **source-ns** 替换为源所在持久性卷声明的命名空间。有关如何替换此功能的说明，请参阅此流程的第 *3-vi* 步。
3. 将 **persistent_volume_claim** 替换为源持久性卷声明的名称。
4. 将 **mykeysecret** 替换为从目标集群复制到源集群的 secret 的名称(**$KEYSECRET**的值)。
5. 将 **my.host.com** 替换为您在配置 **ReplicationDestination** 的 **.status.rsyncTLS.address** 字段复制的主机地址。您可以在下一步中找到 **sed** 命令示例。

如果您的存储驱动程序支持克隆，使用 **Clone** 作为 **copyMethod** 的值，则可能是更精简的复制过程。

可选：如果您使用与环境默认值不同的存储类和卷快照类名称，请指定 **storageClassName** 和 **volumeSnapshotClassName** 参数的值。

现在，您可以设置持久性卷的同步方法。

b. 在源集群中，输入以下命令替换 **ReplicationSource** 对象中的 **address** 和 **keySecret** 的值来修改 **replication_source.yaml** 文件：

```
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
```

将 **my.host.com** 替换为您在配置 **ReplicationDestination** 的 **.status.rsyncTLS.address** 字段复制的主机地址。

将 **keySecret** 替换为您在配置时从 **ReplicationDestination** 的 **.status.rsyncTLS.keySecret** 字段复制的密钥。

使用源所在的持久性卷声明的名称替换 **source**。

注： 您必须在与要复制的持久性卷声明相同的命名空间中创建该文件。

c. 在 **ReplicationSource** 对象中运行以下命令来验证复制是否完成：

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

将 **source-ns** 替换为源所在持久性卷声明的命名空间。

将 **source** 替换为复制源自定义资源的名称。

如果复制成功，输出应类似以下示例：

```
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
```

如果 **Last Sync Time** 没有列出时间，则复制不会完成。

您有原始持久性卷声明的副本。

1.2.1.4. 配置 Rsync 复制

重要：使用 Rsync-TLS 而不是 Rsync 来提高安全性。通过使用 Rsync-TLS，您可以避免使用复制持久性卷不需要的提升用户权限。

您可以使用 Rsync 复制创建持久性卷的 1:1 异步复制。您可以使用基于 Rsync 的复制进行灾难恢复，或者将数据发送到远程站点。

以下示例演示了如何使用 Rsync 方法配置。

1.2.1.4.1. 在受管集群中配置 Rsync 复制

对于基于 Rsync 的复制，请在源和目标集群上配置自定义资源。自定义资源使用 **address** 值将源连接到目的地，**sshKeys** 则用于确保传输的数据安全。

注： 您必须将 **address** 和 **sshKeys** 的值从目的地复制到源，因此请在配置源前配置目的地。

本例显示了一个步骤，将 Rsync 复制的配置从使用 **source-ns** 命名空间中的 **source** 集群中的持久性卷声明，改为使用 **destination-ns** 命名空间中的 **destination** 集群上的持久性卷声明。如果需要，您可以将这些值替换为其他值。

1. 配置您的目标集群。
   a. 在目标集群中运行以下命令以创建命名空间：

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

      将 **destination-ns** 替换为包含目标持久性卷声明的命名空间的名称。

   b. 复制以下 YAML 内容，以创建名为 **replication_destination.yaml** 的新文件：

      ```
      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
```

注意：容量值应与正在复制的持久卷声明的容量匹配。

将 **destination** 替换为复制目的地 CR 的名称。

将 **destination-ns** 替换为目的地所在的命名空间的名称。

在本例中，使用 **LoadBalancer** 的 **ServiceType** 值。负载均衡器服务由源集群创建，以便您的源集群可以将信息传送到不同的目标受管集群。如果您的源和目标位于同一集群中，或者配置了 Submariner 网络服务，则可以使用 **ClusterIP** 作为服务类型。记录配置源集群时要引用的 secret 的地址和名称。

storageClassName 和 **volumeSnapshotClassName** 是可选参数。指定您的环境的值，特别是如果您使用与环境默认值不同的存储类和卷快照类名称。

c. 在目标集群中运行以下命令以创建 **replicationdestination** 资源：

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

将 **destination-ns** 替换为目的地所在的命名空间的名称。

创建 **replicationdestination** 资源后，将以下参数和值添加到资源中：

参数	值
.status.rsync.address	用于连接的目标集群的 IP 地址，用于启用源和目标集群进行通信。
.status.rsync.sshKeys	启用保护从源集群到目标集群的数据传输的 SSH 密钥文件的名称。

d. 运行以下命令复制 **.status.rsync.address** 的值，以便在源集群中使用：

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

将 **destination** 替换为复制目标自定义资源的名称。

将 **destination-ns** 替换为目的地所在的命名空间的名称。

输出结果应该类似以下示例，这适用于 Amazon Web Services 环境：

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

e. 运行以下命令来复制 secret 的名称：

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

将 **destination** 替换为复制目标自定义资源的名称。

将 **destination-ns** 替换为目的地所在的命名空间的名称。

在配置源时，您必须在源集群中输入它。输出应该是 SSH 密钥 secret 文件的名称，该文件可能类似以下名称：

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

f. 通过针对目标集群输入以下命令从目标集群复制 SSH secret：

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

将 **destination-ns** 替换为目标所在持久性卷声明的命名空间。

g. 输入以下命令在 **vi** 编辑器中打开 secret 文件：

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

h. 在目标集群的 open secret 文件中进行以下更改：

- 将命名空间更改为源集群的命名空间。本例中是 **source-ns**。
- 删除所有者引用(**.metadata.ownerReferences**)。

i. 在源集群中，在源集群中输入以下命令来创建 secret 文件：

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

2. 找到您要复制的源持久性卷声明。
注： 源持久性卷声明必须位于 CSI 存储类中。

3. 创建 **ReplicationSource** 项。

a. 复制以下 YAML 内容，在源集群上创建一个名为 **replication_source.yaml** 的新文件：

```
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
```

-

将 **source** 替换为复制源自定义资源的名称。有关如何替换此功能的说明，请参阅此流程的第 *3-vi* 步。

将 **source-ns** 替换为源所在持久性卷声明的命名空间。有关如何替换此功能的说明，请参阅此流程的第 *3-vi* 步。

将 **persistent_volume_claim** 替换为源持久性卷声明的名称。

使用您从 **ReplicationDestination** 的 **.status.rsync.sshKeys** 字段复制的密钥替换 **mysshkeys**。

将 **my.host.com** 替换为您在配置 **ReplicationDestination** 的 **.status.rsync.address** 字段复制的主机地址。

如果您的存储驱动程序支持克隆，使用 **Clone** 作为 **copyMethod** 的值，则可能是更精简的复制过程。

storageClassName 和 **volumeSnapshotClassName** 是可选参数。如果您使用与环境默认值不同的存储类和卷快照类名称，请指定这些值。

现在，您可以设置持久性卷的同步方法。

b. 在源集群中，输入以下命令替换 **ReplicationSource** 对象中的 **address** 和 **sshKeys** 的值来修改 **replication_source.yaml** 文件：

```
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
```

将 **my.host.com** 替换为您在配置 **ReplicationDestination** 的 **.status.rsync.address** 字段复制的主机地址。

使用您从 **ReplicationDestination** 的 **.status.rsync.sshKeys** 字段复制的密钥替换 **mysshkeys**。

使用源所在的持久性卷声明的名称替换 **source**。

注： 您必须在与要复制的持久性卷声明相同的命名空间中创建该文件。

c. 在 **ReplicationSource** 对象中运行以下命令来验证复制是否完成：

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

将 **source-ns** 替换为源所在持久性卷声明的命名空间。

将 **source** 替换为复制源自定义资源的名称。

如果复制成功，输出应类似以下示例：

```
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
```

如果 **Last Sync Time** 没有列出时间，则复制不会完成。

您有原始持久性卷声明的副本。

1.2.1.5. 配置剩余的备份

基于 restic 的备份将持久性卷的 restic 备份副本复制到在 **restic-config.yaml** secret 文件中指定的位置。剩余的备份不会在集群之间同步数据，而是提供数据备份。

完成以下步骤以配置基于剩余的备份：

1. 通过创建类似以下 YAML 内容的 secret 来指定存储备份镜像的存储库：

```
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
```

将 **my-restic-repository** 替换为您要存储备份文件的 S3 存储桶存储库的位置。

将 **my-restic-password** 替换为访问存储库所需的加密密钥。

如果需要，将 **access** 和 **password** 替换为您的供应商凭证。

如果您需要准备新存储库，请参阅为流程准备新存储库。如果使用这个步骤，请跳过运行 **restic init** 命令的步骤来初始化存储库。VolSync 在第一个备份过程中自动初始化存储库。

重要：当将多个持久性卷声明备份到同一 S3 存储桶时，存储桶的路径对于每个持久性卷声明来说必须是唯一的。每个持久性卷声明都使用单独的 **ReplicationSource** 备份，每个声明都需要单独的 restic-config secret。

通过共享相同的 S3 存储桶，每个 **ReplicationSource** 具有对整个 S3 存储桶的写入访问权限。

2. 通过创建类似以下 YAML 内容的 **ReplicationSource** 对象来配置备份策略：

```
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
```

使用您要备份的持久性卷声明替换 **source**。

将 **schedule** 值替换为运行备份的频率。这个示例有每 30 分钟的调度。有关设置计划的更多信息，请参阅调度同步。

将 **PruneIntervalDays** 值替换为重新打包数据实例之间经过的天数，以节省空间。修剪操作可在其运行时生成大量 I/O 流量。

将 **restic-config** 替换为在第 1 步中创建的 secret 的名称。

将 **retain** 的值设置为备份镜像的保留策略。

最佳实践： 将 **Clone** 用于 **CopyMethod** 的值，以确保保存点镜像。

注：默认情况下，Restic movers 在没有 root 权限的情况下运行。如果要以 root 用户身份运行 restic movers，请运行以下命令将升级的权限注解添加到您的命名空间。

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

将 **<namespace>** 替换为您的命名空间的名称。

1.2.1.5.1. 恢复剩余的备份

您可以将复制的数据从其余备份恢复到新的持久性卷声明。**最佳实践：**仅将一个备份恢复到新的持久性卷声明中。要恢复剩余的备份，请完成以下步骤：

1. 创建新的持久性卷声明，使其包含类似以下示例的新数据：

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

-

将 **pvc-name** 替换为新持久性卷声明的名称。

2. 创建一个 **ReplicationDestination** 自定义资源，该资源类似以下示例来指定恢复数据的位置：

```
apiVersion: volsync.backube/v1alpha1
kind: ReplicationDestination
metadata:
  name: <destination>
spec:
  trigger:
    manual: restore-once
  restic:
    repository: <restic-repo>
    destinationPVC: <pvc-name>
    copyMethod: Direct
```

将 **destination** 替换为复制目的地 CR 的名称。

使用存储源的仓库的路径替换 **restic-repo**。

使用您要恢复数据的新持久性卷声明的名称替换 **pvc-name**。使用现有的持久性卷声明，而不是置备一个新的持久性卷声明。

恢复过程只需要完成一次，本例恢复最新的备份。有关恢复选项的更多信息，请参阅 VolSync 文档中的恢复选项。

1.2.1.6. 配置 Rclone 复制

Rclone 备份通过中间对象存储位置（如 AWS S3）使用 Rclone 将单个持久性卷复制到多个位置。将数据分发到多个位置时非常有用。

完成以下步骤以配置 Rclone 复制：

1. 创建一个类似以下示例的 **ReplicationSource** 自定义资源：

```
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>
```

将 **source-pvc** 替换为复制源自定义资源的名称。

将 **source-ns** 替换为源所在持久性卷声明的命名空间。

使用您要复制的持久性卷声明替换 **source**。

将 **schedule** 值替换为运行复制的频率。这个示例有每 6 分钟进行一次的调度。这个值必须包括在引号内。如需更多信息，请参阅调度同步。

将 **intermediate-s3-bucket** 替换为 Rclone 配置文件配置部分的路径。

将 **destination-bucket** 替换为您要复制文件的对象存储桶的路径。

将 **rclone-secret** 替换为包含您的 Rclone 配置信息的 secret 名称。

将 **copyMethod** 的值设置为 **Clone**、**Direct** 或 **Snapshot**。这个值指定是否生成点时复制，如果是，则使用什么方法生成它。

将 **my-sc-name** 替换为您要用于点复制的存储类的名称。如果没有指定，则使用源卷的存储类。

如果您将 **my-vsc** 指定为 **copyMethod**，则将 **my-vsc** 替换为 **VolumeSnapshotClass** 的名称。对于其他类型的 **copyMethod**，这并不是必需的。

2. 创建一个类似以下示例的 **ReplicationDestination** 自定义资源：

```
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>
```

将 **schedule** 值替换为将复制移到目的地的频率。源和目标的调度必须是偏移的，以允许数据在从目的地拉取前完成复制。这个示例有每 6 分钟的调度，将偏移 3 分钟。这个值必须包括在引号内。有关调度的更多信息，请参阅调度同步。

将 **intermediate-s3-bucket** 替换为 Rclone 配置文件配置部分的路径。

将 **destination-bucket** 替换为您要复制文件的对象存储桶的路径。

将 **rclone-secret** 替换为包含您的 Rclone 配置信息的 secret 名称。

将 **copyMethod** 的值设置为 **Clone**、**Direct** 或 **Snapshot**。这个值指定是否生成点时复制，如果是，则使用什么方法生成它。

accessModes 的值指定持久性卷声明的访问模式。有效值为 **ReadWriteOnce** 或 **ReadWriteMany**。

capacity 指定目标卷的大小，它必须足够大来包含传入的数据。

将 **my-sc** 替换为您要用作点时副本的存储类的名称。如果没有指定，则使用系统存储类。

如果您将 **my-vsc** 指定为 **copyMethod**，则将 **my-vsc** 替换为 **VolumeSnapshotClass** 的名称。对于其他类型的 **copyMethod**，这并不是必需的。如果没有包括，则使用系统默认 **VolumeSnapshotClass**。

注：默认情况下，Rclone movers 运行没有 root 权限。如果要以 root 用户身份运行 Rclone movers，请运行以下命令将升级的权限注解添加到您的命名空间。

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

将 **<namespace>** 替换为您的命名空间的名称。

1.2.1.7. 其他资源

如需更多信息，请参阅以下内容：

- 请参阅为 Rsync-TLS 复制创建 secret，以了解如何为 Rsync-TLS 复制创建自己的 secret。
- 有关 Rsync 的更多信息，请参阅 VolSync 文档中的使用情况。
- 有关 restic 选项的更多信息，请参阅 VolSync 文档中的备份选项。
- 返回到在受管集群上安装 VolSync

1.2.2. 将复制镜像转换为可用的持久性卷声明

您可能需要将复制镜像转换为持久性卷声明来恢复数据。

当您使用 **VolumeSnapshot** 从 **ReplicationDestination** 位置复制或恢复 pesistent 卷声明时，会创建一个 **VolumeSnapshot**。**VolumeSnapshot** 包含最近一次成功同步的 **latestImage**。镜像的副本必须转换为持久性卷声明，然后才能使用它。VolSync **ReplicationDestination** 卷填充器可用于将镜像副本转换为可用持久性卷声明。

1. 使用 **dataSourceRef** 创建一个持久性卷声明，指向您要恢复持久性卷声明的 **ReplicationDestination**。这个持久性卷声明会填充在 **ReplicationDestination** 自定义资源定义的 **status.latestImage** 设置中指定的 **VolumeSnapshot** 的内容。
   以下 YAML 内容显示了可能使用的持久性卷声明示例：

```
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
```

将 **pvc-name** 替换为您的新持久性卷声明的名称。

将 **destination-ns** 替换为持久性卷声明和 **ReplicationDestination** 所在的命名空间。

将 **replicationdestination_to_replace** 替换为 **ReplicationDestination** 名称。

最佳实践：当值至少与初始源持久性卷声明大小相同时，您可以使用不同的值更新 **resources.requests.storage**。

2. 输入以下命令验证您的持久性卷声明是否在环境中运行：

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

备注：

如果没有 **latestImage**，持久性卷声明会处于待处理状态，直到 **ReplicationDestination** 完成且快照可用为止。您可以创建一个 **ReplicationDestination** 和一个使用 **ReplicationDestination** 的持久性卷控制器。持久性卷声明仅在 **ReplicationDestination** 完成复制和快照可用后启动卷填充过程。您可以在 **.status.latestImage** 中找到快照。

另外，如果所使用的存储类具有 **WaitForFirstConsumer** 的 **volumeBindingMode** 值，则卷填充器会等待到填充持久性卷声明的消费者为止。当消费者需要访问时，如希望挂载持久性卷声明的 pod，则卷会被填充。VolSync 卷填充器控制器使用 **ReplicationDestination** 中的 **latestImage**。每次创建持久性卷控制后，每次复制都会更新 **latestImage**。

1.2.3. 调度同步

在确定如何启动复制时，从三个选项中选择：始终运行、按计划或手动运行。调度复制是一个经常选择的选项。

Schedule 选项在计划的时间运行复制。调度由 **cronspec** 定义，因此调度可配置为间隔或特定时间。调度值的顺序为：

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

复制将在调度的时间发生时开始。您为此复制选项的设置可能类似以下内容：

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

启用其中一种方法后，同步调度会根据您配置的方法运行。

如需了解更多信息和选项，请参阅 VolSync 文档。

1.2.4. VolSync 高级配置

您可以在复制持久性卷时进一步配置 VolSync，如创建自己的 secret。

1.2.4.1. 为 Rsync-TLS 复制创建 secret

源和目标必须具有对 TLS 连接的共享密钥的访问权限。您可以在 **keySecret** 字段中找到密钥位置。如果您没有在 **.spec.rsyncTLS.keySecret** 中提供 secret 名称，secret 名称会被自动生成并添加到 **.status.rsyncTLS.keySecret** 中。

要创建自己的 secret，请完成以下步骤：

1. secret 使用以下格式：**<id>:<at_least_32_hex_digits>**
   请参见以下示例：**1:23b7395fafc3e842bd8ac0fe142e6ad1**

2. 请参阅以下与上例对应的 **secret.yaml** 示例：

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