Red Hat Advanced Cluster Management for Kubernetes 2.10

비즈니스 연속성

비즈니스 연속성

Last Updated: 2024-09-05

Red Hat Advanced Cluster Management for Kubernetes 2.10 비즈니스 연속성

비즈니스 연속성

법적 공지

초록

클러스터, 재해 복구 등에 대한 자세한 내용을 읽으십시오.

차례

1장. 비즈니스 연속성 **3**
1.1. 백업 및 복원 3
1.2. VXLANSYNC 영구 볼륨 복제 서비스 33

1장. 비즈니스 연속성

재해 복구 솔루션 및 허브 클러스터 및 관리 클러스터에 대한 다음 항목을 참조하십시오.

- 백업 및 복원
  - Operator 아키텍처 백업 및 복원
  - 활성 수동 허브 클러스터 구성
  - 백업 및 복원 Operator 설치
  - 백업 예약 및 복원
- volSync를 사용하여 영구 볼륨 복제
  - volSync를 사용하여 영구 볼륨 복제
  - 복제된 이미지를 사용 가능한 영구 볼륨 클레임으로 변환
  - 동기화 예약

1.1. 백업 및 복원

클러스터 백업 및 복원 Operator는 hub 클러스터에서 실행되며 Kubernetes Hub 클러스터 장애에 대한 Red Hat Advanced Cluster Management를 위한 재해 복구 솔루션을 제공합니다. 허브 클러스터가 실패하면 정책 구성 기반 경고 또는 클러스터 업데이트와 같은 일부 기능이 모든 관리 클러스터가 계속 작동하는 경우에도 작동하지 않습니다. 허브 클러스터를 사용할 수 없게 되면 복구 가능 여부를 결정하거나 새로 배포된 허브 클러스터에서 데이터를 복구해야 하는지 확인해야 합니다.

백업 및 복원 구성 요소는 정책을 사용하여 기본 허브 클러스터를 사용할 수 없는 시기를 관리자에게 알리고 복원 작업이 필요할 수 있습니다. 백업 솔루션이 예상대로 작동하지 않는 경우 관리자에게 경고하고 기본 허브 클러스터가 활성 상태이고 클러스터를 관리하더라도 백업 데이터 문제가 보고됩니다.

클러스터 백업 및 복원 Operator는 OADP Operator 를 사용하여 Velero를 설치하고 허브 클러스터에서 데이터가 저장된 백업 스토리지 위치로 연결을 생성합니다. Velero는 백업 및 복원 작업을 실행하는 구성 요소입니다. 클러스터 백업 및 복원 Operator 솔루션은 관리 클러스터, 애플리케이션 및 정책을 포함하여 모든 Red Hat Advanced Cluster Management Hub 클러스터 리소스에 대한 백업 및 복원 지원을 제공합니다.

클러스터 백업 및 복원 Operator는 허브 클러스터 설치를 확장하는 타사 리소스의 백업을 지원합니다. 이 백업 솔루션을 사용하면 지정된 시간 간격에 실행되는 cron 기반 백업 일정을 정의할 수 있습니다. 허브 클러스터가 실패하면 새 hub 클러스터를 배포할 수 있으며 백업 데이터를 새 hub 클러스터로 이동합니다.

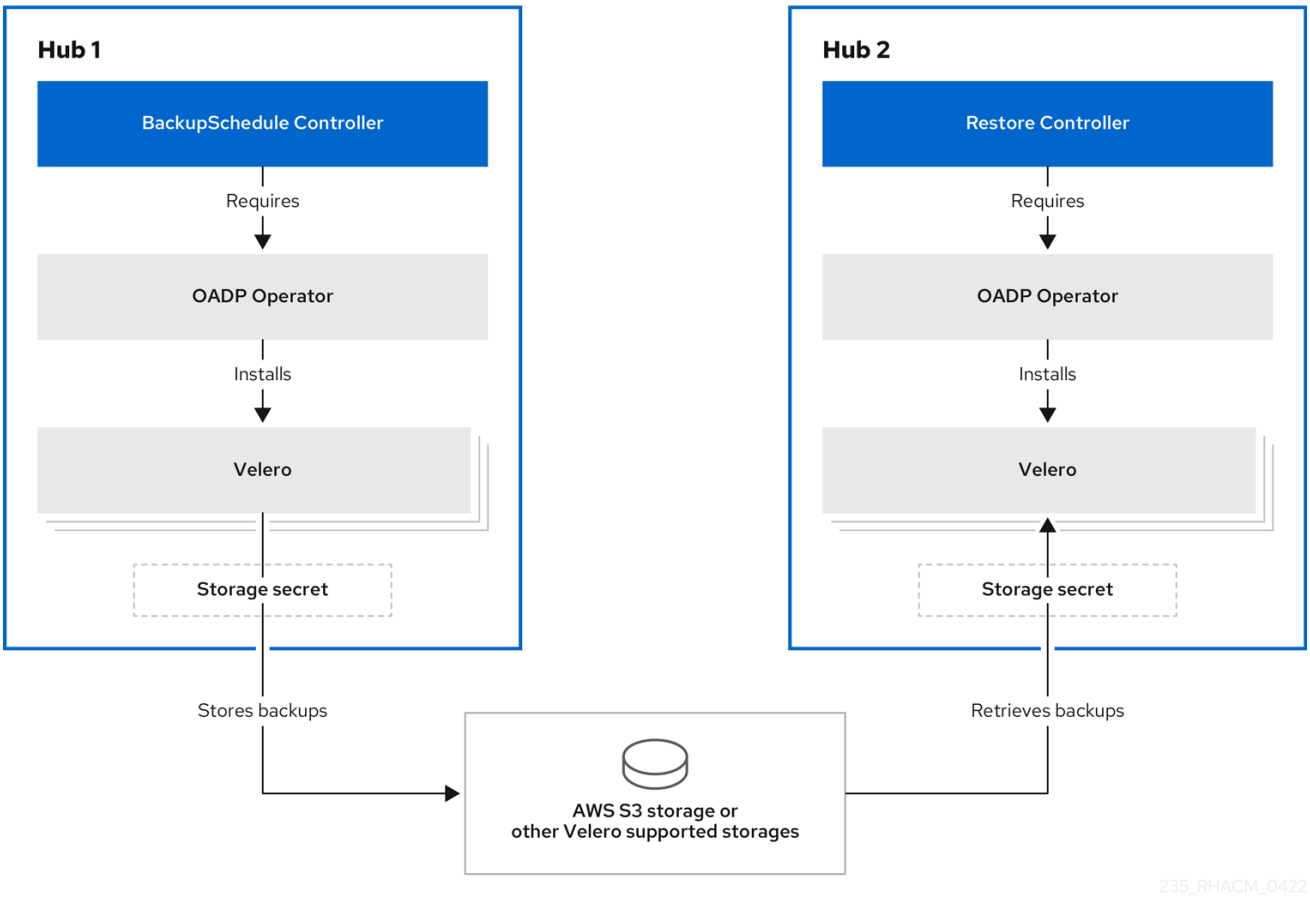
백업 및 복원 Operator에 대한 자세한 내용은 다음 항목을 계속 읽습니다.

- Operator 아키텍처 백업 및 복원
- 활성 수동 허브 클러스터 구성
- 백업 및 복원 Operator 설치
- 백업 예약 및 복원
- 백업 복원
- 백업 또는 복원 구성 검증

- 관리 서비스 계정을 사용하여 자동으로 클러스터 연결
- 고급 구성 백업 및 복원

1.1.1. Operator 아키텍처 백업 및 복원

Operator는 Red Hat Advanced Cluster Management 백업 일정을 설정하는 데 사용되는 **BackupSchedule.cluster.open-cluster-management.io** 리소스와 이러한 백업을 처리하고 복원하는 데 사용되는 **restore.cluster.open-cluster-management.io** 리소스를 정의합니다. Operator는 해당 Velero 리소스를 생성하고 원격 클러스터 및 복원해야 하는 기타 허브 클러스터 리소스를 백업하는 데 필요한 옵션을 정의합니다. 다음 다이어그램을 확인합니다.



1.1.1.1. 백업되는 리소스

클러스터 백업 및 복원 Operator 솔루션은 관리 클러스터, 애플리케이션 및 정책과 같은 모든 허브 클러스터 리소스에 대한 백업 및 복원 지원을 제공합니다. 솔루션을 사용하여 기본 허브 클러스터 설치를 확장하는 타사 리소스를 백업할 수 있습니다. 이 백업 솔루션을 사용하면 지정된 시간 간격에 실행되고 최신 버전의 허브 클러스터 콘텐츠를 지속적으로 백업하는 cron 기반 백업 일정을 정의할 수 있습니다.

허브 클러스터를 교체해야 하거나 허브 클러스터가 실패할 때 재해 시나리오에 있는 경우 새 허브 클러스터를 배포하고 백업할 수 있으며 새 허브 클러스터로 이동됩니다.

백업 데이터를 식별하기 위한 다음 정렬된 클러스터 백업 및 복원 프로세스를 확인합니다.

- **MultiClusterHub** 네임스페이스에서 모든 리소스를 제외합니다. 이는 현재 허브 클러스터 ID에 연결된 설치 리소스를 백업하지 않고 백업해서는 안 됩니다.

- **.open-cluster-management.io** 및 **.hive.openshift.io** 접미사가 지정된 API 버전으로 모든 리소스를 백업합니다. 이러한 접미사는 모든 Red Hat Advanced Cluster Management 리소스가 백업되었음을 나타냅니다.

- **argoproj.io**,**app.k8s.io**,**core.observatorium.io**,**hive.openshift.io** 의 모든 리소스를 백업합니다. 이러한 리소스는 **agent-install.openshift.io** API 그룹의 리소스를 제외하고 **acm-resources-schedule** 백업 내에 백업됩니다. 이러한 리소스는 **acm-managed-clusters-schedule** 백업 내에서 백업됩니다.

- 다음 API 그룹에서 모든 리소스를 제외합니다. **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**.

- 포함된 API 그룹의 일부인 다음 리소스를 모두 제외하지만 필요하지 않거나 백업되는 소유자 리소스에 의해 다시 생성됩니다. **clustermanagementaddon.addon.open-cluster-management.io** ,**backupschedule.cluster.open-cluster-management.io**,**restore.cluster.open-cluster-management.io**,**clusterclaim.cluster.open-cluster-management.io**,**discoveredcluster.discovery.open-cluster-management.io**.

- **cluster.open-cluster-management.io/type**,**hive.openshift.io/secret-type**,**cluster.open-cluster-management.io/backup** 중 하나를 사용하여 보안 및 ConfigMap을 백업합니다.

- 백업하려는 다른 리소스에는 **cluster.open-cluster-management.io/backup** 레이블을 사용하고 이전에 언급된 기준에 포함되지 않거나 제외된 API 그룹의 일부입니다. 다음 예제를 참조하십시오.

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

  참고: hive.openshift.io.ClusterDeployment 리소스에서 사용하는 시크릿은 백업해야 하며 콘솔을 사용하여 클러스터를 생성하는 경우에만 **cluster.open-cluster-management.io/backup** 레이블로 자동으로 주석이 추가됩니다. 대신 GitOps를 사용하여 Hive 클러스터를 배포하는 경우 **ClusterDeployment** 리소스에서 사용하는 보안에 **cluster.open-cluster-management.io/backup** 레이블을 수동으로 추가해야 합니다. **cluster.open-cluster-management.io/backup: cluster-activation** 레이블이 있는 시크릿 및 구성 맵 리소스가 클러스터 활성화 시 복원됩니다.

- 백업할 필요가 없는 특정 리소스를 제외합니다. 백업 프로세스에서 Velero 리소스를 제외하려면 다음 예제를 참조하십시오.

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

1.1.1.2. Red Hat Advanced Cluster Management 일정에 의해 생성된 백업 파일

Red Hat Advanced Cluster Management 일정을 사용하여 리소스 유형 또는 라벨 주석을 기반으로 별도의 백업 파일로 그룹화되는 허브 리소스를 백업할 수 있습니다.

BackupSchedule.cluster.open-cluster-management.io 리소스는 4개의 **schedule.velero.io** 리소스 세트를 생성합니다. 이러한 **schedule.velero.io** 리소스는 리소스라고도 하는 백업 파일을 생성합니다.

예약된 백업 파일 목록을 보려면 다음 명령을 실행합니다. **oc get schedules -A | grep acm**.

예약된 백업 파일은 **backup.velero.io** 입니다. 예약된 백업 파일에 대한 설명을 보려면 다음 표를 참조하십시오.

표 1.1. 예약된 백업 테이블

예약된 백업	설명
인증 정보 백업	Hive 인증 정보, Red Hat Advanced Cluster Management, 사용자 생성 인증 정보 및 **ConfigMap** 을 저장합니다. 이 백업 파일의 이름은 **acm-credentials-schedule-<timestamp>입니다**.
리소스 백업	Red Hat Advanced Cluster Management 리소스, **acm-resources-schedule-<timestamp> 백업 및 일반 리소스인 acm- resources-generic-schedule-<timestamp>** > 용으로 하나의 백업이 포함되어 있습니다. backup 레이블 **cluster.open-cluster-management.io/backup** 으로 주석이 달린 모든 리소스는 백업 **acm-resources-generic-schedule- backup** 에 저장됩니다. 예외는 백업 **acm-credentials-schedule-<timestamp> 에 저장된 시크릿 또는 ConfigMap 리소스입니다**.
관리형 클러스터 백업	허브 클러스터에 대한 관리 클러스터 연결을 활성화하는 리소스만 포함되며 백업이 복원됩니다. 이 백업 파일의 이름은 **acm-managed-clusters-schedule-<timestamp>입니다**.

1.1.1.3. 관리 클러스터 활성화 시 복원된 리소스

cluster.open-cluster-management.io/backup 레이블을 리소스에 추가하면 리소스가 **acm-resources-generic-schedule** 백업에서 자동으로 백업됩니다. 관리 클러스터를 새 허브 클러스터로 이동한 후와 복원된 리소스에서 **veleroManagedClustersBackupName:latest** 를 사용하는 경우에만 리소스를 복원해야 하는 경우 레이블 값을 **cluster-activation** 으로 설정해야 합니다. 이렇게 하면 관리 클러스터 활성화를 호출하지 않는 한 리소스가 복원되지 않습니다. 다음 예제를 확인합니다.

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

참고: 관리되는 클러스터 네임스페이스 또는 해당 리소스의 경우 클러스터 활성화 단계에서 하나씩 복원해야 합니다. 따라서 관리 클러스터 네임스페이스에서 생성된 백업 리소스에 추가해야 하는 경우 **cluster.open-cluster-management.io/backup** 레이블에 **cluster-activation** 값을 사용합니다. 복원 프로세스를 이해하려면 다음 정보를 참조하십시오.

- 네임스페이스를 복원하면 **managedcluster-import-controller** 가 네임스페이스를 삭제합니다.

- **managedCluster** 사용자 정의 리소스를 복원하는 경우 **cluster-manager-registration-controller** 가 네임스페이스를 생성합니다.

cluster.open-cluster-management.io/backup: cluster-activation 라벨을 사용하여 식별되고 **acm-resources-generic-schedule** 백업으로 저장된 활성화 데이터 리소스 외에도 클러스터 백업 및 복원 Operator에는 기본적으로 설정된 활성화에 몇 가지 리소스가 포함됩니다. 다음 리소스는 **acm-managed-clusters-schedule** 백업에서 지원합니다.

- **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. 활성-패시브 허브 클러스터 구성

초기 허브 클러스터가 데이터를 백업하고 하나 이상의 수동 허브 클러스터가 활성 클러스터를 사용할 수 없게 되는 경우 관리 클러스터를 제어하는 데 사용되는 활성-패시브 허브 클러스터 구성을 구성하는 방법을 알아봅니다.

1.1.2.1. 활성-패시브 구성

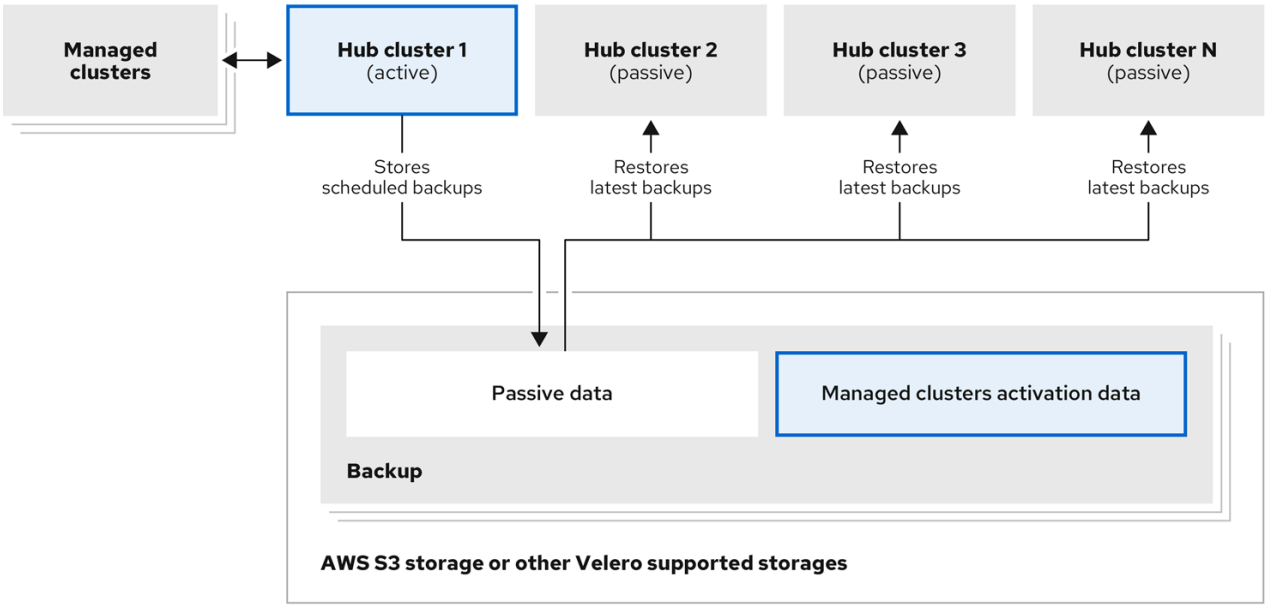
활성-패시브 구성에는 하나의 활성 허브 클러스터 및 패시브 허브 클러스터가 있습니다. 활성 허브 클러스터는 **BackupSchedule.cluster.open-cluster-management.io** 리소스를 사용하여 정의된 시간 간격으로 클러스터를 관리하고 리소스를 백업하는 기본 허브 클러스터로 간주됩니다.

참고: 기본 허브 클러스터 데이터를 백업하려면 **활성-패시브** 구성이 필요하지 않습니다. 허브 클러스터 데이터를 간단히 백업하고 저장할 수 있습니다. 이렇게 하면 문제 또는 오류가 있는 경우 새 허브 클러스터를 배포하고 이 새 허브 클러스터에서 기본 허브 클러스터 데이터를 복원할 수 있습니다. 기본 허브 클러스터 데이터를 복구하는 시간을 줄이기 위해 **활성-패시브** 구성을 사용할 수 있지만 이는 필요하지 않습니다.

패시브 허브 클러스터는 최신 백업을 지속적으로 검색하고 패시브 데이터를 복원합니다. 수동 허브는 **Restore.cluster.open-cluster-management.io** 리소스를 사용하여 새 백업 데이터를 사용할 수 있는 경우 기본 허브 클러스터에서 수동 데이터를 복원합니다. 이러한 허브 클러스터는 기본 허브 클러스터가 실패할 때 기본 허브가 되기 위해 대기 중입니다.

활성 및 수동 허브 클러스터는 동일한 스토리지 위치에 연결됩니다. 여기서 기본 허브 클러스터는 기본 허브 클러스터 백업에 액세스하기 위해 패시브 허브 클러스터의 데이터를 백업합니다. 이 자동 복원 구성을 설정하는 방법에 대한 자세한 내용은 *백업을 확인하는 동안 수동 리소스* 복구를 참조하십시오.

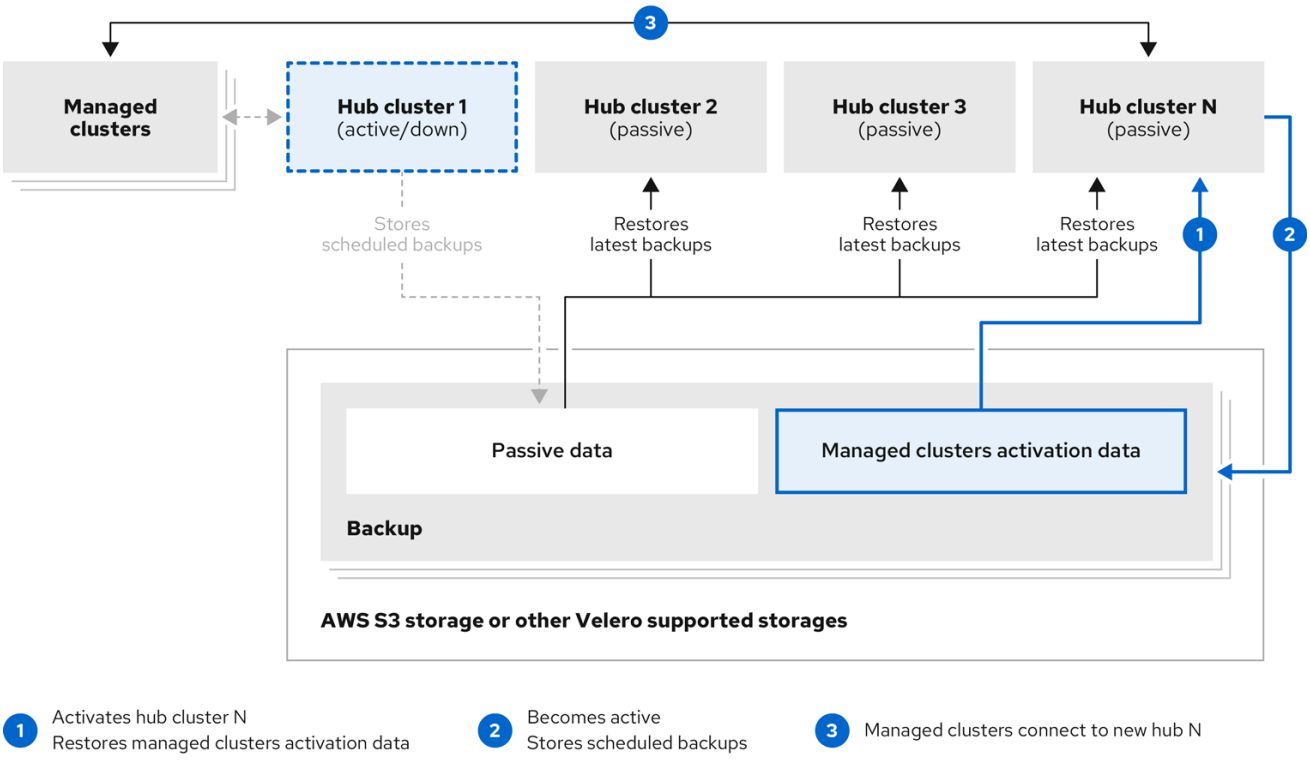
다음 다이어그램에서 활성 허브 클러스터는 로컬 클러스터를 관리하고 정기적으로 허브 클러스터 데이터를 백업합니다.



패시브 허브 클러스터는 관리 클러스터를 패시브 허브 클러스터로 이동하는 관리형 클러스터 활성화 데이터를 제외하고 이 데이터를 복원합니다. 패시브 허브 클러스터는 패시브 데이터를 지속적으로 복원할 수 있습니다. 패시브 허브 클러스터는 수동 데이터를 일회성 작업으로 복원할 수 있습니다. 자세한 *내용은 수동 리소스 복원*을 참조하십시오.

1.1.2.2. 재해 복구

기본 허브 클러스터가 실패하면 관리자가 관리형 클러스터를 인수할 패시브 허브 클러스터를 선택합니다. 다음 이미지에서 관리자는 *Hub 클러스터 N* 을 새 기본 허브 클러스터로 사용하기로 결정합니다.



Hub 클러스터 N 은 관리되는 클러스터 활성화 데이터를 복원합니다. 이 시점에서 관리형 클러스터는 *Hub*

클러스터 N 과 연결됩니다. 관리자는 **BackupSchedule.cluster.open-cluster-management.io** 리소스를 생성하고 초기 기본 허브 클러스터와 동일한 스토리지 위치에 백업을 저장하여 새 기본 허브 클러스터인 *Hub 클러스터 N* 에서 백업을 활성화합니다.

다른 모든 패시브 허브 클러스터는 이제 새 기본 허브 클러스터에서 생성된 백업 데이터를 사용하여 패시브 데이터를 복원합니다. *Hub N* 은 이제 클러스터를 관리하고 데이터를 백업하는 기본 허브 클러스터입니다.

참고:

- 이전 다이어그램의 프로세스 1은 기본 허브 클러스터가 실패했는지 또는 허브 클러스터와 관리 클러스터 간에 네트워크 통신 오류가 있는지 여부를 결정해야 하기 때문에 자동화되지 않습니다. 관리자는 또한 어떤 패시브 허브 클러스터가 기본 허브 클러스터가 될지 결정합니다. 정책 통합: 작업은 백업 정책에서 백업 오류를 보고할 때 작업을 실행하여 이 단계를 자동화하는 데 도움이 될 수 있습니다.

- 이전 다이어그램의 프로세스 2는 수동입니다. 관리자가 새 기본 허브 클러스터에서 백업을 생성하지 않으면 cron 작업으로 적극적으로 실행 중인 백업을 사용하여 관리자에게 알림을 받습니다.

1.1.2.3. 추가 리소스

- 백업을 확인하는 동안 수동 리소스 복원을 참조하십시오.

- 수동 리소스 복원을 참조하십시오.

1.1.3. 백업 및 복원 Operator 설치

클러스터 백업 및 복원 Operator가 자동으로 설치되지 않습니다. Operator를 설치하고 활성화하는 방법을 알아보려면 계속 읽으십시오.

참고:

- 사용자 정의 리소스 definitons는 클러스터 범위이므로 동일한 클러스터에 두 가지 버전의 OADP 또는 Velero를 설치할 수 없습니다. 두 가지 버전이 있는 경우 하나의 버전이 잘못된 사용자 정의 리소스 정의로 실행됩니다.

- **MultiClusterHub** 리소스에서 클러스터 백업 및 restore Operator를 활성화하지 않은 경우 OADP Operator 및 Velero 사용자 정의 리소스 정의가 여전히 hub 클러스터에 설치됩니다. **MultiClusterHub** 리소스는 클러스터 백업 및 복원 Operator를 활성화할 때 설치된 OADP Operator에서 사용하는 버전으로 OADP 및 Velero 사용자 정의 리소스 정의를 조정합니다. 따라서 백업 및 복원 Operator를 활성화할 때 설치된 OADP Operator와 동일한 사용자 정의 리소스 정의를 사용하지 않는 한 Hub 클러스터에 다른 버전의 OADP 또는 Velero를 설치할 수 없습니다.

- 백업 구성 요소는 구성 요소 네임스페이스에 설치된 OADP Operator와 함께 작동합니다.

- backup 및 restore Operator를 사용하려면 먼저 hub 클러스터를 설정해야 합니다.

중요:

OADP Operator를 수동으로 설치하는 경우 OADP Operator 및 Velereo의 사용자 정의 리소스 정의 버전이 정확히 일치해야 합니다. 이러한 버전이 서로 정확히 일치하지 않으면 문제가 발생합니다. 이전에 백업 구성 요소 네임스페이스와 다른 네임스페이스의 hub 클러스터에 OADP Operator를 설치하고 사용한 경우 이 버전을 설치 제거합니다.

Velero는 Kubernetes 허브 클러스터용 Red Hat Advanced Cluster Management의 OADP Operator와 함께 설치됩니다. Red Hat Advanced Cluster Management hub 클러스터 리소스를 백업하고 복원하는 데 사용됩니다.

Velero 에 대해 지원되는 스토리지 공급자 목록은 OADP 설치 정보를 참조하십시오.

Operator를 설치하고 활성화하려면 다음 작업을 완료해야 합니다.

- 백업 및 복원 Operator를 위한 허브 클러스터 설정
- 백업 및 복원 Operator 활성화

1.1.3.1. 백업 및 복원 Operator를 위한 허브 클러스터 설정

backup 및 restore Operator를 사용하려면 hub 클러스터를 설정해야 합니다.

1.1.3.1.1. 스토리지 위치 시크릿 생성

스토리지 위치 시크릿을 생성하려면 다음 단계를 완료합니다.

1. 백업이 저장되는 클라우드 스토리지에 대한 기본 시크릿 생성 단계를 완료합니다.
2. 백업 구성 요소 네임스페이스에 있는 OADP Operator 네임스페이스에 시크릿 리소스를 생성합니다.

1.1.3.1.2. 백업 Operator 활성화

활성 및 수동 허브 클러스터에 대한 백업 Operator를 활성화하려면 다음 단계를 완료합니다.

1. Red Hat OpenShift Container Platform 클러스터에서 Red Hat Advanced Cluster Management for Kubernetes operator 버전 2.10.x를 설치합니다. **MultiClusterHub** 리소스는 Red Hat Advanced Cluster Management를 설치할 때 자동으로 생성되고 다음 상태(**실행 중**)가 표시됩니다.
2. 클러스터 백업 및 복원 Operator를 수동으로 설치합니다.
3. 클러스터 백업 및 복원 운영자(**cluster-backup**)를 활성화합니다.
4. **cluster-backup** 매개변수를 **true** 로 설정하여 **MultiClusterHub** 리소스를 편집합니다. 이 허브 클러스터에 AWS STS(Security Token Service) 옵션이 활성화되어 있지 않은 경우 OADP Operator도 백업 구성 요소와 동일한 네임스페이스에 설치됩니다. STS 옵션이 활성화된 허브 클러스터의 경우 OADP Operator를 수동으로 설치해야 합니다.
5. 복원 허브 클러스터가 백업 허브 클러스터가 사용하는 것과 동일한 Red Hat Advanced Cluster Management 버전을 사용하는지 확인합니다. 백업 허브 클러스터에서 사용하는 버전 이전 버전의 허브 클러스터에서 백업을 복원할 수 없습니다.
6. **선택 사항:** 백업이 생성된 허브 클러스터보다 최신 버전이 있는 허브 클러스터에서 복원 작업을 실행하려면 다음을 완료합니다.
   a. 복원 허브 클러스터에서 백업이 생성된 Hub 클러스터와 동일한 버전으로 Operator를 설치합니다.
   b. 복원 허브 클러스터에서 날짜를 복원합니다.
   c. 복원 허브 클러스터에서 업그레이드 작업을 사용하여 사용하려는 버전으로 업그레이드합니다.

d. hub 클러스터를 수동으로 구성합니다.

e. 활성 허브 클러스터 및 활성 허브 클러스터와 동일한 네임스페이스에 모든 Operator를 설치합니다.

f. Ansible Automation Platform, OpenShift Container Platform GitOps 또는 인증서 관리자와 같은 다른 Operator가 설치되어 있는지 확인합니다.

g. 새 허브 클러스터가 초기 허브 클러스터와 동일한 방식으로 구성되었는지 확인합니다.

h. backup 및 restore operator 및 이전 hub 클러스터에 구성된 operator를 설치할 때 초기 허브 클러스터와 동일한 네임스페이스 이름을 사용합니다.

7. 패시브 허브 클러스터에서 **DataProtectionApplication 리소스** 를 만듭니다.

8. 초기 허브 클러스터가 데이터를 백업한 동일한 스토리지 위치에 패시브 허브 클러스터를 연결합니다.

1.1.3.1.3. DataProtectionApplication 리소스 생성

활성 및 수동 허브 클러스터에 대한 **DataProtectionApplication** 리소스의 인스턴스를 만들려면 다음 단계를 완료하십시오.

1. Red Hat OpenShift Container Platform 콘솔에서 Operator > **설치된 Operator 를 선택합니다**.

2. DataProtectionApplication에서 **인스턴스 만들기** 를 클릭합니다.

3. {ocp-short) 콘솔을 사용하거나 **DataProtectionApplication** 예제에 언급된 YAML 파일을 사용하여 Velero 인스턴스를 생성합니다.

4. **DataProtectionApplication** 네임스페이스를 **open-cluster-management-backup** 으로 설정합니다.

5. **DataProtectionApplication** 리소스에 적합한 사양(**spec:**) 값을 설정합니다. 그런 다음 **생성을** 클릭합니다.
   기본 백업 스토리지 위치를 사용하려는 경우 **backupStorageLocations** 섹션에서 **default: true** 값을 설정합니다. 다음 **DataProtectionApplication** 리소스 샘플을 확인합니다.

```
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. 연결이 끊긴 환경에서 백업 및 복원 구성 요소 활성화

연결이 끊긴 환경에서 Red Hat OpenShift Container Platform을 사용하여 백업 및 복원 구성 요소를 활성화하려면 다음 단계를 완료하십시오.

1. OADP Operator가 설치된 소스를 재정의하도록 다음 주석으로 **MultiClusterHub** 리소스를 업데이트합니다. **MultiClusterHub** 리소스에서 **cluster-backup** 구성 요소가 활성화되기 전에 주석을 생성합니다.

```
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** 는 사용자 정의 이름이며 연결이 끊긴 환경에서 Red Hat OpenShift Operator를 정의하고 액세스하는 데 사용하는 **CatalogSource** 리소스의 이름을 나타냅니다. 다음 명령을 실행하여 **catalogsource** 를 검색합니다.

```
oc get catalogsource -A
```

출력은 다음과 유사할 수 있습니다.

```
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. 백업 및 복원 Operator 활성화

MultiClusterHub 리소스가 처음 생성될 때 클러스터 백업 및 복원 Operator를 활성화할 수 있습니다. **cluster-backup** 매개 변수는 **true** 로 설정됩니다. Operator가 활성화되면 Operator 리소스가 설치됩니다.

MultiClusterHub 리소스가 이미 생성된 경우 **MultiClusterHub** 리소스를 편집하여 클러스터 백업 Operator를 설치하거나 제거할 수 있습니다. 클러스터 백업 Operator를 설치 제거하려면 **cluster-backup** Operator를 **false** 로 설정합니다.

백업 및 복원 Operator가 활성화되면 **MultiClusterHub** 리소스는 다음 YAML 파일과 유사할 수 있습니다.

```
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. 추가 리소스

- Velero 를 참조하십시오.
- 지원되는 Velero 스토리지 공급자 목록은 OpenShift Container Platform 설명서에서 AWS S3 호환 백업 스토리지 공급자를 참조하십시오.
- *DataProtectionApplication* 리소스에 대해 자세히 알아보십시오.

1.1.4. 백업 예약 및 복원

백업을 예약하고 복원하려면 다음 단계를 완료합니다.

1. backup 및 restore operator, **backupschedule.cluster.open-cluster-management.io** 를 사용하여 백업 일정을 생성하고 **restore.cluster.open-cluster-management.io** 리소스를 사용하여 백업을 복원합니다.

2. 다음 명령을 실행하여 **백업schedule.cluster.open-cluster-management.io** 리소스를 생성합니다.

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

cluster_v1beta1_backupschedule.yaml 리소스는 다음 파일과 유사할 수 있습니다.

```
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) 2시간마다 백업을 생성

(2) **선택 사항:** 120h 후에 예약된 백업을 삭제합니다. 지정하지 않으면 최대 Velero 기본값인 720h가 사용됩니다.

백업schedule.cluster.open-cluster-management.io 사양 속성에 대한 다음 설명을 확인합니다.

- **veleroSchedule** 은 필수 속성이며 백업 예약에 필요한 cron 작업을 정의합니다.
- **veleroTtl** 은 선택적 속성이며 예약된 백업 리소스의 만료 시간을 정의합니다. 지정하지 않으면 Velero에서 설정한 최대 기본값이 사용되며, 이는 **720h** 입니다.

3. 3 **schedule.velero.io 리소스에 대한 정의가 표시되는 백업schedule.cluster.open-cluster-management**.io 리소스의 상태를 확인합니다. 다음 명령을 실행합니다.

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

4. 복원 시나리오를 위해 복원 작업이 다른 허브 클러스터에서 실행됩니다. 복원 작업을 시작하려면 백업을 복원하려는 허브 클러스터에서 **restore.cluster.open-cluster-management.io** 리소스를 생성합니다.
참고: 새 허브 클러스터에서 백업을 복원할 때 백업이 생성된 이전 허브 클러스터를 종료했는지 확인합니다. 관리 클러스터 조정에서 관리 클러스터를 더 이상 사용할 수 없는 것으로 확인되면 이전 허브 클러스터가 실행 중인 경우 이전 허브 클러스터에서 관리 클러스터를 다시 가져오려고 합니다.

   클러스터 백업 및 복원 operator, **backupschedule.cluster.open-cluster-management.io** 및 **restore.cluster.open-cluster-management.io** 리소스를 사용하여 백업 또는 복원 리소스를 생성할 수 있습니다. *cluster-backup-operator* 샘플을 참조하십시오.

5. 다음 명령을 실행하여 **restore.cluster.open-cluster-management.io** 리소스를 생성합니다.

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

리소스는 다음 파일과 유사할 수 있습니다.

```
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. 다음 명령을 실행하여 Velero **Restore** 리소스를 확인합니다.

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

7. 다음 명령을 실행하여 Red Hat Advanced Cluster Management **Restore** 이벤트를 확인합니다.

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

YAML 리소스 **복원** 의 매개변수 및 샘플에 대한 설명은 *백업 복원* 섹션을 참조하십시오.

1.1.4.1. 백업 데이터 확장

클러스터에 **cluster.open-cluster-management.io/backup** 레이블을 리소스에 추가하여 클러스터 백업 및 복원을 통해 타사 리소스를 백업할 수 있습니다. 레이블 값은 빈 문자열을 포함하여 모든 문자열이 될 수 있습니다. 백업 중인 구성 요소를 식별하는 데 도움이 되는 값을 사용합니다. 예를 들어 IDP 솔루션에서 구성 요소를 제공하는 경우 **cluster.open-cluster-management.io/backup: idp** 레이블을 사용합니다.

참고: 관리 클러스터 활성화 리소스가 복원될 때 리소스를 복원하려면 **cluster.open-cluster-management.io/backup** 라벨에 **cluster-activation** 값을 사용합니다. 관리 클러스터 활성화 리소스를 복원하면 관리 클러스터가 복원이 시작된 허브 클러스터에서 적극적으로 관리됩니다.

1.1.4.2. 클러스터 백업 예약

backupschedule.cluster.open-cluster-management.io 리소스를 생성할 때 백업 일정이 활성화됩니다. 다음 **백업schedule.cluster.open-cluster-management.io** 샘플을 확인합니다.

```
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 리소스를 생성한 후 다음 명령을 실행하여 예약된 클러스터 백업 상태를 가져옵니다.

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

backupschedule.cluster.open-cluster-management.io 리소스는 백업을 생성하는 데 사용되는 6개의 **schedule.velero.io** 리소스를 생성합니다. 다음 명령을 실행하여 예약된 백업 목록을 확인합니다.

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

리소스는 다음 표에 표시된 대로 그룹에서 별도로 백업됩니다.

표 1.2. 리소스 그룹 테이블

리소스	설명
인증 정보 백업	Hive 인증 정보, Red Hat Advanced Cluster Management 및 사용자 생성 인증 정보 및 ConfigMap 을 저장하는 백업 파일입니다.
리소스 백업	Red Hat Advanced Cluster Management 리소스에 대한 백업 1개와 일반 리소스용 백업이 포함되어 있습니다. 이러한 리소스는 **cluster.open-cluster-management.io/backup** 레이블을 사용합니다.
ManagedClusters backup	허브 클러스터에 대한 관리 클러스터 연결을 활성화하는 리소스만 포함되며 백업이 복원됩니다.

참고: *리소스 백업* 파일에는 관리 클러스터 관련 리소스가 포함되어 있지만 관리 클러스터를 hub 클러스터에 연결하는 리소스의 하위 집합이 포함되어 있지 않습니다. 관리 클러스터를 연결하는 리소스는 활성화 리소스라고 하며 관리 클러스터 백업에 포함됩니다. 새 허브 클러스터에서 *인증 정보* 및 *리소스* 백업에만 대한 백업을 복원할 때 새 허브 클러스터는 Hive API를 분리된 상태로 사용하여 생성된 모든 관리 클러스터를 표시합니다. 가져오기 작업을 사용하여 기본 허브 클러스터에서 가져온 관리형 클러스터는 활성화 데이터가 수동 허브 클러스터에서 복원된 경우에만 표시됩니다. 관리 클러스터는 백업 파일을 생성한 원래 허브 클러스터에 계속 연결됩니다.

활성화 데이터가 복원되면 Hive API를 사용하여 생성된 관리형 클러스터만 새 허브 클러스터와 자동으로 연결됩니다. 다른 모든 관리 클러스터는 *Pending* 상태로 표시됩니다. 새 클러스터에 수동으로 다시 연결해야 합니다.

다양한 **BackupSchedule** 상태에 대한 설명은 다음 표를 참조하십시오.

표 1.3. BackupSchedule 상태 테이블

BackupSchedule 상태	설명
활성화됨	**BackupSchedule** 이 실행 중이고 백업을 생성합니다.

BackupSchedule 상태	설명
FailedValidation	오류로 인해 **BackupSchedule** 이 실행되지 않습니다. 결과적으로 **BackupSchedule** 은 백업을 생성하지 않고 대신 오류를 수정하고 수정되기를 기다리고 있습니다. 리소스가 유효하지 않은 이유는 **BackupSchedule** status 섹션을 참조하십시오. 오류가 해결되면 **BackupSchedule** 상태가 **Enabled** 로 변경되고 리소스가 백업 생성을 시작합니다.
BackupCollision	**BackupSchedule** 은 백업을 생성하지 않습니다. 리소스 상태가 **BackupCollision** 인 이유는 **BackupSchedule** 상태 섹션을 참조하십시오. 백업 생성을 시작하려면 이 리소스를 삭제하고 새 리소스를 생성합니다.

1.1.4.2.1. 백업 충돌 방지

허브 클러스터가 수동 허브 클러스터가 되는 경우 또는 다른 방식으로 기본 허브 클러스터가 되는 경우 백업 충돌이 발생할 수 있으며 다른 방식으로 동일한 스토리지 위치에서 데이터를 백업할 수 있습니다.

결과적으로 더 이상 기본 허브 클러스터로 설정되지 않은 허브 클러스터에서 최신 백업을 생성합니다. **BackupSchedule.cluster.open-cluster-management.io** 리소스가 여전히 활성화되어 있기 때문에 이 허브 클러스터는 여전히 백업을 생성합니다.

백업 충돌을 일으킬 수 있는 두 가지 시나리오에 대해 알아보려면 다음 목록을 참조하십시오.

1. 기본 허브 클러스터는 예기치 않게 실패하므로 다음과 같은 조건이 발생합니다.
   - 기본 허브 클러스터에서 Hub1로의 통신이 실패합니다.
   - Hub1 백업 데이터는 Hub2라는 보조 허브 클러스터에서 복원됩니다.
   - 관리자는 Hub2에 **BackupSchedule.cluster.open-cluster-management.io** 리소스를 생성하여 기본 허브 클러스터이며 공통 스토리지 위치에 백업 데이터를 생성합니다.
   - hub1이 예기치 않게 다시 작동하기 시작합니다.
     Hub1에서 **BackupSchedule.cluster.open-cluster-management.io** 리소스가 계속 활성화되므로 Hub1은 Hub2와 동일한 스토리지 위치에 백업 쓰기를 재개합니다. 두 허브 클러스터 모두 이제 동일한 스토리지 위치에서 백업 데이터를 작성하고 있습니다. 이 스토리지 위치의 최신 백업을 복원하는 허브 클러스터는 Hub2 데이터 대신 Hub1 데이터를 사용할 수 있습니다.
2. 관리자는 Hub2를 기본 허브 클러스터로 설정하여 재해 시나리오를 테스트합니다. 이는 다음과 같은 조건으로 인해 발생합니다.
   - hub1이 중지되었습니다.
   - Hub1 백업 데이터가 Hub2에서 복원됩니다.
   - 관리자는 Hub2에 **BackupSchedule.cluster.open-cluster-management.io** 리소스를 생성하여 기본 허브 클러스터이며 공통 스토리지 위치에 백업 데이터를 생성합니다.
   - 재해 테스트가 완료되면 관리자가 이전 상태로 되돌리고 Hub1을 기본 허브 클러스터로 다시 만듭니다.

- Hub2가 여전히 활성화되어 있는 동안 Hub1이 시작됩니다. **BackupSchedule.cluster.open-cluster-management.io** 리소스는 Hub2에서 계속 활성화되므로 백업 데이터가 손상되는 동일한 스토리지 위치에 백업을 작성합니다. 이 위치의 최신 백업을 복원하는 허브 클러스터는 Hub1 데이터 대신 Hub2 데이터를 사용할 수 있습니다. 이 시나리오에서 Hub2를 중지하거나 Hub1을 시작하기 전에 Hub2에서 **BackupSchedule.cluster.open-cluster-management.io** 리소스를 삭제하면 백업 충돌 문제가 해결됩니다.

백업 충돌을 방지하고 보고하기 위해 **BackupSchedule.cluster.open-cluster-management.io** 리소스에 대한 **BackupCollision** 상태가 있습니다. 컨트롤러는 스토리지 위치의 최신 백업이 현재 허브 클러스터에서 생성되었는지 정기적으로 확인합니다. 그렇지 않은 경우 다른 허브 클러스터에서 최근 스토리지 위치에 백업 데이터를 작성했으며 허브 클러스터가 다른 허브 클러스터와 충돌하고 있음을 나타냅니다.

이 경우 현재 허브 클러스터 **BackupSchedule.cluster.open-cluster-management.io** 리소스 상태가 **BackupCollision** 으로 설정되고 이 리소스에서 생성한 **Schedule.velero.io** 리소스는 데이터 손상을 방지하기 위해 삭제됩니다. **BackupCollision** 은 백업 정책에 의해 보고됩니다. 관리자는 유효하지 않은 허브 클러스터에서 **BackupSchedule.cluster.open-cluster-management.io** 리소스를 제거하고 유효한 기본 허브 클러스터에서 새 **BackupSchedule.cluster.open-cluster-management.io** 리소스를 생성하기 전에 스토리지 위치에 쓰는 허브 클러스터를 확인하고 백업을 다시 시작합니다.

다음 명령을 실행하여 백업 충돌이 있는지 확인합니다.

```
oc get backupschedule -A
```

백업 충돌이 있는 경우 출력은 다음 예와 유사할 수 있습니다.

```
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. 추가 리소스

- YAML 리소스 복원의 매개변수 및 샘플에 대한 설명은 백업 **복원** 섹션을 참조하십시오.
- 자세한 내용은 백업 및 복원을 참조하십시오.

1.1.5. 백업 복원

일반적인 복원 시나리오에서는 백업이 실행되는 허브 클러스터를 사용할 수 없게 되고 백업 데이터를 새 허브 클러스터로 이동해야 합니다. 이 작업은 새 허브 클러스터에서 클러스터 복원 작업을 실행하여 수행됩니다. 이 경우 복원 작업은 백업이 생성되는 위치와 다른 허브 클러스터에서 실행됩니다.

이전 스냅샷에서 데이터를 복구할 수 있도록 백업이 수집된 동일한 허브 클러스터에서 데이터를 복원하려는 경우도 있습니다. 이 경우 복원 및 백업 작업이 모두 동일한 허브 클러스터에서 실행됩니다.

hub 클러스터에서 **restore.cluster.open-cluster-management.io** 리소스를 생성한 후 다음 명령을 실행하여 복원 작업의 상태를 가져올 수 있습니다.

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

백업 파일에 포함된 백업 리소스가 생성되었는지 확인할 수도 있습니다.

참고: restore.cluster.open-cluster-management.io 리소스는 **syncRestoreWithNewBackups** 옵션을 사용하고 복구 수동 리소스 섹션에 언급된 대로 **true** 로 설정하지 않는 한 한 번 실행됩니다. 복원 작업이 완료된 후 동일한 복원 작업을 다시 실행하려면 동일한 **사양** 옵션을 사용하여 새 **restore.cluster.open-cluster-management.io** 리소스를 생성해야 합니다.

복원 작업은 백업 작업에서 만든 3가지 백업 유형을 모두 복원하는 데 사용됩니다. 관리 클러스터만, 사용자 인증 정보 또는 허브 클러스터 리소스만 설치하도록 선택할 수 있습니다.

복원은 백업 파일 유형에 대해 복원 논리가 정의된 다음 3가지 필수 **spec** 속성을 정의합니다.

- **veleroManagedClustersBackupName** 은 관리되는 클러스터 활성화 리소스에 대한 복원 옵션을 정의하는 데 사용됩니다.
- **veleroCredentialsBackupName** 은 사용자 인증 정보에 대한 복원 옵션을 정의하는 데 사용됩니다.
- **veleroResourcesBackupName** 은 hub 클러스터 리소스(**애플리케이션**,**정책** 및 관리 클러스터 수동 데이터와 같은 기타 허브 클러스터 리소스)에 대한 복원 옵션을 정의하는 데 사용됩니다.
  앞서 언급한 속성에 유효한 옵션은 다음과 같습니다.
  - **latest** - 이 속성은 이 유형의 백업에 마지막으로 사용 가능한 백업 파일을 복원합니다.
  - **skip** - 이 속성은 현재 복원 작업을 사용하여 이 유형의 백업을 복원하지 않습니다.
  - **<backup_name** > - 이 속성은 지정된 백업을 이름으로 복원합니다.

restore.cluster.open-cluster-management.io 에서 생성한 **restore.velero.io** 리소스의 이름은 < **restore.cluster.open-cluster-management.io name>-<velero-backup-resource-name** > 을 사용하여 생성됩니다. 다음 설명을 확인합니다.

- **restore.cluster.open-cluster-management.io 이름은** 복원을 시작하는 현재 **restore.cluster.open-cluster-management.io** 리소스의 이름입니다.
- **Velero-backup-resource-name** 은 데이터를 복원하는 데 사용되는 Velero 백업 파일의 이름입니다. 예를 들어 **restore-acm** 이라는 **restore.cluster.open-cluster-management.io** 리소스는 **restore.velero.io** 복원 리소스를 생성합니다. 형식에 대한 다음 예제를 확인합니다.
  - **restore-acm-acm-managed-clusters-schedule-20210902205438** 은 관리되는 클러스터 활성화 데이터 백업을 복원하는 데 사용됩니다. 이 샘플에서 리소스를 복원하는 데 사용되는 **backup.velero.io** 백업 이름은 **acm-managed-clusters-schedule-20210902205438** 입니다.
  - **restore-acm-acm-credentials-schedule-20210902206789** 는 인증 정보 백업을 복원하는 데 사용됩니다. 이 샘플에서 리소스를 복원하는 데 사용되는 **backup.velero.io** 백업 이름은 **acm-managed-clusters-schedule-20210902206789** 입니다.
  - **restore-acm-acm-resources-schedule-20210902201234** 는 관리형 클러스터 수동 데이터 백업과 같은 애플리케이션, 정책 및 기타 허브 클러스터 리소스를 복원하는 데 사용됩니다. 이 샘플에서 리소스를 복원하는 데 사용되는 **backup.velero.io** 백업 이름은 **acm-managed-clusters-schedule-20210902201234** 입니다.

참고: skip 이 백업 유형에 사용되는 경우 **restore.velero.io** 가 생성되지 않습니다.

클러스터 **복원** 리소스의 다음 YAML 샘플을 확인합니다. 이 샘플에서는 사용 가능한 최신 백업 파일을 사용하여 백업 파일의 세 가지 유형이 모두 복원됩니다.

```
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
```

참고: 관리형 클러스터 백업의 **acm-managed-clusters** 백업이 다른 허브 클러스터에서 복원되는 경우 Hive API에서 생성한 관리 클러스터만 새 허브 클러스터와 자동으로 연결됩니다. 다른 모든 관리 클러스터는 **Pending Import** 상태로 유지되며 새 hub 클러스터로 다시 가져와야 합니다. 자세한 내용은 가져온 관리 클러스터 복원을 참조하십시오.

1.1.5.1. 데이터를 초기 기본 허브로 다시 복원

클러스터에서 백업 데이터를 복원해야 하는 경우 복원 작업 전에 사용자 리소스가 생성된 허브 클러스터 또는 기본 허브 클러스터를 사용하는 대신 새 클러스터를 생성합니다. 허브 클러스터 복원 작업 중에 복원 중인 백업 데이터의 일부가 아닌 경우 기존 리소스를 정리하도록 hub 클러스터 백업 복원을 구성할 수 있습니다. 복원은 이전 백업에서 만든 리소스를 정리하지만 사용자 리소스를 정리하지는 않습니다. 결과적으로 이 허브 클러스터에서 사용자가 생성한 리소스는 정리되지 않으므로 이 허브 클러스터의 데이터는 복원된 리소스와 함께 사용 가능한 데이터를 반영하지 않습니다.

패시브 허브에서만 복원 작업을 검증한 다음 기본 허브 클러스터를 사용하도록 다시 이동하는 재해 복구 테스트는 기본 허브 클러스터를 수동 클러스터로 사용할 수 있는 한 가지 상황입니다. 이 복구 테스트 시나리오에서는 허브 백업 시나리오만 테스트 중입니다. 기본 허브 클러스터를 사용하여 새 리소스를 생성하지 않습니다. 대신 백업 데이터가 기본 허브 클러스터에서 패시브 허브 클러스터로 일시적으로 이동했습니다. 모든 리소스를 이 허브 클러스터로 복원하는 경우 초기 기본 허브 클러스터를 기본 허브 클러스터로 다시 만들 수 있습니다. 이 작업은 관리 클러스터를 이 허브 클러스터로 다시 이동하는 post restore 작업을 실행합니다.

1.1.5.2. 새 허브 클러스터 준비

새 허브 클러스터에서 복원 작업을 실행하기 전에 hub 클러스터를 수동으로 구성하고 초기 허브 클러스터에와 동일한 운영자를 설치해야 합니다. 초기 허브 클러스터와 동일한 네임스페이스에 Red Hat Advanced Cluster Management Operator를 설치하고, *DataProtectionApplication* 리소스를 만든 다음 초기 허브 클러스터가 이전에 데이터를 백업한 동일한 스토리지 위치에 연결해야 합니다.

MultiClusterEngine 리소스에 대한 변경 사항을 포함하여 Red Hat Advanced Cluster Management Operator가 생성한 **MultiClusterHub** 리소스의 초기 허브 클러스터에서와 동일한 구성을 사용합니다.

예를 들어, 초기 허브 클러스터에 Ansible Automation Platform, Red Hat OpenShift GitOps, **cert-manager** 와 같은 다른 Operator가 설치되어 있는 경우 복원 작업을 실행하기 전에 설치해야 합니다. 이렇게 하면 새 허브 클러스터가 초기 허브 클러스터와 동일한 방식으로 구성됩니다.

1.1.5.3. 복원 후 허브 클러스터 정리

Velero는 현재 복원된 백업으로 변경된 경우 기존 리소스를 업데이트합니다. Velero는 이전 복원에서 생성된 리소스이며 현재 복원된 백업의 일부가 아닌 delta 리소스를 정리하지 않습니다. 이렇게 하면 새 허브 클러스터에서 허브 클러스터 데이터를 복원할 때 사용할 수 있는 시나리오가 제한됩니다. 복원이 한 번만 적용되지 않는 한 새 Hub 클러스터를 수동 구성으로 안정적으로 사용할 수 없습니다. 허브 클러스터의 데이터는 복원된 리소스에서 사용 가능한 데이터를 반영하지 않습니다.

이 제한을 해결하기 위해 **Restore.cluster.open-cluster-management.io** 리소스가 생성되면 백업 Operator는 허브 클러스터를 정리하는 후 복원 작업을 실행합니다. 이 작업은 현재 복원된 백업의 일부가 아닌 이전 Red Hat Advanced Cluster Management 복원에서 생성한 모든 리소스를 제거합니다.

복원 후 정리는 **cleanupBeforeRestore** 속성을 사용하여 정리할 오브젝트의 하위 집합을 식별합니다. 복원 후 정리에 다음 옵션을 사용할 수 있습니다.

- **none**: 정리할 필요는 없으며 Velero 복원을 시작합니다. 새로운 허브 클러스터에서는 **None** 을 사용합니다.
- **cleanupRestored : 현재 복원된 백업에 포함되지 않은 이전 Red Hat Advanced Cluster Management 복원에서 만든 모든 리소스를 정리** 합니다.
- **cleanupAll : 복원 작업 결과로 생성되지 않은 경우에도 Red Hat Advanced Cluster Management 백업에 포함될 수 있는 hub 클러스터에서 모든 리소스를 정리** 합니다. 이는 복원 작업이 시작되기 전에 hub 클러스터에서 추가 콘텐츠를 생성할 때 사용해야 합니다.
  best Practice: **cleanupAll** 옵션을 사용하지 마십시오. 극단적인 주의를 두고 마지막 수단으로만 사용하십시오. 또한 **cleanupAll는 사용자가 생성한 hub 클러스터에서 리소스를 정리하고 이전에 복원된 백업으로 만든 리소스도 정리** 합니다. 대신 hub 클러스터가 재해 시나리오에 대한 수동 후보로 지정될 때 hub 클러스터 콘텐츠를 업데이트하지 못하도록 **cleanupRestored** 옵션을 사용합니다. 클린 허브 클러스터를 수동 클러스터로 사용합니다.

참고:

- Velero는 복원된 백업에 리소스가 없는 경우 velero 복원 리소스에 대한 상태 **PartiallyFailed** 를 설정합니다. 즉, 해당 백업이 비어 있으므로 **restore.cluster.open-cluster-management.io** 리소스는 생성된 **restore.velero.io** 리소스 중 하나가 비어 있으므로 리소스를 복원하지 않는 경우 **부분적으로Failed** 상태에 있을 수 있습니다.
- 새 백업을 사용할 수 있을 때 수동 데이터를 계속 복원하는 데 **syncRestoreWithNewBackups:true** 를 사용하지 않는 한 **restore.cluster.open-cluster-management.io** 리소스가 한 번 실행됩니다. 이 경우 동기화 샘플을 사용하여 수동 복원을 수행합니다. 백업을 확인하는 동안 수동 리소스 복원을 참조하십시오. 복원 작업이 완료되고 동일한 허브 클러스터에서 다른 복원 작업을 실행하려면 새 **restore.cluster.open-cluster-management.io** 리소스를 생성해야 합니다.
- **restore.cluster.open-cluster-management.io** 리소스를 여러 개 생성할 수 있지만 언제든지 하나만 활성화할 수 있습니다.

1.1.5.4. 백업을 확인하는 동안 수동 리소스 복원

restore-passive-sync 샘플을 사용하여 수동 데이터를 복원하면서 새 백업을 사용할 수 있는지 확인하고 자동으로 복원합니다. 새 백업을 자동으로 복원하려면 **syncRestoreWithNewBackups** 매개변수를 **true** 로 설정해야 합니다. 또한 최신 수동 데이터만 복원해야 합니다. 이 섹션의 끝에 샘플 예제를 찾을 수 있습니다.

VeleroResourcesBackupName 및 **VeleroCredentialsBackupName** 매개변수를 **latest** 로 설정하고 **VeleroManagedClustersBackupName** 매개변수를 **건너뛰** 려면 .
VeleroManagedClustersBackupName 이 **latest** 로 설정된 직후 관리 클러스터는 새 허브 클러스터에서 활성화되며 이제 기본 허브 클러스터입니다.

활성화된 관리 클러스터가 기본 허브 클러스터가 되면 복원 리소스가 **Finished** 로 설정되고 **true** 로 설정된 경우에도 **syncRestoreWithNewBackups** 가 무시됩니다.

기본적으로 제어자는 **syncRestoreWithNewBackups** 가 **true** 로 설정된 경우 30분마다 새 백업을 확인합니다. 새 백업이 있으면 백업된 리소스를 복원합니다. **restoreSyncInterval** 매개변수를 업데이트하여 검사 기간을 변경할 수 있습니다.

예를 들어 10분마다 백업을 확인하는 다음 리소스를 참조하십시오.

```
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. 수동 리소스 복원

restore-acm-passive 샘플을 사용하여 hub 클러스터 리소스를 수동 구성으로 복원합니다. 패시브 데이터는 보안, ConfigMap, 애플리케이션, 정책 및 관리되는 모든 클러스터 사용자 정의 리소스와 같은 백업 데이터로, 관리 클러스터와 허브 클러스터 간의 연결을 활성화하지 않습니다. 백업 리소스는 자격 증명 백업 및 복원 리소스를 통해 허브 클러스터에서 복원됩니다.

다음 샘플을 참조하십시오.

```
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. 활성화 리소스 복원

패시브 허브 클러스터에서 활성화 데이터를 복원하기 전에 백업이 생성된 이전 허브 클러스터를 종료합니다. 기본 허브 클러스터가 여전히 실행 중인 경우 이 허브 클러스터에서 실행되는 조정 절차에 따라 더 이상 사용할 수 없는 관리 클러스터에 다시 연결하려고 합니다.

hub 클러스터에서 클러스터를 관리하려는 경우 **restore-acm-passive-activate** 샘플을 사용합니다. 이 경우 다른 데이터가 패시브 리소스를 사용하는 hub 클러스터에서 이미 복원되었다고 가정합니다.

```
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
```

수동 리소스를 복원하는 방법에 따라 활성화 리소스를 복원할 수 있는 몇 가지 옵션이 있습니다.

- **restore-acm-passive-sync cluster.open-cluster-management.io** *리소스에 설명된 대로 백업에서 수동 데이터 복원* 섹션을 사용한 경우 이 리소스에서 **veleroManagedClustersBackupName** 값을 **최신** 상태로 업데이트합니다. 결과적으로 관리 클러스터 리소스 및 **restore-acm-passive-sync** 리소스가 복원됩니다.

- 패시브 리소스를 한 번 작업으로 복원하거나 아직 리소스를 복원하지 않은 경우 모든 리소스 섹션에 지정된 대로 모든 리소스 를 복원하도록 선택합니다.

1.1.5.7. 관리형 클러스터 활성화 데이터 복원

관리되는 클러스터 활성화 데이터 또는 기타 활성화 데이터 리소스는 **cluster.open-cluster-management.io/backup: cluster-activation** 레이블을 사용할 때 관리 클러스터 백업 및 resource-generic 백업에 의해 저장됩니다. 새 허브 클러스터에서 활성화 데이터가 복원되면 관리 클러스터가 복원이 실행되는 허브 클러스터에서 적극적으로 관리되고 있습니다. Operator를 사용하는 방법을 알아보려면 *백업 예약 및 복원을* 참조하십시오.

1.1.5.8. 모든 리소스 복원

모든 데이터를 한 번에 복원하고 hub 클러스터가 한 단계에서 관리 클러스터를 관리하도록 하려면 **restore-acm** 샘플을 사용합니다. hub 클러스터에서 **restore.cluster.open-cluster-management.io** 리소스를 생성한 후 다음 명령을 실행하여 복원 작업의 상태를 가져옵니다.

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

샘플은 다음 리소스와 유사할 수 있습니다.

```
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
```

허브 클러스터에서 백업 파일에 포함된 백업 리소스가 생성되었는지 확인합니다.

1.1.5.9. 가져온 관리 클러스터 복원

Hive API를 사용하여 기본 허브 클러스터와 연결된 관리형 클러스터만 활성화 데이터가 복원되는 새 허브 클러스터와 자동으로 연결됩니다. 이러한 클러스터는 클러스터 탭의 **Create cluster** 버튼을 사용하거나 CLI를 통해 Hive API에서 **기본** 허브 클러스터에 생성되었습니다. 초기 허브 클러스터와 연결된 관리 클러스터는 활성화 데이터가 복원될 때 **Pending Import** (가져오기)로 표시되고 새 허브 클러스터에서 다시 가져와야 합니다.

Hive는 허브 클러스터의 관리 클러스터 네임스페이스에 관리 클러스터 kubeconfig 를 저장하므로 Hive 관리 클러스터는 새 허브 클러스터와 연결할 수 있습니다. 새 hub 클러스터에서 백업 및 복원됩니다. 그런 다음 가져오기 컨트롤러는 Hive API를 사용하여 생성된 관리 클러스터에서만 사용할 수 있는 복원된 구성을 사용하여 관리 클러스터에서 부트스트랩 kubeconfig 를 업데이트합니다. 가져온 클러스터에서는 사용할 수 없습니다.

새 허브 클러스터에서 가져온 클러스터를 다시 연결하려면 복원 작업을 시작한 후 **auto-import-secret** 리소스를 수동으로 생성합니다. 자세한 내용은 *자동 가져오기 보안을 사용하여 클러스터 가져오기*를 참조하십시오.

각 클러스터의 관리 클러스터 네임스페이스에서 가져오기 보류 중 상태의 **auto-import-secret** 리소스를 생성합니다. 가져오기 구성 요소에 충분한 권한이 있는 **kubeconfig** 또는 토큰을 사용하여 새 허브 클러스터에서 자동 가져오기를 시작합니다. 관리 클러스터와 연결하는 데 토큰을 사용하여 각 관리 클러스터에 대한 액세스 권한이 있어야 합니다. 토큰에는 **klusterlet** 역할 바인딩 또는 동일한 권한이 있는 역할이 있어야 합니다.

1.1.5.10. 다른 복원 샘플 사용

다음 Restore 섹션을 보고 YAML 예제를 보고 다양한 유형의 백업 파일을 복원합니다.

- 백업 리소스의 세 가지 유형을 모두 복원합니다.

```
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
```

- 관리형 클러스터 리소스만 복원하십시오.

```
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** 백업을 사용하여 관리 클러스터의 리소스만 복원합니다.

```
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
```

참고:

- **restore.cluster.open-cluster-management.io** 리소스는 한 번 실행됩니다. 복원 작업이 완료되면 선택적으로 동일한 허브 클러스터에서 다른 복원 작업을 실행할 수 있습니다. 새 복원 작업을 실행하려면 새 **restore.cluster.open-cluster-management.io** 리소스를 생성해야 합니다.

- **restore.cluster.open-cluster-management.io** 를 여러 개 생성할 수 있지만 언제든지 하나만 실행할 수 있습니다.

1.1.5.11. 복원 이벤트 보기

다음 명령을 사용하여 복원 이벤트에 대한 정보를 가져옵니다.

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

이벤트 목록은 다음 샘플과 유사할 수 있습니다.

```
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. 추가 리소스

- *DataProtectionApplication* 을 참조하십시오.
- 자동 가져오기 보안이 있는 클러스터 가져오기를참조하십시오.
- 백업 예약 및 복원을 참조하십시오.

1.1.6. 관리 서비스 계정을 사용하여 자동으로 클러스터 연결

백업 컨트롤러는 Managed Service Account 구성 요소를 사용하여 가져온 클러스터를 새 허브 클러스터에 자동으로 연결합니다. 관리형 서비스 계정은 각 관리 클러스터 네임스페이스에서 가져온 각 클러스터에 대해 백업되는 토큰을 생성합니다. 토큰은 **klusterlet-bootstrap-kubeconfig ClusterRole** 바인딩을 사용하므로 자동 가져오기 작업에서 토큰을 사용할 수 있습니다. **klusterlet-bootstrap-kubeconfig ClusterRole** 은 **bootstrap-hub-kubeconfig** 시크릿을 가져오거나 업데이트할 수 있습니다. Managed Service Account 구성 요소에 대한 자세한 내용은 *Managed Service Account란?* 를 참조하십시오.

새 허브 클러스터에서 활성화 데이터가 복원되면 복원 컨트롤러에서 복원 후 작업을 실행하고 가져오기 보류 상태의 모든 관리 클러스터를 찾습니다. 관리 서비스 계정에서 생성한 유효한 토큰이 있는 경우 컨트롤러

는 토큰을 사용하여 자동 가져오기-비밀번호 를 생성합니다. 결과적으로 가져오기 구성 요소는 관리되는 클러스터의 연결을 시도합니다. 클러스터에 액세스할 수 있으면 작업에 성공합니다.

1.1.6.1. 자동 가져오기 활성화

Managed Service Account 구성 요소를 사용하는 자동 가져오기 기능은 기본적으로 비활성화되어 있습니다. 자동 가져오기 기능을 활성화하려면 다음 단계를 완료합니다.

1. **MultiClusterEngine** 리소스에서 **managedserviceaccount enabled** 매개변수를 **true** 로 설정하여 관리 서비스 계정 구성 요소를 활성화합니다. 다음 예제를 참조하십시오.

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

2. **useManagedServiceAccount** 매개변수를 **true** 로 설정하여 **BackupSchedule.cluster.open-cluster-management.io** 리소스에 대한 자동 가져오기 기능을 활성화합니다. 다음 예제를 참조하십시오.

```
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
```

기본 토큰 유효 기간은 **veleroTtl** 의 값을 두 배로 설정하여 전체 라이프사이클에 대해 토큰을 저장하는 모든 백업에 대해 토큰을 유효할 가능성을 늘립니다. 경우에 따라 선택적 **managedServiceAccountTTL** 속성의 값을 설정하여 토큰의 유효 기간을 제어해야 할 수 있습니다.

생성된 토큰의 기본 토큰 만료 시간을 업데이트해야 하는 경우 **managedServiceAccountTTL** 을 사용합니다. 기본값에서 토큰 만료 시간을 변경하면 백업 라이프사이클 중에 만료되도록 토큰이 설정된 백업이 생성될 수 있습니다. 결과적으로 관리 클러스터에서 가져오기 기능이 작동하지 않습니다.

중요: 토큰의 유효 기간을 제어할 필요가 없는 한**managedServiceAccountTTL** 을 사용하지 마십시오.

managedServiceAccountTTL 속성을 사용하려면 다음 예제를 참조하십시오.

```
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
```

자동 가져오기 기능을 활성화하면 백업 구성 요소가 가져온 관리 클러스터 처리를 시작합니다.

- **managed-serviceaccount** 라는 **ManagedServiceAddon** 입니다.
- 이름이 **auto-import-account** 인 **ManagedServiceAccount**
- 관리 클러스터에서 **ManagedServiceAccount** 토큰에 대한 **klusterlet-bootstrap-kubeconfig RoleBinding** 을 설정하는 각 **ManagedServiceAccount** 에 대한 **ManifestWork** 입니다.

토큰은 관리 서비스 계정을 생성할 때 관리 클러스터에 액세스할 수 있는 경우에만 생성됩니다. 그렇지 않으면 관리 클러스터를 사용할 수 있게 되면 나중에 생성됩니다.

1.1.6.2. 자동 가져오기 고려 사항

다음 시나리오를 사용하면 새 허브 클러스터로 이동할 때 관리 클러스터를 자동으로 가져오지 못할 수 있습니다.

- **ManagedServiceAccount** 토큰 없이 허브 백업을 실행하는 경우(예: 관리 클러스터에 액세스할 수 없는 동안 **ManagedServiceAccount** 리소스를 생성하는 경우) 백업에 관리 클러스터를 자동으로 가져오는 토큰이 포함되지 않습니다.
- **auto-import-account** 시크릿 토큰이 유효하고 백업되지만 백업과 함께 사용할 수 있는 토큰이 이미 만료된 경우 자동 가져오기 작업이 실패합니다. **restore.cluster.open-cluster-management.io** 리소스는 각 관리 클러스터에 대한 잘못된 토큰 문제를 보고합니다.
- 복원 시 생성된 **auto-import-secret** 은 **ManagedServiceAccount** 토큰을 사용하여 관리 클러스터에 연결하므로 관리 클러스터에서 kube **apiserver** 정보도 제공해야 합니다. **apiserver** 는 **ManagedCluster** 리소스에 설정해야 합니다. 다음 예제를 참조하십시오.

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

  hub 클러스터에서 클러스터를 가져올 때 **apiserver** 는 OpenShift Container Platform 클러스터에서만 자동으로 설정됩니다. EKS 클러스터와 같은 다른 유형의 관리 클러스터에서 **apiserver** 를 수동으로 설정해야 합니다. 그렇지 않으면 자동 가져오기 기능이 클러스터를 무시합니다. 결과적으로 복원 허브 클러스터로 이동할 때 클러스터는 **Pending** 가져오기 상태로 유지됩니다.
- 백업 일정이 **ManagedServiceAccount** 보안에 설정되기 전에 백업 일정이 실행되는 경우 **ManagedServiceAccount** 시크릿이 백업에 포함되지 않을 수 있습니다. **ManagedServiceAccount** 시크릿에는 생성 시 클러스터 **open-cluster-management.io/backup** 라벨이 설정되어 있지 않습니다. 따라서 백업 컨트롤러는 관리 클러스터의 네임스페이스에서 **ManagedServiceAccount** 시크릿을 정기적으로 검색하고 없는 경우 backup 레이블을 추가합니다.

1.1.6.3. 자동 가져오기 비활성화

BackupSchedule 리소스에서 **useManagedServiceAccount** 매개변수를 **false** 로 설정하여 자동 가져오기 클러스터 기능을 비활성화할 수 있습니다. 다음 예제를 참조하십시오.

```
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
```

기본값은 **false**입니다. 값을 **false** 로 설정한 후 백업 Operator는 **ManagedServiceAddon**,**ManagedServiceAccount**, **ManifestWork** 를 포함하여 생성된 모든 리소스를 제거합니다. 리소스를 제거하면 허브 클러스터 및 관리 클러스터에서 자동 가져오기 토큰이 삭제됩니다.

1.1.6.4. 추가 리소스

- Managed Service Account 구성 요소에 대한 자세한 내용은 Managed Service Account?를 참조하십시오.
- 관리 서비스 계정을 사용하여 자동으로 연결 클러스터로돌아갑니다.

1.1.7. 백업 또는 복원 구성 검증

MultiClusterHub 리소스에서 **cluster-backup** 옵션을 **true** 로 설정하면 다중 클러스터 엔진 Operator가 클러스터 백업을 설치하고 **cluster-backup-chart** 라는 Operator Helm 차트를 복원합니다. 그런 다음 이 차트는 **backup-restore-enabled** 및 **backup-restore-auto-import** 정책을 설치합니다. 이러한 정책을 사용하여 백업 및 복원 구성 요소 관련 문제에 대한 정보를 볼 수 있습니다.

참고: 허브 클러스터는로컬 클러스터로 자동으로 가져오고 자체 관리됩니다.**MultiClusterHub** 리소스에서 **disableHubSelfManagement** 를 **true** 로 설정하여 자체 관리를 비활성화하면**backup-restore-enabled** 정책이 허브 클러스터에 배치되지 않고 정책 템플릿에 보고서가 생성되지 않습니다.

허브 클러스터가 글로벌 허브 클러스터에서 관리되거나 관리 클러스터 인스턴스에 설치된 경우 **disableHubSelfManagement** 를 **true** 로 설정하여 자체 관리 옵션을 비활성화할 수 있습니다. 이 경우 hub 클러스터에서 **backup-restore-enabled** 정책을 활성화할 수 있습니다. 로컬 클러스터를 나타내는 **ManagedCluster** 리소스에 **is-hub=true** 레이블을 설정합니다.

backup-restore-enabled 정책에는 다음 제약 조건을 확인하는 템플릿 세트가 포함됩니다.

- OADP 채널 검증
  - **MultiClusterHub** 에서 백업 구성 요소를 활성화하면 클러스터 백업 및 복원 Operator Helm 차트가 OADP Operator를 설치합니다. **OADP-channel** 템플릿은 설치된 Red Hat OADP Operator 버전이 Red Hat Advanced Cluster Management 클러스터 백업 및 복원 Operator에서 설정한 버전과 일치하는지 확인합니다.
  - 템플릿에는 hub 클러스터에서 설치된 Red Hat OADP Operator를 발견했지만 Red Hat OADP Operator가 Red Hat Advanced Cluster Management 클러스터 백업 및 복원 operator Helm 차트에서 설치한 버전과 일치하지 않는 경우 위반이 표시됩니다. 위반은 클러스

터에서 잘못된 OADP Operator 버전을 찾아 표시합니다. OADP Operator 및 Velero CRD(Custom Resource Definitions)는 클러스터 범위 이므로 동일한 클러스터에 여러 버전의 버전을 설치할 수 없습니다. 대신 올바른 버전만 설치해야 합니다.

  - 다음 예제에서는 backup 및 restore Operator가 잘못된 CRD로 실행되어 잘못된 동작이 발생할 수 있습니다.
    - Red Hat Advanced Cluster Management에는 여러 버전의 OADP가 설치되어 있습니다.
    - **MultiClusterHub** 에서 설치한 OADP 버전이 제거되고 다른 버전을 수동으로 설치하는 경우

- Pod 검증
  다음 템플릿은 백업 구성 요소 및 종속 항목의 Pod 상태를 확인합니다.
  - **ACM-backup-pod-running** 템플릿은 백업 및 복원 Operator Pod가 실행 중인지 확인합니다.
  - OA **DP-pod-running** 템플릿은 OADP Operator Pod가 실행 중인지 확인합니다.
  - **Velero-pod-running** 템플릿은 Velero pod가 실행 중인지 확인합니다.
- 데이터 보호 애플리케이션 검증
  - **data-protection-application-available** 템플릿은 **DataProtectioApplicatio.oadp.openshift.io** 리소스가 생성되었는지 확인합니다. 이 OADP 리소스는 Velero 구성을 설정합니다.
- 백업 스토리지 검증
  - **backup-storage-location-available** 템플릿은 **BackupStorageLocation.velero.io** 리소스가 생성되고 상태 값이 **Available** 인지 확인합니다. 즉, 백업 스토리지에 대한 연결이 유효합니다.
- BackupSchedule 충돌 검증
  - ACM **-backup-clusters-collision-report** 템플릿은 현재 hub 클러스터에 BackupSchedule.cluster.open-cluster-management.io가 있는 경우 **BackupSchedule.cluster.open-cluster-management.io** 상태가 **BackupCollision** 이 아닌지 확인합니다. 이렇게 하면 스토리지 위치에 백업 데이터를 쓸 때 현재 허브 클러스터가 다른 허브 클러스터와 충돌하지 않는지 확인합니다.
    BackupCollision 에 대한 정의는백업 충돌 방지를 참조하십시오.
- BackupSchedule 및 복원 상태 검증
  - ACM **-backup-phase-validation** 템플릿은 현재 클러스터에 **BackupSchedule.cluster.open-cluster-management.io** 가 있는 경우 상태가**Failed** 또는 **Empty** 상태가 아닌지 확인합니다. 이렇게 하면 이 클러스터가 기본 허브 클러스터이고 백업을 생성하는 경우 **BackupSchedule.cluster.open-cluster-management.io** 상태가 정상입니다.
  - 동일한 템플릿이 현재 클러스터에 **Restore.cluster.open-cluster-management.io** 가 있는 경우 상태가 **Failed** 또는 **Empty** 상태가 아닌지 확인합니다. 이렇게 하면 이 클러스터가 보조 허브 클러스터이고 백업을 복원하는 경우 **Restore.cluster.open-cluster-management.io** 상태가 정상입니다.
- 백업이 검증됨
  - ACM **-managed-clusters-schedule-backups-available** 템플릿은 **Backup.velero.io** 리소스에서 지정한 위치에서 **Backup.velero.io** 리소스를 사용할 수 있는지, **BackupSchedule.cluster.open-cluster-management.io** 리소스에서 백업이 생성되는지 확

인합니다. 이렇게 하면 백업 및 복원 연산자를 사용하여 백업이 한 번 이상 실행되었는지 확인합니다.

- 완료를 위한 백업
  - **acm-backup-in-progress-report** 템플릿은 **Backup.velero.io** 리소스가 **InProgress** 상태에 있는지 확인합니다. 이 검증은 많은 리소스를 사용하면 백업이 실행되면 velero Pod가 다시 시작되고 백업이 완료되지 않고 진행 중인 상태로 유지되기 때문입니다. 일반 백업 중에 백업 리소스가 실행될 때 특정 시점에서 백업 리소스가 진행 중이지만 중단되지 않고 완료되도록 실행됩니다. **acm-backup-in-progress-report** 템플릿은 일정이 실행되는 동안 경고를 보고하고 백업이 진행 중인 것을 확인하는 것이 정상입니다.
- cron 작업으로 적극적으로 실행되는 백업
  - **BackupSchedule.cluster.open-cluster-management.io** 를 적극적으로 실행하고 스토리지 위치에 새 백업을 저장합니다. 이 검증은 **backup-schedule-cron-enabled** 정책 템플릿에서 수행합니다. 템플릿은 스토리지 위치에 **velero.io/schedule-name: acm-validation-policy-schedule** 라벨이 있는 **Backup.velero.io** 가 있는지 확인합니다.
  - 백업 cron 일정에 시간이 설정된 후 **acm-validation-policy-schedule** 백업은 만료되도록 설정됩니다. 백업을 생성하기 위해 cron 작업이 실행되지 않으면 만료된 이전 **acm-validation-policy-schedule** 백업이 삭제되고 새 작업이 생성되지 않습니다. 결과적으로 언제든지**acm-validation-policy-schedule** 백업이 없는 경우 백업을 생성하는 활성 cron 작업이 없음을 의미합니다.
  - 이 정책은 허브 클러스터가 활성 상태이고 백업을 생성하거나 복원할 때 허브 클러스터 관리자에게 백업 문제를 알리는 데 도움이 됩니다.

backup-restore-auto-import 정책에는 다음 제약 조건을 확인하는 템플릿 세트가 포함됩니다.

- 자동 가져오기 보안 검증
  - **auto-import-account-secret** 템플릿은 **local-cluster** 이외의 관리 클러스터 네임스페이스에 **ManagedServiceAccount** 시크릿이 생성되었는지 확인합니다. 백업 컨트롤러는 가져온 관리 클러스터를 정기적으로 검사합니다. 관리 클러스터가 검색되면 백업 컨트롤러에서 관리 클러스터 네임스페이스에 **ManagedServiceAccount** 리소스를 생성합니다. 이 프로세스는 관리 클러스터에서 토큰 생성을 시작합니다. 그러나 이 작업 시 관리 클러스터에 액세스할 수 없는 경우 **ManagedServiceAccount** 는 토큰을 생성할 수 없습니다. 예를 들어 관리 클러스터가 손상되면 토큰을 생성할 수 없습니다. 따라서 이 기간 동안 허브 백업을 실행하면 백업에 관리 클러스터를 자동 가져오기 위한 토큰이 없습니다.
- 자동 가져오기 백업 라벨 검증
  - **auto-import-backup-label** 템플릿은 **local-cluster** 이외의 관리 클러스터 네임스페이스에 **ManagedServiceAccount** 시크릿이 있는지 확인합니다. 템플릿에서 **ManagedServiceAccount** 시크릿을 찾으면 템플릿에서 시크릿에**cluster.open-cluster-management.io/backup** 레이블을 적용합니다. 이 레이블은 Red Hat Advanced Cluster Management 백업에 **ManagedServiceAccount** 시크릿을 포함하는 데 중요합니다.

1.1.7.1. 서버 측 암호화를 사용하여 데이터 보호

서버 측 암호화는 스토리지 위치에서 데이터를 수신하는 애플리케이션 또는 서비스의 데이터 암호화입니다. 백업 메커니즘 자체는 전송 중 데이터(백업 스토리지 위치로 이동 중) 또는 미사용(백업 스토리지 위치에 있는 디스크에 저장됨)을 암호화하지 않습니다. 대신 오브젝트 및 스냅샷 시스템의 기본 메커니즘을 사용합니다.

모범 사례: 사용 가능한 백업 스토리지 서버 측 암호화를 사용하여 대상에서 데이터를 암호화합니다. 백업에는 허브 클러스터 외부에 저장할 때 암호화해야 하는 자격 증명 및 구성 파일과 같은 리소스가 포함되어 있습니다.

serverSideEncryption 및 **kmsKeyId** 매개변수를 사용하여 Amazon S3에 저장된 백업의 암호화를 활성화할 수 있습니다. 자세한 내용은 *Backup Storage Location YAML* 을 참조하십시오. 다음 샘플은 **DataProtectionApplication** 리소스를 설정할 때 AWS KMS 키 ID를 지정합니다.

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

*기타 스토리지 공급자의 구성 가능한 모든 매개 변수를 알아보려면 Velero 지원*스토리지 공급자를 참조하십시오.

1.1.7.2. 추가 리소스

- Backup Storage 위치 YAML을 참조하십시오.
- Velero 지원 스토리지 공급자를 참조하십시오.
- 백업 검증 또는 복원 구성으로돌아갑니다.

1.1.8. 고급 구성 백업 및 복원

다음 섹션을 확인하여 백업 및 복원을 추가로 구성할 수 있습니다.

1.1.8.1. 리소스 요청 및 사용자 지정 제한

Velero가 처음 설치되면 Velero pod는 다음 샘플에 정의된 대로 기본 CPU 및 메모리 제한으로 설정됩니다.

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

이전 샘플의 제한은 일부 시나리오에서 잘 작동하지만 클러스터가 많은 리소스를 백업할 때 업데이트해야 할 수 있습니다. 예를 들어 2000 클러스터를 관리하는 허브 클러스터에서 백업이 실행되면 OOM(메모리 부족 오류)으로 인해 Velero Pod가 실패합니다. 다음 구성을 사용하면 이 시나리오에 대해 백업을 완료할 수 있습니다.

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

Velero pod 리소스에 대한 제한 및 요청을 업데이트하려면 **DataProtectionApplication** 리소스를 업데이트하고 Velero pod에 대한 **resourceAllocation** 템플릿을 삽입해야 합니다. 다음 샘플을 확인합니다.

```
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. 추가 리소스

- **DataProtectionApplication** 매개변수에 대한 자세한 내용은 Red Hat OpenShift Container Platform 설명서의 Default Velero 클라우드 공급자 플러그인 주제를 참조하십시오.
- 클러스터 사용량을 기반으로 하는 CPU 및 메모리 요구 사항에 대한 자세한 내용은 OpenShift Container Platform 설명서의 구성에 대한 CPU 및 메모리 요구 사항을 참조하십시오.

1.2. VXLANSYNC 영구 볼륨 복제 서비스

VXLANSync는 클러스터 내에서 또는 복제와 달리 호환되지 않는 스토리지 유형이 있는 클러스터 전체에서 비동기식 볼륨을 복제할 수 있는 Kubernetes 운영자입니다. CSI(Container Storage Interface)를 사용하여 호환성 제한을 해결합니다. 사용 중인 환경에 ScalaSync Operator를 배포한 후 이를 활용하여 영구 데이터의 복사본을 생성하고 유지 관리할 수 있습니다. ScalaSync는 버전 4.13 이상인 Red Hat OpenShift Container Platform 클러스터에서만 영구 볼륨 클레임을 복제할 수 있습니다.

중요: volSync는 **Filesystem** 의 **volumeMode** 가 있는 영구 볼륨 클레임만 지원합니다.**volumeMode** 를 선택하지 않으면 기본값은 **Filesystem** 입니다.

- volSync를 사용하여 영구 볼륨 복제
  - 관리형 클러스터에volSync 설치
  - Rsync-TLS 복제 구성
  - Rsync 복제 구성
  - restic 백업 구성
  - Rclone 복제 구성
- 복제된 이미지를 사용 가능한 영구 볼륨 클레임으로 변환

- 동기화 예약

1.2.1. volSync를 사용하여 영구 볼륨 복제

rsync, rsync-tls, restic 또는 Rclone과 같은 동기화 위치 수에 따라volSync로 영구 볼륨을 복제하는 세 가지 지원 방법을 사용할 수 있습니다.

1.2.1.1. 사전 요구 사항

EgressSync를 클러스터에 설치하기 전에 다음과 같은 요구 사항이 있어야 합니다.

- Red Hat Advanced Cluster Management 버전 2.10 이상 허브 클러스터를 실행하는 구성된 Red Hat OpenShift Container Platform 환경
- 동일한 Red Hat Advanced Cluster Management Hub 클러스터에서 관리하는 두 개 이상의 구성된 클러스터
- volSync로 구성 중인 클러스터 간 네트워크 연결. 클러스터가 동일한 네트워크에 없는 경우 Submariner 다중 클러스터 네트워킹 및 서비스 검색을구성하고 **ServiceType** 의 **ClusterIP** 값을 클러스터를 네트워크에 사용하거나 **ServiceType** 의 **LoadBalancer** 값이 있는 로드 밸런서를 사용할 수 있습니다.
- 소스 영구 볼륨에 사용하는 스토리지 드라이버는 CSI와 호환되어야 하며 스냅샷을 지원할 수 있어야 합니다.

1.2.1.2. 관리형 클러스터에volSync 설치

EgressSync가 한 클러스터에 있는 영구 볼륨 클레임을 다른 클러스터의 영구 볼륨 클레임에 복제하려면 소스 및 대상 관리 클러스터 둘 다에volSync를 설치해야 합니다.

VXLANSync는 자체 네임스페이스를 생성하지 않으므로 다른 OpenShift Container Platform all-namespace Operator와 동일한 네임스페이스에 있습니다. volSync의 운영자 설정을 변경하면 채널 업데이트에 대한 수동 승인을 변경하는 경우와 같이 동일한 네임스페이스의 다른 Operator에도 영향을 미칩니다.

두 가지 방법 중 하나를 사용하여 환경에 있는 두 개의 클러스터에 ChronySync를 설치할 수 있습니다. hub 클러스터의 각 관리 클러스터에 레이블을 추가하거나 다음 섹션에 설명된 대로 수동으로 **ManagedClusterAddOn** 을 생성하고 적용할 수 있습니다.

1.2.1.2.1. 라벨을 사용하여volSync 설치

레이블을 추가하여 관리 클러스터에volSync를 설치하려면 다음을 수행합니다.

- Red Hat Advanced Cluster Management 콘솔에서 다음 단계를 완료합니다.
  1. 허브 클러스터 콘솔의 클러스터 페이지에서 관리 클러스터 중 하나를 선택하여 세부 정보를 확인합니다.
  2. 라벨 필드에 다음 레이블을 추가합니다.

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

     volSync 서비스 Pod가 관리 클러스터에 설치되어 있습니다.
  3. 다른 관리 클러스터를 동일한 레이블을 추가합니다.

4. 각 관리 클러스터에서 다음 명령을 실행하여volSync Operator가 설치되었는지 확인합니다.

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

SkySync가 설치될 때 Operator가 나열되어 있습니다.

- 명령줄 인터페이스에서 다음 단계를 완료합니다.

1. hub 클러스터에서 명령줄 세션을 시작합니다.
2. 다음 명령을 입력하여 첫 번째 클러스터에 라벨을 추가합니다.

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

managed-cluster-1 을 관리 클러스터 중 하나의 이름으로 교체합니다.

3. 다음 명령을 입력하여 두 번째 클러스터에 레이블을 추가합니다.

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

managed-cluster-2 를 다른 관리 클러스터의 이름으로 교체합니다.

ManagedClusterAddOn 리소스는 해당 관리 클러스터의 네임스페이스에 있는 허브 클러스터에 자동으로 생성해야 합니다.

1.2.1.2.2. ManagedClusterAddOn을 사용하여volSync 설치

ManagedClusterAddOn 을 수동으로 추가하여 관리 클러스터에volSync를 설치하려면 다음 단계를 완료하십시오.

1. hub 클러스터에서 다음 예와 유사한 콘텐츠가 포함된 **volsync-mcao.yaml** 이라는 YAML 파일을 생성합니다.

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

managed-cluster-1-namespace 를 관리 클러스터 중 하나의 네임스페이스로 교체합니다. 이 네임스페이스는 관리 클러스터의 이름과 동일합니다.

참고: 이름은 **volsync** 여야 합니다.

2. 다음 예와 유사한 명령을 입력하여 구성에 파일을 적용합니다.

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

3. 다른 관리 클러스터에 대해 절차를 반복합니다.
ManagedClusterAddOn 리소스는 해당 관리 클러스터의 네임스페이스에 있는 허브 클러스터에 자동으로 생성해야 합니다.

1.2.1.2.3. collectdSync *ManagedClusterAddOn*업데이트

사용 중인 Red Hat Advanced Cluster Management 버전에 따라volSync 버전을 업데이트해야 할 수 있습니다. volSync **ManagedClusterAddOn** 리소스를 업데이트하려면 다음 단계를 완료하십시오.

1. **ManagedClusterAddOn** 리소스에 다음 주석을 추가합니다.

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

2. ChronylySync를 배포할 **operator-subscription-channel** 을 정의합니다.

3. **ManagedClusterAddOn** 리소스로 이동하여 선택한**operator-subscription-channel** 이 포함되어 있는지 확인하여volsync 버전을 업데이트했는지 확인합니다.

1.2.1.3. Rsync-TLS 복제 구성

Rsync-TLS 복제를 사용하여 영구 볼륨의 1:1 비동기 복제를 생성할 수 있습니다. 재해 복구 또는 원격 사이트로 데이터를 전송하는 데 Rsync-TLS 기반 복제를 사용할 수 있습니다. Rsync-TLS를 사용하는 경우 CryostatSync는 stunnel에서 제공하는 TLS 보호 터널에서 Rsync를 사용하여 데이터를 동기화합니다. 자세한 내용은 stunnel 설명서 를 참조하십시오.

다음 예제에서는 Rsync-TLS 메서드를 사용하여 구성하는 방법을 보여줍니다. Rsync-TLS에 대한 자세한 내용은 reflectSync 설명서의 사용을 참조하십시오.
https://volsync.readthedocs.io/en/latest/usage/index.html

1.2.1.3.1. 관리 클러스터에서 Rsync-TLS 복제 구성

Rsync-TLS 기반 복제의 경우 소스 및 대상 클러스터에서 사용자 지정 리소스를 구성합니다. 사용자 지정 리소스는 **address** 값을 사용하여 소스를 대상에 연결하고 stunnel에서 제공하는 TLS 보호 터널을 사용하여 전송된 데이터가 안전한지 확인합니다.

source -ns 네임스페이스의 소스 클러스터의 영구 볼륨 클레임에서 **destination -ns** 네임스페이스의 대상 클러스터의 영구 볼륨 클레임으로 Rsync-TLS 복제를 구성하려면 다음 정보 및 예를 참조하십시오. 필요한 경우 값을 바꿉니다.

1. 대상 클러스터를 구성합니다.

   a. 대상 클러스터에서 다음 명령을 실행하여 네임스페이스를 생성합니다.

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

   destination-ns 를 복제 대상이 있는 네임스페이스로 바꿉니다.

   b. **replication_destination** 이라는 새 YAML 파일을 생성하고 다음 콘텐츠를 복사합니다.

```
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) 이 예에서는 **LoadBalancer** 의 **ServiceType** 값이 사용됩니다. 로드 밸런서 서비스는 소스 관리 클러스터가 다른 대상 관리 클러스터로 정보를 전송할 수 있도록 소스 클러스터에 의해 생성됩니다. 소스 및 대상이 동일한 클러스터에 있거나 Submariner 네트워크 서비스가 구성된 경우 **ClusterIP** 를 서비스 유형으로 사용할 수 있습니다. 소스 클러스터를 구성할 때 참조할 주소 및 시크릿 이름을 확인합니다. 용량 값이 복제 중인 영구 볼륨 클레임의 용량과 일치하는지 확인합니다.

(2) 용량 값이 복제 중인 영구 볼륨 클레임의 용량과 일치하는지 확인합니다.

선택 사항: 환경의 기본값과 다른 스토리지 클래스 및 볼륨 스냅샷 클래스 이름을 사용하는 경우 **storageClassName** 및 **volumeSnapshotClassName** 매개변수 값을 지정합니다.

c. 대상 클러스터에서 다음 명령을 실행하여 **replicationdestination** 리소스를 생성합니다.

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

destination-ns 를 대상이 있는 네임스페이스의 이름으로 교체합니다.

replicationdestination 리소스가 생성되면 다음 매개변수와 값이 리소스에 추가됩니다.

매개변수	현재의
.status.rsyncTLS.address	소스 및 대상 클러스터가 통신할 수 있도록 하는 데 사용되는 대상 클러스터의 IP 주소입니다.
.status.rsyncTLS.keySecret	소스 클러스터와의 연결을 인증하는 TLS 키가 포함된 시크릿의 이름입니다.

d. 다음 명령을 실행하여 소스 클러스터에서 사용할 **.status.rsyncTLS.address** 값을 복사합니다. **destination** 을 복제 대상 사용자 정의 리소스의 이름으로 교체합니다.**destination-ns** 를 대상이 있는 네임스페이스 이름으로 교체합니다.

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

출력은 Amazon Web Services 환경에 대한 다음과 유사합니다.

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

e. 다음 명령을 실행하여 보안 이름을 복사합니다.

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

destination 을 복제 대상 사용자 정의 리소스의 이름으로 교체합니다.

destination-ns 를 대상이 있는 네임스페이스의 이름으로 교체합니다.

소스를 구성할 때 소스 클러스터에 입력해야 합니다. 출력은 다음 이름과 유사할 수 있는 SSH 키 시크릿 파일의 이름이어야 합니다.

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

f. 대상 클러스터에 대해 다음 명령을 입력하여 대상 클러스터에서 키 시크릿을 복사합니다.

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

destination-ns 를 복제 대상이 있는 네임스페이스로 바꿉니다.

g. 다음 명령을 입력하여 **vi** 편집기에서 시크릿 파일을 엽니다.

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

h. 대상 클러스터의 열린 시크릿 파일에서 다음과 같이 변경합니다.

- 네임스페이스를 소스 클러스터의 네임스페이스로 변경합니다. 이 예제에서는 **source-ns** 입니다.
- 소유자 참조(**.metadata.ownerReferences**)를 제거합니다.

i. 소스 클러스터에서 소스 클러스터에 다음 명령을 입력하여 시크릿 파일을 생성합니다.

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

2. 복제하려는 소스 영구 볼륨 클레임을 식별합니다.
참고: 소스 영구 볼륨 클레임은 CSI 스토리지 클래스에 있어야 합니다.

3. **ReplicationSource** 항목을 만듭니다.

a. 소스 클러스터에서 **replication_source** 라는 새 YAML 파일을 생성하고 다음 콘텐츠를 복사합니다.

```
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** 를 복제 소스 사용자 정의 리소스의 이름으로 교체합니다. 이 작업을 자동으로 교체하는 방법에 대한 지침은 이 절차의 *3vi* 단계를 참조하십시오.

2 **source-ns** 를 소스가 있는 영구 볼륨 클레임의 네임스페이스로 교체합니다. 이 작업을 자동으로 교체하는 방법에 대한 지침은 이 절차의 *3vi* 단계를 참조하십시오.

3 **persistent_volume_claim** 을 소스 영구 볼륨 클레임의 이름으로 교체합니다.

4 **mykeysecret** 을 대상 클러스터에서 소스 클러스터(**$KEYSECRET**)로 복사한 시크릿 이름으로 교체합니다.

5 **my.host.com** 을 구성할 때 **ReplicationDestination** 의 **.status.rsyncTLS.address** 필드에서 복사한 호스트 주소로 바꿉니다. 다음 단계에서 **sed** 명령의 예를 찾을 수 있습니다.

스토리지 드라이버가 복제를 지원하는 경우 **copyMethod** 의 값으로 **Clone** 을 사용하면 복제에 더 간소화된 프로세스가 될 수 있습니다.

선택 사항: 환경의 기본값과 다른 스토리지 클래스 및 볼륨 스냅샷 클래스 이름을 사용하는 경우 **storageClassName** 및 **volumeSnapshotClassName** 매개변수 값을 지정합니다.

이제 영구 볼륨의 동기화 방법을 설정할 수 있습니다.

b. 소스 클러스터에서 다음 명령을 입력하여 **ReplicationSource** 개체의 **address** 및 **keySecret** 값을 대상 클러스터에서 기록한 값으로 교체하여 **replication_source.yaml** 파일을 수정합니다.

```
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 을 구성할 때 **ReplicationDestination** 의 **.status.rsyncTLS.address** 필드에서 복사한 호스트 주소로 바꿉니다.

keySecret 을 구성할 때 **ReplicationDestination** 의 **.status.rsyncTLS.keySecret** 필드에서 복사한 키로 교체합니다.

source 를 소스가 있는 영구 볼륨 클레임의 이름으로 교체합니다.

참고: 복제하려는 영구 볼륨 클레임과 동일한 네임스페이스에 파일을 생성해야 합니다.

c. **ReplicationSource** 개체에서 다음 명령을 실행하여 복제가 완료되었는지 확인합니다.

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

source-ns 를 소스가 있는 영구 볼륨 클레임의 네임스페이스로 교체합니다.

source 를 복제 소스 사용자 정의 리소스의 이름으로 교체합니다.

복제에 성공한 경우 출력은 다음 예와 유사해야 합니다.

```
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
```

마지막 동기화 시간 목록에 시간이 없으면 복제가 완료되지 않습니다.

원래 영구 볼륨 클레임의 복제본이 있습니다.

1.2.1.4. Rsync 복제 구성

중요: 보안을 강화하려면 Rsync 대신 Rsync-TLS를 사용합니다. Rsync-TLS를 사용하면 영구 볼륨을 복제하는 데 필요하지 않은 승격된 사용자 권한을 사용하지 않을 수 있습니다.

Rsync 복제를 사용하여 영구 볼륨의 1:1 비동기 복제를 생성할 수 있습니다. 재해 복구 또는 원격 사이트로 데이터를 전송하는 데 Rsync 기반 복제를 사용할 수 있습니다.

다음 예제에서는 Rsync 메서드를 사용하여 구성하는 방법을 보여줍니다.

1.2.1.4.1. 관리 클러스터에서 Rsync 복제 구성

Rsync 기반 복제의 경우 소스 및 대상 클러스터에서 사용자 지정 리소스를 구성합니다. 사용자 정의 리소스는 **address** 값을 사용하여 소스를 대상에 연결하고**sshKeys** 를 사용하여 전송된 데이터가 안전한지 확인합니다.

참고: 주소 및 **sshKeys** 의 값을 대상에서 소스로 복사해야 하므로 소스를 구성하기 전에 대상을 구성합니다.

이 예제에서는 **source-ns** 네임스페이스의 소스 클러스터의 영구 볼륨 클레임에서 **destination -ns** 네임스페이스의 대상 클러스터의 영구 볼륨 클레임으로 Rsync 복제를 구성하는 단계를 제공합니다. 필요한 경우 해당 값을 다른 값으로 교체할 수 있습니다.

1. 대상 클러스터를 구성합니다.
   a. 대상 클러스터에서 다음 명령을 실행하여 네임스페이스를 생성합니다.

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

   destination-ns 를 대상 영구 볼륨 클레임을 포함할 네임스페이스의 이름으로 교체합니다.

   b. 다음 YAML 콘텐츠를 복사하여 **replication_destination.yaml** 이라는 새 파일을 생성합니다.

   ```
   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
```

참고: 용량 값은 복제 중인 영구 볼륨 클레임의 용량과 일치해야 합니다.

destination 을 복제 대상 CR의 이름으로 교체합니다.

destination-ns 를 대상이 있는 네임스페이스의 이름으로 교체합니다.

이 예에서는 **LoadBalancer** 의 **ServiceType** 값이 사용됩니다. 로드 밸런서 서비스는 소스 관리 클러스터가 다른 대상 관리 클러스터로 정보를 전송할 수 있도록 소스 클러스터에 의해 생성됩니다. 소스 및 대상이 동일한 클러스터에 있거나 Submariner 네트워크 서비스가 구성된 경우 **ClusterIP** 를 서비스 유형으로 사용할 수 있습니다. 소스 클러스터를 구성할 때 참조할 주소와 시크릿 이름을 기록해 둡니다.

storageClassName 및 **volumeSnapshotClassName** 은 선택적 매개변수입니다. 특히 환경의 기본값과 다른 스토리지 클래스 및 볼륨 스냅샷 클래스 이름을 사용하는 경우 환경의 값을 지정합니다.

c. 대상 클러스터에서 다음 명령을 실행하여 **replicationdestination** 리소스를 생성합니다.

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

destination-ns 를 대상이 있는 네임스페이스의 이름으로 교체합니다.

replicationdestination 리소스가 생성되면 다음 매개변수와 값이 리소스에 추가됩니다.

매개변수	현재의
.status.rsync.address	소스 및 대상 클러스터가 통신할 수 있도록 하는 데 사용되는 대상 클러스터의 IP 주소입니다.
.status.rsync.sshKeys	소스 클러스터에서 대상 클러스터로 데이터 전송을 가능하게 하는 SSH 키 파일의 이름입니다.

d. 다음 명령을 실행하여 소스 클러스터에서 사용할 **.status.rsync.address** 값을 복사합니다.

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

destination 을 복제 대상 사용자 정의 리소스의 이름으로 교체합니다.

destination-ns 를 대상이 있는 네임스페이스의 이름으로 교체합니다.

출력은 Amazon Web Services 환경에 대한 다음 출력과 유사해야 합니다.

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

e. 다음 명령을 실행하여 보안 이름을 복사합니다.

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

destination 을 복제 대상 사용자 정의 리소스의 이름으로 교체합니다.

destination-ns 를 대상이 있는 네임스페이스의 이름으로 교체합니다.

소스를 구성할 때 소스 클러스터에 입력해야 합니다. 출력은 다음 이름과 유사할 수 있는 SSH 키 시크릿 파일의 이름이어야 합니다.

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

f. 대상 클러스터에 대해 다음 명령을 입력하여 대상 클러스터에서 SSH 시크릿을 복사합니다.

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

destination-ns 를 대상이 있는 영구 볼륨 클레임의 네임스페이스로 바꿉니다.

g. 다음 명령을 입력하여 **vi** 편집기에서 시크릿 파일을 엽니다.

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

h. 대상 클러스터의 열린 시크릿 파일에서 다음과 같이 변경합니다.

- 네임스페이스를 소스 클러스터의 네임스페이스로 변경합니다. 이 예제에서는 **source-ns** 입니다.
- 소유자 참조(**.metadata.ownerReferences**)를 제거합니다.

i. 소스 클러스터에서 소스 클러스터에 다음 명령을 입력하여 시크릿 파일을 생성합니다.

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

2. 복제하려는 소스 영구 볼륨 클레임을 식별합니다.
참고: 소스 영구 볼륨 클레임은 CSI 스토리지 클래스에 있어야 합니다.

3. **ReplicationSource** 항목을 만듭니다.

a. 다음 YAML 콘텐츠를 복사하여 소스 클러스터에 **replication_source.yaml** 이라는 새 파일을 생성합니다.

```
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 를 복제 소스 사용자 정의 리소스의 이름으로 교체합니다. 이 작업을 자동으로 교체하는 방법에 대한 지침은 이 절차의 *3vi* 단계를 참조하십시오.

source-ns 를 소스가 있는 영구 볼륨 클레임의 네임스페이스로 교체합니다. 이 작업을 자동으로 교체하는 방법에 대한 지침은 이 절차의 *3vi* 단계를 참조하십시오.

persistent_volume_claim 을 소스 영구 볼륨 클레임의 이름으로 교체합니다.

mysshkeys 를 구성할 때 **ReplicationDestination** 필드의 **.status.rsync.sshKeys** 필드에서 복사한 키로 바꿉니다.

my.host.com 을 구성할 때 **ReplicationDestination** 의 **.status.rsync.address** 필드에서 복사한 호스트 주소로 바꿉니다.

스토리지 드라이버가 복제를 지원하는 경우 **copyMethod** 의 값으로 **Clone** 을 사용하면 복제에 더 간소화된 프로세스가 될 수 있습니다.

storageClassName 및 **volumeSnapshotClassName** 은 선택적 매개변수입니다. 환경의 기본값과 다른 스토리지 클래스 및 볼륨 스냅샷 클래스 이름을 사용하는 경우 해당 값을 지정합니다.

이제 영구 볼륨의 동기화 방법을 설정할 수 있습니다.

b. 소스 클러스터에서 다음 명령을 입력하여 **ReplicationSource** 개체의 주소 와 **sshKeys** 값을 대상 클러스터에서 기록한 값으로 교체하여 **replication_source.yaml** 파일을 수정합니다.

```
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 을 구성할 때 **ReplicationDestination** 의 **.status.rsync.address** 필드에서 복사한 호스트 주소로 바꿉니다.

mysshkeys 를 구성할 때 **ReplicationDestination** 필드의 **.status.rsync.sshKeys** 필드에서 복사한 키로 바꿉니다.

source 를 소스가 있는 영구 볼륨 클레임의 이름으로 교체합니다.

참고: 복제하려는 영구 볼륨 클레임과 동일한 네임스페이스에 파일을 생성해야 합니다.

c. **ReplicationSource** 개체에서 다음 명령을 실행하여 복제가 완료되었는지 확인합니다.

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

source-ns 를 소스가 있는 영구 볼륨 클레임의 네임스페이스로 교체합니다.

source 를 복제 소스 사용자 정의 리소스의 이름으로 교체합니다.

복제에 성공한 경우 출력은 다음 예와 유사해야 합니다.

```
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
```

마지막 동기화 시간 목록에 시간이 없으면 복제가 완료되지 않습니다.

원래 영구 볼륨 클레임의 복제본이 있습니다.

1.2.1.5. restic 백업 구성

restic 기반 백업은 영구 볼륨의 restic 기반 백업 사본을 **restic-config.yaml** 시크릿 파일에 지정된 위치에 복사합니다. restic 백업은 클러스터 간에 데이터를 동기화하지 않지만 데이터 백업을 제공합니다.

restic 기반 백업을 구성하려면 다음 단계를 완료합니다.

1. 다음 YAML 콘텐츠와 유사한 보안을 생성하여 백업 이미지가 저장된 리포지토리를 지정합니다.

```
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 를 백업 파일을 저장하려는 S3 버킷 리포지토리의 위치로 바꿉니다.

my-restic-password 를 리포지토리에 액세스하는 데 필요한 암호화 키로 교체합니다.

필요한 경우 액세스 및 암호를 공급자의 인증 정보로 교체합니다.

새 리포지토리를 준비해야 하는 경우 절차를 위한 새 리포지토리 준비를 참조하십시오. 이 절차를 사용하는 경우 리포지토리를 초기화하기 위해 **restic init** 명령을 실행하는 데 필요한 단계를 건너뜁니다. volSync는 첫 번째 백업 중에 리포지토리를 자동으로 초기화합니다.

중요: 동일한 S3 버킷에 여러 영구 볼륨 클레임을 백업할 때 버킷의 경로는 각 영구 볼륨 클레임에 대해 고유해야 합니다. 각 영구 볼륨 클레임은 별도의 **ReplicationSource** 로 백업되며 각각 별도의 restic-config 시크릿이 필요합니다.

동일한 S3 버킷을 공유함으로써 각 **ReplicationSource** 는 전체 S3 버킷에 대한 쓰기 액세스 권한을 갖습니다.

2. 다음 YAML 콘텐츠와 유사한 **ReplicationSource** 오브젝트를 생성하여 백업 정책을 구성합니다.

```
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
```

소스 를 백업 중인 영구 볼륨 클레임으로 교체합니다.

schedule 값을 백업을 실행하는 빈도로 바꿉니다. 이 예제에는 30분마다 일정이 있습니다.일정 설정에 대한 자세한 내용은 동기화 예약을 참조하십시오.

PruneIntervalDays 값을 공간을 절약하기 위해 데이터를 다시 패키징하는 인스턴스 간에 경과한 일 수로 바꿉니다. 정리 작업은 실행 중 중요한 I/O 트래픽을 생성할 수 있습니다.

restic-config 를 1단계에서 생성한 시크릿 이름으로 교체합니다.

백업된 이미지의 보존 정책에 적용되는 값을 설정합니다.

모범 사례: **CopyMethod** 값으로 **Clone** 을 사용하여 지정 시간 이미지가 저장되도록 합니다.

참고: **Restic movers**는 기본적으로 **root** 권한 없이 실행됩니다. **restic movers**를 **root**로 실행하려면 다음 명령을 실행하여 상승된 권한 주석을 네임스페이스에 추가합니다.

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

& lt;namespace >를 네임스페이스 이름으로 바꿉니다.

1.2.1.5.1. restic backup 복원

복사된 데이터를 **restic** 백업에서 새 영구 볼륨 클레임으로 복원할 수 있습니다. 모범 사례: 하나의 백업만 새 영구 볼륨 클레임으로 복원합니다. **restic** 백업을 복원하려면 다음 단계를 완료합니다.

1. 다음 예와 유사한 새 데이터를 포함하도록 새 영구 볼륨 클레임을 생성합니다.

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

pvc-name 을 새 영구 볼륨 클레임의 이름으로 교체합니다.

2. 다음 예제와 유사한 **ReplicationDestination** 사용자 지정 리소스를 생성하여 데이터를 복원할 위치를 지정합니다.

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

destination 을 복제 대상 **CR**의 이름으로 교체합니다.

restic-repo 를 소스가 저장된 리포지토리의 경로로 바꿉니다.

pvc-name 을 데이터를 복원하려는 새 영구 볼륨 클레임의 이름으로 교체합니다. 새 영구 볼륨 클레임을 프로비저닝하는 대신 기존 영구 볼륨 클레임을 사용합니다.

복원 프로세스는 한 번만 완료해야 하며 이 예에서는 최신 백업을 복원합니다. 복원 옵션에 대한 자세

한 내용은volSync 설명서의 복원 옵션을 참조하십시오.

1.2.1.6. Rclone 복제 구성

Rclone 백업은 AWS S3와 같은 중간 오브젝트 스토리지 위치를 통해 Rclone을 사용하여 단일 영구 볼륨을 여러 위치에 복사합니다. 데이터를 여러 위치에 배포할 때 유용할 수 있습니다.

Rclone 복제를 구성하려면 다음 단계를 완료합니다.

1. 다음 예와 유사한 ReplicationSource 사용자 지정 리소스를 만듭니다.

```
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 를 복제 소스 사용자 정의 리소스의 이름으로 교체합니다.

source-ns 를 소스가 있는 영구 볼륨 클레임의 네임스페이스로 교체합니다.

source 를 복제 중인 영구 볼륨 클레임으로 교체합니다.

schedule 값을 복제를 실행하는 빈도로 바꿉니다. 이 예제에는 6분마다 일정이 있습니다. 이 값은 따옴표 내에 있어야 합니다. 자세한 내용은 동기화 예약을 참조하십시오.

intermediate-s3-bucket 을 Rclone 구성 파일의 구성 섹션 경로로 교체합니다.

destination-bucket 을 복제된 파일이 복사하려는 오브젝트 버킷의 경로로 바꿉니다.

rclone-secret 을 **Rclone** 구성 정보가 포함된 보안 이름으로 교체합니다.

copyMethod 의 값을 **Clone,Direct** 또는 **Snapshot** 으로 설정합니다. 이 값은 특정 시점 복사본이 생성되는지 여부를 지정하고, 이 경우 해당 복사본을 생성하는 데 사용되는 방법을 지정합니다.

my-sc-name 을 **point-in-time copy**에 사용하려는 스토리지 클래스의 이름으로 바꿉니다. 지정하지 않으면 소스 볼륨의 스토리지 클래스가 사용됩니다.

Snapshot 을 **copyMethod** 로 지정한 경우 사용할 **VolumeSnapshotClass** 의 이름으로 **my-vsc** 를 바꿉니다. 다른 유형의 **copyMethod** 에는 필요하지 않습니다.

2. 다음 예와 유사한 **ReplicationDestination** 사용자 정의 리소스를 만듭니다.

```
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>
```

schedule 값을 대상으로 복제를 이동하는 빈도로 바꿉니다. 대상에서 가져오기 전에 데이터가 복제를 완료할 수 있도록 소스 및 대상에 대한 스케줄이 오프셋되어야 합니다. 이 예제에는 **6**분마다 일정이 있으며 **3**분으로 오프셋됩니다. 이 값은 따옴표 내에 있어야 합니다. 스케줄링에 대한 자세한 내용은 동기화 예약을 참조하십시오.

intermediate-s3-bucket 을 **Rclone** 구성 파일의 구성 섹션 경로로 교체합니다.

destination-bucket 을 복제된 파일이 복사하려는 오브젝트 버킷의 경로로 바꿉니다.

rclone-secret 을 **Rclone** 구성 정보가 포함된 보안 이름으로 교체합니다.

copyMethod 의 값을 **Clone,Direct** 또는 **Snapshot** 으로 설정합니다. 이 값은 특정 시점 복사본이 생성되는지 여부와 이 경우 해당 복사본을 생성하는 데 사용되는 방법을 지정합니다.

accessModes 의 값은 영구 볼륨 클레임에 대한 액세스 모드를 지정합니다. 유효한 값은 **ReadWriteOnce** 또는 **ReadWriteMany** 입니다.

용량 은 대상 볼륨의 크기를 지정합니다. 이 크기는 들어오는 데이터를 포함할 수 있을 만큼 커야 합니다.

my-sc 를 시점 복사의 대상으로 사용할 스토리지 클래스의 이름으로 바꿉니다. 지정하지 않으면 시스템 스토리지 클래스가 사용됩니다.

Snapshot 을 **copyMethod** 로 지정한 경우 사용할 **VolumeSnapshotClass** 의 이름으로 **my-vsc** 를 바꿉니다. 다른 유형의 **copyMethod** 에는 필요하지 않습니다. 포함되지 않은 경우 시스템 기본 **VolumeSnapshotClass** 가 사용됩니다.

참고: **Rclone** 이동기는 기본적으로 **root** 권한 없이 실행됩니다. **Rclone movers**를 **root**로 실행하려면 다음 명령을 실행하여 네임스페이스에 승격된 권한 주석을 추가합니다.

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

& lt;namespace >를 네임스페이스 이름으로 바꿉니다.

1.2.1.7. 추가 리소스

자세한 내용은 다음 항목을 참조하십시오.

- **Rsync-TLS** 복제에 대한 고유 시크릿을 생성하는 방법을 알아보려면 **Rsync-TLS** 복제에 대한 시크릿 생성 을 참조하십시오.

- Rsync에 대한 자세한 내용은 ScalaSync 설명서의 사용을 참조하십시오. https://volsync.readthedocs.io/en/latest/usage/index.html

- restic 옵션에 대한 자세한 내용은 ScalaSync 설명서의 백업 옵션을 참조하십시오.

- 관리 클러스터에서volSync 설치로돌아가기

1.2.2. 복제된 이미지를 사용 가능한 영구 볼륨 클레임으로 변환

데이터를 복구하려면 복제된 이미지를 영구 볼륨 클레임으로 변환해야 할 수 있습니다.

VolumeSnapshot 를 사용하여 ReplicationDestination 위치에서 영구 볼륨 클레임을 복제하거나 복원할 때 VolumeSnapshot 가 생성됩니다. VolumeSnapshot 에는 마지막 성공적인 동기화의 latestImage 가 포함되어 있습니다. 이미지 사본을 사용하려면 먼저 영구 볼륨 클레임으로 변환해야 합니다. EgressSync ReplicationDestination 볼륨 팝업을 사용하여 이미지 사본을 사용 가능한 영구 볼륨 클레임으로 변환할 수 있습니다.

1. 영구 볼륨 클레임을 복원하려는 ReplicationDestination 을 가리키는 dataSourceRef 를 사용하여 영구 볼륨 클레임을 생성합니다. 이 영구 볼륨 클레임은 ReplicationDestination 사용자 정의 리소스 정의의 status.latestImage 설정에 지정된 VolumeSnapshot 콘텐츠로 채워집니다.

   다음 YAML 콘텐츠는 사용할 수 있는 샘플 영구 볼륨 클레임을 보여줍니다.

```
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 을 새 영구 볼륨 클레임의 이름으로 교체합니다.

destination-ns 를 영구 볼륨 클레임 및 **ReplicationDestination** 이 있는 네임스페이스로 교체합니다.

replicationdestination_to_replace 를 **ReplicationDestination** 이름으로 교체합니다.

모범 사례: 값이 초기 소스 영구 볼륨 클레임과 동일한 크기인 경우 **resources.requests.storage** 를 다른 값으로 업데이트할 수 있습니다.

2. 다음 명령을 입력하여 영구 볼륨 클레임이 환경에서 실행 중인지 확인합니다.

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

참고:

latestImage 가 없는 경우 영구 볼륨 클레임은 **ReplicationDestination** 이 완료되고 스냅샷을 사용할 수 있을 때까지 보류 상태로 유지됩니다. **ReplicationDestination** 및 **ReplicationDestination** 을 사용하는 영구 볼륨 컨트롤러를 동시에 생성할 수 있습니다. 영구 볼륨 클레임은 **ReplicationDestination** 이 복제를 완료하고 스냅샷을 사용할 수 있는 경우에만 볼륨 채우기 프로세스를 시작합니다. **.status.latestImage** 에서 스냅샷을 찾을 수 있습니다.

또한 사용되는 스토리지 클래스에 **WaitForFirstConsumer** 의 **volumeBindingMode** 값이 있는 경우 볼륨 팝업기는 채우기 전에 영구 볼륨 클레임의 소비자가 있을 때까지 기다립니다. 소비자가 영구 볼륨 클레임을 마운트하려는 **Pod**와 같이 액세스가 필요한 경우 볼륨이 채워집니다. **EgressSync** 볼륨 팝업 컨트롤러는 **ReplicationDestination** 의 **latestImage** 를 사용합니다. 영구 볼륨 제어가 생성된 후 복제가 완료될 때마다 **latestImage** 가 업데이트됩니다.

1.2.3. 동기화 예약

복제 시작 방법(항상 실행, 일정 또는 수동으로)을 결정할 때 세 가지 옵션 중에서 선택합니다. 복제 예약은 종종 선택한 옵션입니다.

스케줄 옵션은 예약된 시간에 복제를 실행합니다. 일정은 **cronspec** 로 정의되므로 일정을 시간 간격 또는 특정 시간으로 구성할 수 있습니다. 스케줄 값의 순서는 다음과 같습니다.

"분 (0-59) 시간 (0-59) 개월 (1-31) 개월 (1-12) 일 (0-59) 시간 (0-59))

예약된 시간이 발생하면 복제가 시작됩니다. 이 복제 옵션에 대한 설정은 다음 내용과 유사할 수 있습니다.

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

이러한 방법 중 하나를 활성화한 후 구성한 방법에 따라 동기화 일정이 실행됩니다.

자세한 내용 및 옵션은 **reflectSync** 설명서를 참조하십시오.

1.2.4. VXLANSync 고급 구성

자체 시크릿 생성과 같은 영구 볼륨을 복제할 때volSync를 추가로 구성할 수 있습니다.

1.2.4.1. Rsync-TLS 복제에 대한 시크릿 생성

소스 및 대상은 TLS 연결의 공유 키에 대한 액세스 권한이 있어야 합니다. keySecret 필드에서 키 위치를 찾을 수 있습니다. .spec.rsyncTLS.keySecret 에 시크릿 이름을 제공하지 않으면 시크릿 이름이 자동으로 생성되고 .status.rsyncTLS.keySecret 에 추가됩니다.

자체 보안을 생성하려면 다음 단계를 완료합니다.

1. 시크릿에 다음 형식을 사용합니다. < id>:<at_least_32_hex_digits>

   다음 예: 1:23b7395fafc3e842bd8ac0fe142e6ad1

2. 이전 예에 해당하는 다음 secret.yaml 예제를 참조하십시오.

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