Red Hat Advanced Cluster Management for Kubernetes 2.5

클러스터

클라우드 공급자 전반에서 클러스터를 생성, 가져오기 및 관리하는 방법을 알아보려면 자세히 알아보십시오.

Last Updated: 2023-09-18

Red Hat Advanced Cluster Management for Kubernetes 2.5 클러스터

클라우드 공급자 전반에서 클러스터를 생성, 가져오기 및 관리하는 방법을 알아보려면 자세히 알아보십시오.

법적 공지

초록

클라우드 공급자 전반에서 클러스터를 생성, 가져오기 및 관리하는 방법을 알아보려면 자세히 알아보십시오.

차례

1장. 클러스터 관리

Kubernetes 콘솔용 Red Hat Advanced Cluster Management를 사용하여 클라우드 공급자 전반에서 클러스터를 생성, 가져오기 및 관리하는 방법을 알아보십시오. 다음 주제에서 공급자 간에 클러스터를 관리하는 방법을 알아보십시오.

- 지원되는 공급자
- 관리형 클러스터 스케일링
- 이미지 릴리스
- 베어 메탈 자산 생성 및 수정
- 인프라 환경 생성
- 인증 정보 관리 개요
- 클러스터 생성
- 대상 관리 클러스터를 허브 클러스터로 가져오기
- 프록시 환경에서 클러스터 생성
- 클러스터 프록시 애드온 활성화
- 특정 클러스터 관리 역할 구성
- 클러스터 라벨 관리
- ManagedClusterSets 생성 및 관리 (기술 프리뷰)
- 배치와 함께 ManagedClusterSets 사용
- 클러스터 풀 관리 (기술 프리뷰)
- 관리 클러스터에서 실행되도록 Ansible Tower 작업 구성
- 클러스터 풀에서 클러스터 요청
- 호스트된 컨트롤 플레인 클러스터 사용 (기술 프리뷰)
- Discovery 소개
- 클러스터 업그레이드
- 관리에서 클러스터 제거
- 클러스터 백업 및 복원 Operator

1.1. 클러스터 라이프사이클 아키텍처

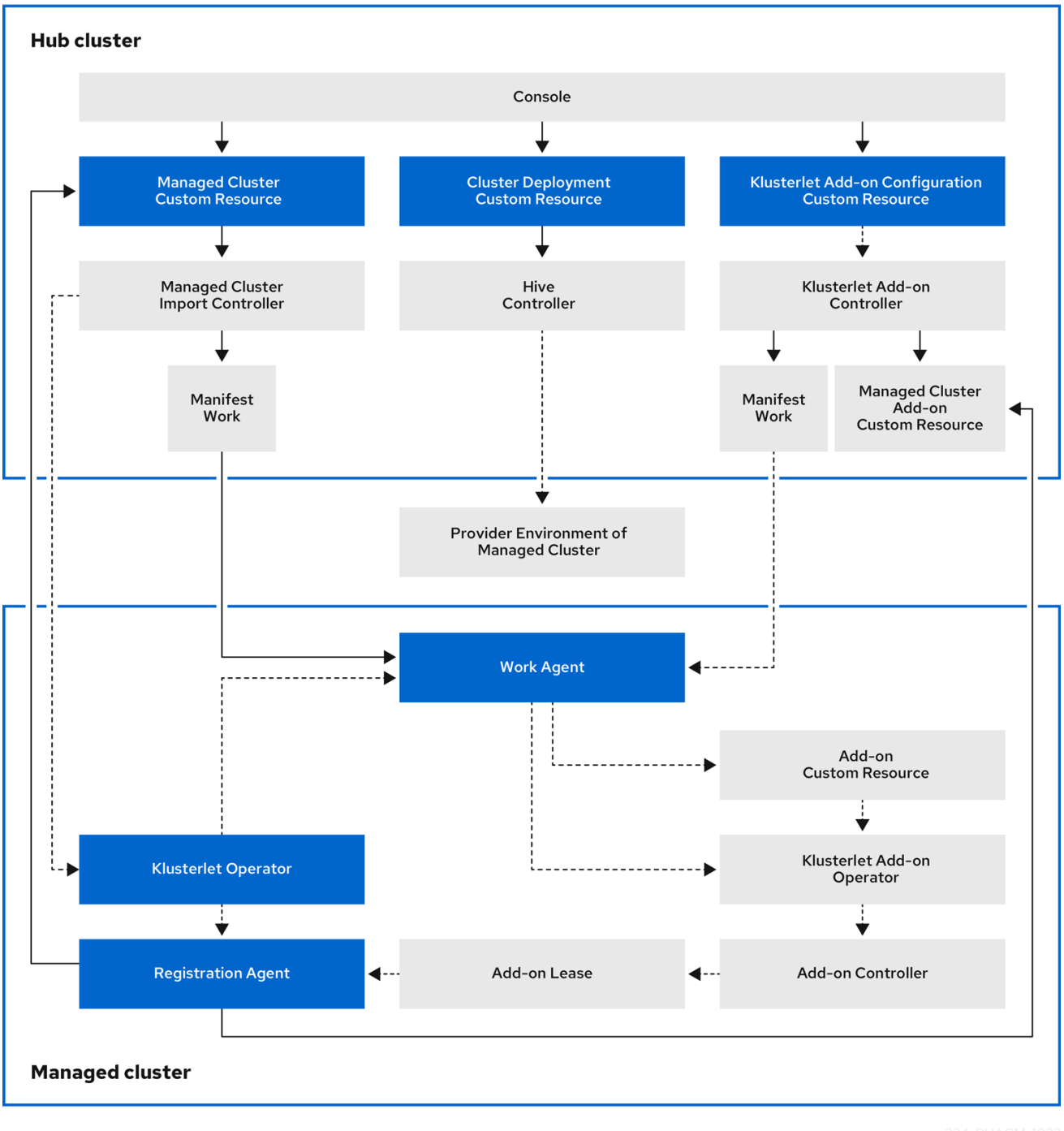
Red Hat Advanced Cluster Management for Kubernetes에는 *허브 클러스터와 관리형 클러스터의 두 가지 주요 유형이 있습니다.*

hub 클러스터는 Red Hat Advanced Cluster Management for Kubernetes가 설치된 기본 클러스터입니다. 허브 클러스터를 사용하여 다른 Kubernetes 클러스터를 생성, 관리 및 모니터링할 수 있습니다.

관리형 클러스터는 hub 클러스터에서 관리하는 Kubernetes 클러스터입니다. Red Hat Advanced Cluster Management hub 클러스터를 사용하여 일부 클러스터를 생성할 수 있지만 허브 클러스터에서 관리할 기존 클러스터를 가져올 수도 있습니다.

Red Hat Advanced Cluster Management를 사용하여 관리형 클러스터를 생성하면 Hive 리소스가 있는 Red Hat OpenShift Container Platform 클러스터 설치 프로그램을 사용하여 클러스터가 생성됩니다. OpenShift Container Platform 설치 개요는 OpenShift Container Platform 설명서에서 확인하여 OpenShift Container Platform 설치 프로그램으로 클러스터 설치 프로세스에 대한 자세한 내용을 확인할 수 있습니다.

다음 다이어그램은 클러스터 관리를 위해 Red Hat Advanced Cluster Management와 함께 설치된 구성 요소를 보여줍니다.



클러스터 라이프사이클 관리 아키텍처의 구성 요소에는 다음 항목이 포함됩니다.

hub 클러스터의 구성 요소:

- 콘솔: Red Hat Advanced Cluster Management 관리 클러스터의 클러스터 라이프사이클을 관리할 수 있는 웹 기반 인터페이스를 제공합니다.
- Hive Controller: Red Hat Advanced Cluster Management로 생성한 클러스터를 프로비저닝합니다. 또한 Hive 컨트롤러는 Red Hat Advanced Cluster Management에서 생성한 관리형 클러스터를 분리하고 삭제합니다.
- 관리형 클러스터 가져오기 컨트롤러: klusterlet Operator를 관리형 클러스터에 배포합니다.
- Klusterlet Add-on Controller: klusterlet add-on Operator를 관리 클러스터에 배포합니다.

관리형 클러스터의 구성 요소:

- Klusterlet Operator: 관리 클러스터에 등록 및 작업 컨트롤러를 배포합니다.
- 등록 에이전트: hub 클러스터와 관리 클러스터를 등록합니다. 관리 클러스터가 hub 클러스터에 액세스할 수 있도록 다음 권한이 자동으로 생성됩니다.
  - ClusterRole
    - 에이전트가 인증서를 순환할 수 있도록 허용
    - 에이전트가 hub 클러스터에서 관리하는 클러스터를 **/list/update/watch**
    - **에이전트가 허브 클러스터에서 관리하는 클러스터의 상태를 업데이트할 수 있습니다.**
  - **hub 클러스터의 hub 클러스터 네임스페이스에서 생성된 역할**
    - **관리형 클러스터 등록 에이전트가 조정.k8s.io 리스를 가져오 거나 업데이트할 수 있습니다.**
    - **에이전트가 관리되는 클러스터 애드온 을/list/watch 할 수 있도록 허용**
    - **에이전트가 관리 클러스터 애드온의 상태를 업데이트할 수 있도록 허용**
- **작업 에이전트: 매니페스트 작업을 관리 클러스터에 적용합니다. 관리 클러스터가 hub 클러스터에 액세스할 수 있도록 다음 권한이 자동으로 생성됩니다.**
  - **hub 클러스터의 hub 클러스터 네임스페이스에서 생성된 역할**

- 작업 에이전트에서 허브 클러스터로 이벤트를 보낼 수 있음
- 에이전트가 **/list/watch/update the manifestworks** 리소스를 가져올 수있도록 허용
- 에이전트가 **manifestworks** 리소스의 상태를 업데이트할 수 있도록 허용

1.2. 관리형 클러스터 스케일링 (기술 프리뷰)

Red Hat Advanced Cluster Management에서 생성한 클러스터의 경우 가상 머신 크기 및 노드 수와 같은 관리형 클러스터 사양을 사용자 지정하고 조정할 수 있습니다. 다른 공급자에서 가져온 관리형 클러스터를 확장하려면 공급자 관리 클러스터 스케일링을 참조하십시오.

기술 프리뷰: **Kubernetes**용 **Red Hat Advanced Cluster Management**에서 관리하는 많은 클러스터는 **Red Hat Advanced Cluster Management** 콘솔 또는 명령줄과 **MachinePool** 리소스를 사용하여 확장할 수 있습니다.

- **MachinePool** 리소스를 사용하는 것은 **Red Hat Advanced Cluster Management**에서 생성한 베어 메탈 클러스터에서 지원되지 않는 기능입니다.
- **MachinePool** 리소스는 관리 클러스터에서 **MachineSet** 리소스를 그룹화하는 **hub** 클러스터의 **Kubernetes** 리소스입니다.
- **MachinePool** 리소스는 영역 구성, 인스턴스 유형, 루트 스토리지를 포함하여 머신 리소스 집합을 균일하게 구성합니다.
- **MachinePool** 을 사용하면 원하는 노드 수를 수동으로 구성하거나 관리형 클러스터에서 노드의 자동 스케일링을 구성할 수 있습니다.

1.2.1. 자동 확장

자동 스케일링을 구성하면 트래픽이 부족할 때 리소스 비용을 줄이고 리소스에 대한 수요가 증가할 때 리소스가 충분한지 확인하기 위해 필요에 따라 확장할 수 있는 클러스터의 유연성을 제공합니다.

1.2.1.1. 자동 스케일링 활성화

- **Red Hat Advanced Cluster Management** 콘솔을 사용하여 **MachinePool** 리소스에서 자동 스케일링을 활성화하려면 다음 단계를 완료하십시오.

  1. **Red Hat Advanced Cluster Management** 탐색에서 **인프라 > 클러스터**를 선택합니다.

  2. 대상 클러스터의 이름을 클릭하고 *머신 풀* 탭을 선택합니다.

  3. 시스템 풀 페이지의 대상 시스템 풀의 *옵션* 메뉴에서 **자동 스케일링 활성화**를 선택합니다.

  4. 최소 및 최대 머신 세트 복제본 수를 선택합니다. 머신 세트 복제본은 클러스터의 노드에 직접 매핑됩니다.

     스케일 을 클릭한 후 콘솔에 반영하는 데 몇 분이 걸릴 수 있습니다. *머신 풀 탭 알림이 있는 경우 시스템 보기* 를 클릭하여 스케일링 작업의 상태를 볼 수 있습니다.

- 명령줄을 사용하여 **MachinePool** 리소스에서 자동 스케일링을 활성화하려면 다음 단계를 완료합니다.

  1. 다음 명령을 입력하여 머신 풀 목록을 확인합니다.

     ```
     oc get machinepools -n <managed-cluster-namespace>
     ```

     managed-cluster-namespace 를 대상 관리 클러스터의 네임스페이스로 교체합니다.

  2. 다음 명령을 입력하여 머신 풀의 **YAML** 파일을 편집합니다.

     ```
     oc edit machinepool <name-of-MachinePool-resource> -n <namespace-of-managed-cluster>
     ```

     name-of-MachinePool-resource 를 **MachinePool** 리소스의 이름으로 바꿉니다.

     namespace-of-managed-cluster 를 관리 클러스터의 네임스페이스 이름으로 교체합

니다.

3. YAML 파일에서 **spec.replicas** 필드를 삭제합니다.

4. **spec.autoscaling.minReplicas** 설정 및 **spec.autoscaling.maxReplicas** 필드를 리소스 YAML에 추가합니다.

5. **minReplicas** 설정에 최소 복제본 수를 추가합니다.

6. 최대 복제본 수를 **maxReplicas** 설정에 추가합니다.

7. 파일을 저장하여 변경 사항을 제출합니다.

머신 풀에 자동 스케일링이 활성화됩니다.

1.2.1.2. 자동 스케일링 비활성화

콘솔 또는 명령줄을 사용하여 자동 스케일링을 비활성화할 수 있습니다.

- **Red Hat Advanced Cluster Management** 콘솔을 사용하여 자동 스케일링을 비활성화하려면 다음 단계를 완료하십시오.

  1. **Red Hat Advanced Cluster Management** 탐색에서 **인프라 > 클러스터**를 선택합니다.

  2. 대상 클러스터의 이름을 클릭하고 *머신 풀* 탭을 선택합니다.

  3. 시스템 풀 페이지의 대상 시스템 풀의 ***Options*** 메뉴에서 **Disable autoscale** 을 선택합니다.

  4. 원하는 머신 세트 복제본 수를 선택합니다. 머신 세트 복제본은 클러스터의 노드와 직접 매핑됩니다.

스케일 을 클릭한 후 콘솔에 표시되는 데 몇 분이 걸릴 수 있습니다. 머신 풀 탭의 알림에서 머신 보기 를 클릭하여 스케일링 상태를 볼 수 있습니다.

- 명령줄을 사용하여 자동 스케일링을 비활성화하려면 다음 단계를 완료합니다.

  1. 다음 명령을 입력하여 머신 풀 목록을 확인합니다.

     ```
     oc get machinepools -n <managed-cluster-namespace>
     ```

     managed-cluster-namespace 를 대상 관리 클러스터의 네임스페이스로 교체합니다.

  2. 다음 명령을 입력하여 머신 풀의 **YAML** 파일을 편집합니다.

     ```
     oc edit machinepool <name-of-MachinePool-resource> -n <namespace-of-managed-cluster>
     ```

     name-of-MachinePool-resource 를 **MachinePool** 리소스의 이름으로 바꿉니다.

     namespace-of-managed-cluster 를 관리 클러스터의 네임스페이스 이름으로 교체합니다.

  3. **YAML** 파일에서 **spec.autoscaling** 필드를 삭제합니다.

  4. **spec.replicas** 필드를 리소스 **YAML**에 추가합니다.

  5. **replicas** 설정에 복제본 수를 추가합니다.

  6. 파일을 저장하여 변경 사항을 제출합니다.

자동 스케일링이 비활성화되어 있습니다.

1.2.2. 클러스터 수동 스케일링

클러스터 자동 스케일링을 활성화하지 않으려면 **Red Hat Advanced Cluster Management** 콘솔 또는 명령줄을 사용하여 클러스터가 유지 관리하려는 복제본의 정적 수를 변경할 수 있습니다. 필요에 따라 크기를 늘리거나 줄이는 데 도움이 될 수 있습니다.

- **Red Hat Advanced Cluster Management** 콘솔을 사용하여 **MachinePool** 리소스를 수동으로 확장하려면 다음 단계를 완료하십시오.

  1. **Red Hat Advanced Cluster Management** 탐색에서 **인프라 > 클러스터**를 선택합니다.
  2. 대상 클러스터의 이름을 클릭하고 *머신 풀* 탭을 선택합니다.

     참고: ***Autoscale*** 필드의 값이 활성화되어 있는 경우 계속하기 전에 자동 스케일링 비활성화 단계를 완료하여 먼저 자동 스케일링 기능을 비활성화 해야 합니다.
  3. 시스템 풀의 *옵션* 메뉴에서 스케일 머신 풀(**Scale machine pool**)을 선택합니다.
  4. 머신 풀 크기를 조정하도록 머신 세트 복제본 수를 조정합니다.

- 명령줄을 사용하여 **MachinePool** 리소스를 확장하려면 다음 단계를 완료합니다.

  1. 다음 명령을 입력하여 머신 풀 목록을 확인합니다.

     ```
     oc get machinepools -n <managed-cluster-namespace>
     ```

     managed-cluster-namespace 를 대상 관리 클러스터의 네임스페이스로 교체합니다.
  2. 다음 명령을 입력하여 머신 풀의 **YAML** 파일을 편집합니다.

     ```
     oc edit machinepool <name-of-MachinePool-resource> -n <namespace-of-managed-cluster>
     ```

     name-of-MachinePool-resource 를 **MachinePool** 리소스의 이름으로 바꿉니다.

namespace-of-managed-cluster 를 관리 클러스터의 네임스페이스 이름으로 교체합니다.

3. YAML의 spec.replicas 구성을 복제본 수로 업데이트합니다.

4. 파일을 저장하여 변경 사항을 제출합니다.

참고: 가져오기 관리형 클러스터에는 Red Hat Advanced Cluster Management에서 생성한 클러스터와 동일한 리소스가 없습니다. 따라서 클러스터를 확장하는 절차가 다릅니다. 가져온 클러스터의 클러스터 확장 방법에 대한 정보가 포함된 공급자의 제품 설명서를 참조하십시오.

예를 들어 권장 클러스터 스케일링 관행 및 사용 중인 버전에 적용되는 OpenShift Container Platform 설명서에서 MachineSet을 수동으로 스케일링 할 수 있습니다.

1.3. 이미지 릴리스

Kubernetes용 Red Hat Advanced Cluster Management를 사용하여 공급자에 클러스터를 생성할 때 새 클러스터에 사용할 릴리스 이미지를 지정해야 합니다. 릴리스 이미지는 클러스터를 빌드하는 데 사용되는 Red Hat OpenShift Container Platform 버전을 지정합니다.

릴리스 이미지를 참조하는 파일은 acm-hive-openshift-releases GitHub 리포지토리에서 유지 관리되는 YAML 파일입니다. Red Hat Advanced Cluster Management는 해당 파일을 사용하여 콘솔에서 사용 가능한 릴리스 이미지 목록을 생성합니다. 여기에는 OpenShift Container Platform의 최신 빠른 채널 이미지가 포함됩니다. 콘솔은 최신 OpenShift Container Platform의 최신 버전용 최신 릴리스 이미지만 표시합니다. 예를 들어 콘솔 옵션에 다음 릴리스 이미지가 표시될 수 있습니다.

- quay.io/openshift-release-dev/ocp-release:4.6.23-x86_64
- quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64

참고: console에서 클러스터를 생성할 때: visible: 'true' 레이블이 있는 이미지만 선택할 수 있습니다. ClusterImageSet 리소스의 이 레이블 예는 다음 콘텐츠에서 제공됩니다.

```
apiVersion: config.openshift.io/v1
kind: ClusterImageSet
metadata:
```

```
  labels:
    channel: fast
    visible: 'true'
  name: img4.10.1-x86-64-appsub
spec:
  releaseImage: quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64
```

추가 릴리스 이미지가 저장되지만 콘솔에는 표시되지 않습니다. 사용 가능한 릴리스 이미지를 모두 보려면 **CLI**에서 **kubectl get clusterimageset** 를 실행합니다. 최신 릴리스 이미지로 클러스터를 생성하도록 최신 버전만 콘솔에 있습니다. 경우에 따라 특정 버전인 클러스터를 생성해야 할 수도 있습니다. 따라서 이전 버전을 사용할 수 있습니다. **Red Hat Advanced Cluster Management**는 해당 파일을 사용하여 콘솔에서 사용 가능한 릴리스 이미지 목록을 생성합니다. 여기에는 **OpenShift Container Platform**의 최신 빠른 채널 이미지가 포함됩니다.

리포지토리에는 릴리스 이미지로 작업할 때 사용하는 디렉터리인 **clusterImageSets** 디렉터리와 서브스크립션 디렉터리가 포함되어 있습니다.

clusterImageSets 디렉터리에는 다음 디렉터리가 포함되어 있습니다.

- **fast:** 지원되는 각 **OpenShift Container Platform** 버전의 최신 릴리스 이미지를 참조하는 파일이 포함되어 있습니다. 이 폴더의 릴리스 이미지는 테스트, 확인 및 지원됩니다.
- 릴리스**:** 각 **OpenShift Container Platform** 버전(테이블, 빠른, 후보 채널)의 모든 릴리스 이미지를 참조하는 파일이 포함되어 있습니다. 이러한 릴리스는 모두 테스트되고 안정적인 것으로 확인되지 않았습니다.
- **stable:** 지원되는 각 **OpenShift Container Platform** 버전에 대한 릴리스 이미지의 최신 두 가지 안정적인 버전을 참조하는 파일이 포함되어 있습니다.

참고**:** 기본적으로 현재 릴리스 이미지 목록은 한 시간씩 업데이트됩니다. 제품을 업그레이드한 후 새 버전의 제품에 권장되는 릴리스 이미지 버전을 반영하는 데 최대 **1**시간이 걸릴 수 있습니다.

다음과 같은 세 가지 방법으로 자체 **ClusterImageSets** 를 큐레이션할 수 있습니다.

세 가지 방법 중 첫 번째 단계는 포함된 서브스크립션을 비활성화하여 최신 빠른 채널 이미지를 자동으로 업데이트하는 것입니다. **multiclusterhub** 리소스에서 **installer** 매개변수를 사용하여 최신 **fast ClusterImageSets** 의 자동 큐레이션을 비활성화할 수 있습니다. **spec.disableUpdateClusterImageSets** 매개변수를 **true** 와 **false** 로 전환하면 **Red Hat Advanced Cluster Management**와 함께 설치된 서브스크립션이 각각 비활성화 또는 활성화됩니다. 자체 이미지를

큐레이팅하려면 **spec.disableUpdateClusterImageSets** 를 **true** 로 설정하여 서브스크립션을 비활성화합니다.

옵션 **1:** 클러스터를 생성할 때 콘솔에서 사용할 특정 **ClusterImageSet** 의 이미지 참조를 지정합니다. 지정한 각 새 항목은 **persists**이며 향후 모든 클러스터 프로비저닝에 사용할 수 있습니다. 항목의 예는 **quay.io/openshift-release-dev/ocp-release:4.6.8-x86_64** 입니다.

옵션 **2: acm-hive-openshift-releases GitHub** 리포지토리에서 **ClusterImageSets YAML** 파일을 수동으로 생성하고 적용합니다.

옵션 **3: acm-hive-openshift-releases GitHub** 리포지토리의 **README.md** 를 따라 분기된 **GitHub** 리포지토리에서 **ClusterImageSets** 자동 업데이트를 활성화합니다.

Subscription 디렉터리에는 릴리스 이미지 목록을 가져올 위치를 지정하는 파일이 포함되어 있습니다.

Red Hat Advanced Cluster Management의 기본 릴리스 이미지는 **Quay.io** 디렉터리에 제공됩니다.

이미지는 릴리스 2.5의 acm-hive-openshift-releases GitHub 리포지토리에 있는 파일에서 참조합니다.

1.3.1. 다른 아키텍처에 클러스터를 배포할 릴리스 이미지 생성

두 아키텍처의 파일이 포함된 릴리스 이미지를 수동으로 생성하여 허브 클러스터의 아키텍처와 다른 아키텍처에 클러스터를 생성할 수 있습니다.

예를 들어 **ppc64le,aarch64** 또는 **s390x** 아키텍처에서 실행 중인 허브 클러스터에서 **x86_64** 클러스터를 생성해야 할 수 있습니다. 새 릴리스 이미지를 사용하면 **OpenShift Container Platform** 릴리스 레지스트리에서 다중 아키텍처 이미지 매니페스트를 제공할 수 있으므로 두 파일 세트로 릴리스 이미지를 생성하면 클러스터 생성에 성공합니다.

릴리스 이미지를 생성하려면 아키텍처 유형에 대한 다음 예제와 유사한 단계를 완료합니다.

1. OpenShift Container Platform 릴리스 레지스트리에서 **x86_64,s390x,aarch64** 및 **ppc64le** 릴리스 이미지가 포함된 매니페스트 목록을 생성합니다.

a. 다음 예제 명령을 사용하여 Quay 리포지토리에서 해당 환경의 두 아키텍처에 대한 매니페스트 목록을 가져옵니다.

```
podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64
podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-ppc64le
podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-s390x
podman pull quay.io/openshift-release-dev/ocp-release:4.10.1-aarch64
```

b. 이미지를 유지보수하는 프라이빗 리포지토리에 로그인합니다.

```
podman login <private-repo>
```

private-repo 를 리포지토리 경로로 바꿉니다.

c. 환경에 적용되는 다음 명령을 실행하여 프라이빗 리포지토리에 릴리스 이미지 매니페스트를 추가합니다.

```
podman push quay.io/openshift-release-dev/ocp-release:4.10.1-x86_64 <private-repo>/ocp-release:4.10.1-x86_64
podman push quay.io/openshift-release-dev/ocp-release:4.10.1-ppc64le <private-repo>/ocp-release:4.10.1-ppc64le
podman push quay.io/openshift-release-dev/ocp-release:4.10.1-s390x <private-repo>/ocp-release:4.10.1-s390x
podman push quay.io/openshift-release-dev/ocp-release:4.10.1-aarch64 <private-repo>/ocp-release:4.10.1-aarch64
```

private-repo 를 리포지토리 경로로 바꿉니다.

d. 새 정보에 대한 매니페스트를 생성합니다.

```
podman manifest create mymanifest
```

e. 두 릴리스 이미지에 대한 참조를 매니페스트 목록에 추가합니다.

```
podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-x86_64
podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-ppc64le
podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-s390x
podman manifest add mymanifest <private-repo>/ocp-release:4.10.1-aarch64
```

private-repo 를 리포지토리 경로로 바꿉니다.

f. 매니페스트 목록의 목록을 기존 매니페스트와 병합합니다.

```
podman manifest push mymanifest docker://<private-repo>/ocp-release:4.10.1
```

private-repo 를 리포지토리 경로로 바꿉니다.

2. **hub** 클러스터에서 리포지터리의 매니페스트를 참조하는 릴리스 이미지를 만듭니다.

a. 다음 예와 유사한 정보가 포함된 **YAML** 파일을 생성합니다.

```
apiVersion: hive.openshift.io/v1
kind: ClusterImageSet
metadata:
  labels:
    channel: fast
    visible: "true"
  name: img4.10.1-appsub
spec:
  releaseImage: <private-repo>/ocp-release:4.10.1
```

private-repo 를 리포지토리 경로로 바꿉니다.

b. **hub** 클러스터에서 다음 명령을 실행하여 변경 사항을 적용합니다.

```
oc apply -f <file-name>.yaml
```

file-name 을 방금 생성한 **YAML** 파일의 이름으로 바꿉니다.

3. **OpenShift Container Platform** 클러스터를 생성할 때 새 릴리스 이미지를 선택합니다.

4. **Red Hat Advanced Cluster Management** 콘솔을 사용하여 관리형 클러스터를 배포하는 경우 클러스터 생성 프로세스 중 *아키텍처* 필드에서 관리 클러스터의 아키텍처를 지정합니다.

생성 프로세스에서는 병합된 릴리스 이미지를 사용하여 클러스터를 생성합니다.

1.3.2. 사용 가능한 릴리스 이미지 동기화

릴리스 이미지가 자주 업데이트되므로 릴리스 이미지 목록을 동기화하여 사용 가능한 최신 버전을 선택할 수 있습니다. 릴리스 이미지는 릴리스 2.5용 acm-hive-openshift-releases GitHub 리포지토리에서 사용할 수 있습니다.

릴리스 이미지의 안정성에는 세 가지 수준이 있습니다.

표 1.1. 릴리스 이미지의 안정성 수준

카테고리	설명
stable	클러스터 설치 및 빌드가 올바르게 확인되는 완전히 테스트된 이미지입니다.
신속 (Fast)	부분적으로 테스트되었지만 안정된 버전보다 안정적이지 않을 수 있습니다.
candidate	아직 테스트되지 않았지만 가장 최신 이미지입니다. 버그가 있을 수 있습니다.

다음 단계를 완료하여 목록을 새로 고칩니다.

1. 설치 관리자 관리 acm-hive-openshift-releases 서브스크립션이 활성화된 경우 multiclusterhub 리소스에서 disableUpdateClusterImageSets 값을 true 로 설정하여 서브스크립션을 비활성화합니다.

2. 릴리스 2.5에 대해 acm-hive-openshift-releases GitHub 리포지토리를 복제합니다.

3. 다음 명령과 유사한 명령을 입력하여 서브스크립션을 제거합니다.

```
oc delete -f subscribe/subscription-fast
```

4. 안정적인 릴리스 이미지에 연결하고 다음 명령을 입력하여 Kubernetes 허브 클러스터용 Red Hat Advanced Cluster Management 클러스터를 동기화합니다.

```
make subscribe-stable
```

참고: **Linux** 또는 **MacOS** 운영 체제를 사용하는 경우에만 이 **make** 명령을 실행할 수 있습니다.

약 **1**분 후에 안정적인 릴리스 이미지의 최신 목록을 사용할 수 있습니다.

- 빠른 릴리스 이미지를 동기화하고 표시하려면 다음 명령을 입력합니다.

```
make subscribe-fast
```

참고: **Linux** 또는 **MacOS** 운영 체제를 사용하는 경우에만 이 **make** 명령을 실행할 수 있습니다.

명령을 실행한 후 약 **1**분 후에 사용 가능한 안정적인 빠른 릴리스 이미지 목록이 현재 사용 가능한 이미지로 업데이트됩니다.

- 후보 릴리스 이미지를 동기화하고 표시하려면 다음 명령을 입력합니다.

```
make subscribe-candidate
```

참고: **Linux** 또는 **MacOS** 운영 체제를 사용하는 경우에만 이 **make** 명령을 실행할 수 있습니다.

명령을 실행한 후 약 **1**분 후 사용 가능한 안정적인,**fast** 및 **candidate** 릴리스 이미지 목록이 현재 사용 가능한 이미지로 업데이트됩니다.

5. 클러스터를 생성할 때 **Red Hat Advanced Cluster Management** 콘솔에서 현재 사용 가능한 릴리스 이미지 목록을 확인합니다.

6. 다음 형식으로 명령을 입력하여 이러한 채널에서 업데이트 보기를 취소하여 업데이트 보기를 중지할 수 있습니다.

```
oc delete -f subscribe/subscription-fast
```

1.3.3. 연결된 경우 릴리스 이미지의 사용자 정의 목록 유지

모든 클러스터에 동일한 릴리스 이미지를 사용하도록 할 수 있습니다. 단순화하기 위해 클러스터를 생성할 때 사용할 수 있는 자체 사용자 정의 릴리스 이미지 목록을 생성할 수 있습니다. 사용 가능한 릴리스 이미지를 관리하려면 다음 단계를 완료합니다.

1. 설치 관리자 관리 **acm-hive-openshift-releases** 서브스크립션이 활성화된 경우 **multiclusterhub** 리소스에서 **disableUpdateClusterImageSets** 값을 **true** 로 설정하여 비활성화합니다.

2. **acm-hive-openshift-releases GitHub** 리포지토리 **2.5** 브랜치 를 분기합니다.

3. **stolostron** 대신 분기된 리포지토리의 **GitHub** 이름에 액세스하도록 **spec: pathname** 을 변경하여 **./subscribe/channel.yaml** 파일을 업데이트합니다. 이 단계에서는 **hub** 클러스터가 릴리스 이미지를 검색하는 위치를 지정합니다. 업데이트된 콘텐츠는 다음 예와 유사해야 합니다.

```
spec:
  type: Git
  pathname: https://github.com/<forked_content>/acm-hive-openshift-releases.git
```

forked_content 를 분기된 리포지토리의 경로로 바꿉니다.

4. **Kubernetes** 콘솔을 사용하여 클러스터를 생성할 때 사용할 수 있는 이미지의 **YAML** 파일을 *./clusterImageSets/stable/** 또는 *./clusterImageSets/fast/** 디렉터리에 추가합니다.

팁: 분기된 리포지토리에 변경 사항을 병합하여 기본 리포지토리에서 사용 가능한 **YAML** 파일을 검색할 수 있습니다.

5. 분기된 리포지토리에 변경 사항을 커밋하고 병합합니다.

6. **acm-hive-openshift-releases** 리포지토리를 복제한 후 빠른 릴리스 이미지 목록을 동기화하려면 다음 명령을 입력하여 빠른 이미지를 업데이트합니다.

```
make subscribe-fast
```

참고: Linux 또는 **MacOS** 운영 체제를 사용하는 경우에만 이 **make** 명령을 실행할 수 있습니다.

이 명령을 실행하면 사용 가능한 빠른 릴리스 이미지 목록이 현재 사용 가능한 이미지로 약 1분 후에 업데이트됩니다.

7. 기본적으로 빠른 이미지만 나열됩니다. 안정적인 릴리스 이미지를 동기화하고 표시하려면 다음 명령을 입력합니다.

```
make subscribe-stable
```

참고: Linux 또는 MacOS 운영 체제를 사용하는 경우에만 이 make 명령을 실행할 수 있습니다.

이 명령을 실행하면 약 1분 후에 현재 사용 가능한 이미지와 함께 사용 가능한 안정적인 릴리스 이미지 목록이 업데이트됩니다.

8. 기본적으로 Red Hat Advanced Cluster Management는 몇 가지 ClusterImageSets를 사전 로드합니다. 다음 명령을 사용하여 사용 가능한 항목을 나열하고 기본값을 제거할 수 있습니다.

```
oc get clusterImageSets
oc delete clusterImageSet <clusterImageSet_NAME>
```

참고: 다중 클러스터hub 리소스에서 disableUpdate ClusterImageSets 값을 true 로 설정하여 설치 관리자 관리 자동 업데이트를 비활성화하지 않은 경우 삭제한 모든 이미지가 자동으로 다시 생성됩니다.

9. 클러스터를 생성할 때 Red Hat Advanced Cluster Management 콘솔에서 현재 사용 가능한 릴리스 이미지 목록을 확인합니다.

1.3.4. 연결이 끊긴 동안 사용자 정의 릴리스 이미지 목록 유지

허브 클러스터에 인터넷 연결이 없는 경우 릴리스 이미지의 사용자 정의 목록을 유지 관리해야 하는 경우도 있습니다. 클러스터를 생성할 때 사용 가능한 릴리스 이미지의 자체 사용자 정의 목록을 생성할 수 있습니다. 연결이 끊긴 동안 사용 가능한 릴리스 이미지를 관리하려면 다음 단계를 완료합니다.

1. 연결된 시스템에 있는 동안 acm-hive-openshift-releases GitHub 리포지토리로 이동하여 버전 2.5에 사용할 수 있는 클러스터 이미지 세트에 액세스합니다.

2. **Kubernetes** 허브 클러스터의 연결이 끊긴 **Red Hat Advanced Cluster Management**에 액세스할 수 있는 시스템에 **clusterImageSets** 디렉터리를 복사합니다.

3. 관리형 클러스터에 적합한 다음 단계를 완료하여 관리 대상 클러스터와 연결이 끊긴 리포지토리와 클러스터 이미지 세트 간의 매핑을 추가합니다.

   - **OpenShift Container Platform** 관리 클러스터의 경우 **ImageContentSourcePolicy** 오브젝트를 사용하여 매핑을 완료하는 방법에 대한 정보는 이미지 레지스트리 저장소 미러링 구성을 참조하십시오.

   - **OpenShift Container Platform** 클러스터가 아닌 관리형 클러스터의 경우 **ManageClusterImageRegistry CRD**를 사용하여 이미지 세트의 위치를 덮어씁니다. 매핑 을 위해 클러스터를 덮어쓰는 방법에 대한 정보는 사용자 정의 **ManagedClusterImageRegistry CRD**를 사용하여 클러스터 가져오기 를 참조하십시오.

4. **clusterImageSet YAML** 콘텐츠를 수동으로 추가하여 **Red Hat Advanced Cluster Management** 콘솔을 사용하여 클러스터를 생성할 때 사용 가능한 이미지의 **YAML** 파일을 추가합니다.

5. 나머지 **OpenShift Container Platform** 릴리스 이미지의 **clusterImageSet YAML** 파일을 수정하여 이미지를 저장하는 올바른 오프라인 리포지토리를 참조합니다. 업데이트는 다음 예와 유사해야 합니다.

```
apiVersion: hive.openshift.io/v1
kind: ClusterImageSet
metadata:
  name: img4.4.0-rc.6-x86-64
spec:
  releaseImage: IMAGE_REGISTRY_IPADDRESS_or_DNSNAME/REPO_PATH/ocp-release:4.4.0-rc.6-x86_64
```

   이미지가 **YAML** 파일에서 참조되는 오프라인 이미지 레지스트리에 로드되었는지 확인합니다.

6. 각 **YAML** 파일에 대해 다음 명령을 입력하여 각 **clusterImageSets** 를 생성합니다.

```
oc create -f <clusterImageSet_FILE>
```

   clusterImageSet_FILE 을 클러스터 이미지 세트 파일의 이름으로 교체합니다. 예를 들면 다

음과 같습니다.

```
oc create -f img4.9.9-x86_64.yaml
```

추가할 각 리소스에 대해 이 명령을 실행하면 사용 가능한 릴리스 이미지 목록이 제공됩니다.

7. 또는 **Red Hat Advanced Cluster Management**의 **create** 클러스터 콘솔에 이미지 **URL**을 직접 붙여넣을 수 있습니다. 이미지 **URL**을 추가하면 새 **clusterImageSets**가 생성되지 않는 경우 생성됩니다.

8. 클러스터를 생성할 때 **Red Hat Advanced Cluster Management** 콘솔에서 현재 사용 가능한 릴리스 이미지 목록을 확인합니다.

1.4. 베어 메탈 자산 생성 및 수정

사용 중단 알림: 베어 메탈 자산을 사용하여 베어 메탈 클러스터를 생성하는 절차는 더 이상 사용되지 않습니다. 권장 프로세스에 대해서는 온-프레미스 환경에서 클러스터 생성을 참조하십시오.

베어 메탈 자산은 **OpenShift Container Platform** 클러스터를 실행하도록 구성하는 가상 또는 물리적 서버입니다. **Red Hat Advanced Cluster Management for Kubernetes**는 관리자가 생성하는 베어메탈 자산에 연결됩니다. 그런 다음 관리 클러스터에 베어 메탈 자산을 배포할 수 있습니다.

hub 클러스터 인벤토리 컨트롤러는 베어 메탈 자산 인벤토리 레코드를 보유하는 **BareMetalAsset** 라는 **CRD**(사용자 정의 리소스 정의)를 정의합니다. 관리형 클러스터를 프로비저닝할 때 인벤토리 컨트롤러는 관리 클러스터에서 해당 **BareMetalHost** 리소스로 **BareMetalAsset** 인벤토리 레코드를 조정합니다.

Red Hat Advanced Cluster Management는 **BareMetalAsset CR**을 사용하여 구성 관리 데이터베이스(**CMDB**) 또는 유사한 시스템에 입력한 레코드를 기반으로 클러스터 하드웨어를 프로비저닝합니다. 외부 도구 또는 자동화는 **CMDB**를 폴링하고 **Red Hat Advanced Cluster Management API**를 사용하여 관리 클러스터에서 후속 배포를 위해 **hub** 클러스터에 해당 **BareMetalAsset** 및 해당 **Secret** 리소스를 생성합니다.

다음 절차에 따라 **Red Hat Advanced Cluster Management**에서 관리하는 클러스터의 베어 메탈 자산을 생성하고 관리할 수 있습니다.

- 사전 요구 사항

- 콘솔을 사용하여 베어 메탈 자산 생성
- CLI를 사용하여 베어 메탈 자산 생성
- 콘솔을 사용하여 베어 메탈 자산 대량 가져오기
- 베어 메탈 자산 수정
- 베어 메탈 자산 제거
- REST API를 사용하여 베어 메탈 자산 생성

1.4.1. 사전 요구 사항

베어 메탈 자산을 생성하기 전에 다음 사전 요구 사항이 필요합니다.

- **OpenShift Container Platform** 버전 **4.6** 이상에 배포된 **Red Hat Advanced Cluster Management hub** 클러스터입니다.
- **Red Hat Advanced Cluster Management hub** 클러스터에 액세스하여 베어 메탈 자산에 연결합니다.
- 구성된 베어 메탈 자산으로 로그인하고, 로그인하고 관리하는 데 필요한 권한으로 자격 증명을 로그인합니다.

  참고: 베어 메탈 자산에 대한 인증 정보에는 관리자가 제공하는 자산에 대한 다음 항목이 포함됩니다. **BMC(** 사용자 이름 암호 베이스 보드 관리 컨트롤러) 주소 부팅 **NIC MAC** 주소

1.4.2. 콘솔을 사용하여 베어 메탈 자산 생성

Kubernetes 콘솔용 **Red Hat Advanced Cluster Management**를 사용하여 베어 메탈 자산을 생성하려면 인프라 > 베어 메탈 자산으로 이동합니다. 베어 메탈 자산 만들기 를 선택하고 콘솔에서 절차를 완료합니다.

베어 메탈 자산의 이름은 클러스터를 생성할 때 이를 식별합니다.

베어 메탈 자산, 관리형 베어 메탈 클러스터 및 관련 시크릿은 동일한 네임스페이스에 있어야 합니다.

+ 이 네임스페이스에 액세스할 수 있는 사용자는 클러스터를 생성할 때 이 자산을 클러스터에 연결할 수 있습니다.

Baseboard Management Controller 주소는 호스트와의 통신을 활성화하는 컨트롤러입니다. 지원되는 프로토콜은 다음과 같습니다.

- 자세한 내용은 **IPMI 2.0 사양** 을 참조하십시오.
- 자세한 내용은 **iDRAC(Integrated Dell Remote Access Controller 9)** 지원을 참조하십시오.
- **iRMC**는 자세한 내용은 **FUJITSU Software ServerView Suite integrated Remote Management Controller - iRMC S5** 를 참조하십시오.
- 자세한 내용은 **Redfish 사양** 을 참조하십시오.

부팅 **NIC MAC** 주소는 베어 메탈 자산에서 호스트를 프로비저닝하는 데 사용되는 호스트의 네트워크 연결 **NIC**의 **MAC** 주소입니다.

베어 메탈에서 클러스터 생성을 계속할 수 있습니다.

1.4.3. CLI를 사용하여 베어 메탈 자산 생성

BareMetalAsset CR을 사용하여 클러스터의 특정 네임스페이스에 대한 베어 메탈 자산을 생성합니다. 각 **BareMetalAsset** 에는 동일한 네임스페이스에 **BMC(Baseboard Management Controller)** 인증 정보 및 시크릿 이름이 포함된 해당 **Secret** 도 있습니다.

1.4.3.1. 사전 요구 사항

- 허브 클러스터에 Kubernetes용 Red Hat Advanced Cluster Management를 설치합니다.
- Red Hat OpenShift CLI(oc)를 설치합니다.
- cluster-admin 권한이 있는 사용자로 로그인합니다.

1.4.3.2. 베어 메탈 자산 생성

1. 사용자 환경에 베어 메탈 자산을 설치 및 프로비저닝합니다.
2. BMC의 전원을 켜고 하드웨어의 IPMI 또는 Redfish BMC 주소 및 MAC 주소를 기록하십시오.
3. 다음 BareMetalAsset 및 Secret CR을 생성하고 파일을 baremetalasset-cr.yaml 로 저장합니다.

```
apiVersion: inventory.open-cluster-management.io/v1alpha1
kind: BareMetalAsset
metadata:
  name: <baremetalasset-machine>
  namespace: <baremetalasset-namespace>
spec:
  bmc:
    address: ipmi://<out_of_band_ip>:<port>
    credentialsName: baremetalasset-machine-secret
  bootMACAddress: "00:1B:44:11:3A:B7"
  hardwareProfile: "hardwareProfile"
  role: "<role>"
  clusterName: "<cluster name>"
---
apiVersion: v1
kind: Secret
metadata:
  name: baremetalasset-machine-secret
type: Opaque
data:
  username: <username>
  password: <password>
```

- baremetalasset-machine 을 베어 메탈 자산이 있는 머신의 이름으로 교체합니다. 생성되면 관리 클러스터의 BareMetalHost 가 hub 클러스터의 해당 BareMetalAsset 와 동일

한 이름을 가져옵니다. **BareMetalHost** 이름은 항상 해당 **BareMetalAsset** 이름과 일치해야 합니다.

- **baremetalasset-namespace** 를 베어 메탈 자산이 생성된 클러스터 네임스페이스로 교체합니다.
- **out_of_band_ip** 및 포트를 베어 메탈 자산의 주소 및 포트로 바꿉니다. **Redfish** 주소 지정의 경우 **redfish://<out-of-band-ip>/redfish/v1/Systems/1** 주소 형식을 사용합니다.
- **role** 을 **worker,master** 로 교체하거나 시스템 역할 유형에 따라 비워 둡니다. 역할 설정은 베어 메탈 자산을 클러스터의 특정 머신 역할 유형과 일치시키는 데 사용됩니다. 지정된 머신 역할 유형의 **BareMetalAsset** 리소스를 사용하여 다른 역할을 채우지 않아야 합니다. **role** 값은 **key inventory.open-cluster-management.io/role** 가 있는 레이블의 값으로 사용됩니다. 이를 통해 클러스터 관리 애플리케이션 또는 사용자가 특정 역할을 위한 인벤토리를 쿼리할 수 있습니다.
- **cluster_name** 을 클러스터 관리 애플리케이션 또는 사용자가 특정 클러스터와 연결된 인벤토리를 쿼리하는 데 사용하는 클러스터 이름으로 바꿉니다. 클러스터 배포에 추가하지 않고 베어 메탈 자산을 생성하려면 이 값을 비워 둡니다.
- **username** 을 시크릿의 사용자 이름으로 교체합니다.
- 암호를 보안의 암호로 바꿉니다.

4. 다음 명령을 실행하여 **BareMetalAsset CR**을 생성합니다.

```
oc create -f baremetalasset-cr.yaml
```

5. **BareMetalAsset** 가 성공적으로 생성되었는지 확인합니다.

```
oc get baremetalassets -A
```

출력 예:

```
NAMESPACE       NAME                          AGE
ocp-example-bm   baremetalasset-machine        2m
ocp-example-bm   csv-f24-h27-000-r630-master-1-1   4d21h
```

1.4.4. 콘솔을 사용하여 베어 메탈 자산 대량 가져오기

CSV 형식 목록을 사용하여 **Kubernetes** 콘솔용 **Red Hat Advanced Cluster Management**를 사용하여 베어 메탈 자산을 대량으로 가져올 수 있습니다.

1.4.4.1. 사전 요구 사항

- 하나 이상의 대화 상자 클러스터를 관리하는 허브 클러스터에 **Red Hat Advanced Cluster Management**를 설치합니다.
- **OpenShift Container Platform CLI, oc**를 설치합니다.
- **cluster-admin** 권한이 있는 사용자로 로그인합니다.

1.4.4.2. 자산을 가져옵니다.

베어 메탈 자산 세트를 가져오려면 다음 단계를 완료합니다.

1. **Red Hat Advanced Cluster Management** 콘솔에서 탐색 메뉴에서 클러스터 관리 > 베어 메탈 자산을 선택합니다.
2. 자산 가져오기 를 선택하고 베어 메탈 자산 데이터가 포함된 **CSV** 파일을 가져옵니다. **CSV** 파일에는 다음 헤더 열이 있어야 합니다.

```
hostName, hostNamespace, bmcAddress, macAddress, role (optional), username, password
```

1.4.5. 베어 메탈 자산 수정

베어 메탈 자산의 설정을 수정해야 하는 경우 다음 단계를 완료합니다.

1. **Kubernetes** 콘솔용 **Red Hat Advanced Cluster Management** 탐색에서 인프라 > 베어 메탈 자산을 선택합니다.

2. 표에서 수정할 자산의 옵션 메뉴를 선택합니다.

3. 자산 편집을 선택합니다.

1.4.6. 베어 메탈 자산 제거

베어 메탈 자산을 더 이상 클러스터에 사용하지 않는 경우 사용 가능한 베어 메탈 자산 목록에서 제거할 수 있습니다. 사용되지 않는 자산을 제거하면 사용 가능한 자산 목록이 간소화되고 해당 자산의 실수로 선택되지 않습니다.

콘솔에서 베어 메탈 자산을 제거하려면 다음 단계를 완료합니다.

1. **Kubernetes** 콘솔용 **Red Hat Advanced Cluster Management** 탐색에서 인프라 > 베어 메탈 자산을 선택합니다.

2. 표에서 제거할 자산의 옵션 메뉴를 선택합니다.

3. 자산 삭제를 선택합니다.

1.4.7. REST API를 사용하여 베어 메탈 자산 생성

OpenShift Container Platform REST API를 사용하여 **Red Hat Advanced Cluster Management** 클러스터에서 사용할 베어 메탈 자산을 관리할 수 있습니다. 이 기능은 환경에서 베어 메탈 자산을 관리할 별도의 **CMDB** 애플리케이션 또는 데이터베이스가 있는 경우 유용합니다.

1.4.7.1. 사전 요구 사항

- 허브 클러스터에 **Kubernetes**용 **Red Hat Advanced Cluster Management**를 설치합니다.
- **OpenShift Container Platform CLI, oc**를 설치합니다.
- **cluster-admin** 권한이 있는 사용자로 로그인합니다.

1.4.7.2. 베어 메탈 자산 생성

REST API를 사용하여 베어 메탈 자산을 생성하려면 다음을 수행하십시오.

1. **hub** 클러스터에 대한 로그인 토큰을 가져와서 명령줄에서 클러스터에 로그인합니다. 예를 들면 다음과 같습니다.

```
oc login --token=<login_token> --server=https://<hub_cluster_api_url>:6443
```

2. 클러스터에 추가할 베어 메탈 자산의 세부 정보를 사용하여 다음 **curl** 명령을 수정하고 명령을 실행합니다.

```
$ curl --location --request POST '<hub_cluster_api_url>:6443/apis/inventory.open-cluster-management.io/v1alpha1/namespaces/<bare_metal_asset_namespace>/baremetalassets?fieldManager=kubectl-create' \
--header 'Authorization: Bearer <login_token>' \
--header 'Content-Type: application/json' \
--data-raw '{
    "apiVersion": "inventory.open-cluster-management.io/v1alpha1",
    "kind": "BareMetalAsset",
    "metadata": {
        "name": "<baremetalasset_name>",
        "namespace": "<bare_metal_asset_namespace>"
    },
    "spec": {
        "bmc": {
            "address": "ipmi://<ipmi_address>",
            "credentialsName": "<credentials-secret>"
        },
        "bootMACAddress": "<boot_mac_address>",
        "clusterName": "<cluster_name>",
        "hardwareProfile": "hardwareProfile",
        "role": "worker"
    }
}'
```

- **baremetalasset-name** 을 베어 메탈 자산의 이름으로 교체합니다. 생성되면 관리 클러스터의 **BareMetalHost** 가 **hub** 클러스터의 해당 **BareMetalAsset** 와 동일한 이름을 가져옵니다. **BareMetalHost** 이름은 항상 해당 **BareMetalAsset** 이름과 일치해야 합니다.
- **baremetalasset-namespace** 를 베어 메탈 자산이 생성된 클러스터 네임스페이스로 교체합니다.
- **out_of_band_ip** 및 포트를 베어 메탈 자산의 주소 및 포트로 바꿉니다. **Redfish** 주소 지정의 경우 **redfish://<out-of-band-ip>/redfish/v1/Systems/1** 주소 형식을 사용합니다.

- role 을 worker,master 로 교체하거나 시스템 역할 유형에 따라 비워 둡니다. 역할 설정은 베어 메탈 자산을 클러스터의 특정 머신 역할 유형과 일치시키는 데 사용됩니다. 지정된 머신 역할 유형의 BareMetalAsset 리소스를 사용하여 다른 역할을 채우지 않아야 합니다. role 값은 key inventory.open-cluster-management.io/role 가 있는 레이블의 값으로 사용됩니다. 이를 통해 클러스터 관리 애플리케이션 또는 사용자가 특정 역할을 위한 인벤토리를 쿼리할 수 있습니다.

- cluster_name 을 클러스터 관리 애플리케이션 또는 사용자가 특정 클러스터와 연결된 인벤토리를 쿼리하는 데 사용하는 클러스터 이름으로 바꿉니다. 클러스터 배포에 추가하지 않고 베어 메탈 자산을 생성하려면 이 값을 비워 둡니다.

  참고: 이전 curl 명령의 경우 API 서버가 HTTPS를 통해 제공되며 안전하게 액세스되는 것으로 가정합니다. 개발 또는 테스트 환경에서는 --insecure 매개변수를 전달할 수 있습니다.

팁: oc 명령에 --v=9 를 추가하여 결과 작업의 원시 출력을 확인할 수 있습니다. 이 기능은 oc 명령의 REST API 경로를 확인하는 데 유용할 수 있습니다.

1.5. 인프라 환경 생성

Kubernetes 콘솔용 Red Hat Advanced Cluster Management for Kubernetes 콘솔을 사용하여 호스트를 관리하고 해당 호스트에서 클러스터를 생성할 수 있는 인프라 환경을 생성할 수 있습니다.

- 사전 요구 사항
- 중앙 인프라 관리 서비스 활성화
  - 프로비저닝 사용자 정의 리소스(CR)를 수동으로 생성
  - Amazon Web Services에서 중앙 인프라 관리 활성화
- 콘솔을 사용하여 인프라 환경 생성

인프라 환경은 다음과 같은 기능을 지원합니다.

- 클러스터의 제로 프로비저닝: 스크립트를 사용하여 클러스터를 배포합니다. 자세한 내용은 Red Hat OpenShift Container Platform 설명서의 연결이 끊긴 환경에서 대규모로 분산 단위 배포를 참조하십시오.
- 늦은 바인딩: 인프라 관리자가 호스트를 부팅하고 클러스터 작성자는 나중에 해당 호스트에 클러스터를 바인딩할 수 있습니다. 클러스터 작성자는 늦은 바인딩을 사용할 때 인프라에 대한 관리자 권한이 필요하지 않습니다.
- 듀얼 스택: IPv4 및 IPv6 주소가 모두 있는 클러스터를 배포합니다. 듀얼 스택은 OVN-Kubernetes 네트워킹 구현을 사용하여 여러 서브넷을 지원합니다.
- 원격 작업자 노드 추가: 생성 및 실행 후 클러스터에 원격 작업자 노드를 추가하여 백업 목적으로 다른 위치에 노드를 추가할 수 있는 유연성을 제공합니다.
- NMState를 사용하는 고정 IP: NMState API를 사용하여 환경에 대한 고정 IP 주소를 정의합니다.

1.5.1. 사전 요구 사항

인프라 환경을 생성하기 전에 다음 사전 요구 사항을 참조하십시오.

- OpenShift Container Platform이 hub 클러스터에 배포되어 있어야 합니다.
- Red Hat Advanced Cluster Management hub 클러스터(연결됨) 클러스터의 인터넷 액세스 또는 환경 생성에 필요한 이미지를 검색하려면 인터넷에 연결되지 않은 내부 또는 미러 레지스트리에 연결해야 합니다.
- 허브 클러스터에 구성된 CIM(Central Infrastructure Management) 기능의 구성된 인스턴스가 필요합니다. 절차에 대한 중앙 인프라 관리 서비스 활성화를 참조하십시오.
- OpenShift Container Platform 풀 시크릿이 필요합니다. 자세한 내용은 이미지 풀 시크릿 사용을 참조하십시오.

- 기본적으로 ~/.ssh/id_rsa.pub 파일에 있는 SSH 키가 필요합니다.
- 구성된 스토리지 클래스가 필요합니다.
- 연결이 끊긴 환경만 해당: OpenShift Container Platform 설명서에서 연결이 끊긴 환경을 준비하는 절차를 완료합니다.

1.5.2. 중앙 인프라 관리 서비스 활성화

중앙 인프라 관리 서비스는 {mce-short}와 함께 제공되며 OpenShift Container Platform 클러스터를 배포합니다. CIM은 허브 클러스터에서 MultiClusterHub Operator를 활성화할 때 배포되지만 활성화해야 합니다.

CIM 서비스를 활성화하려면 다음 단계를 완료합니다.

중요: 베어 메탈, Red Hat OpenStack Platform, VMware vSphere 또는 사용자 프로비저닝 인프라(UPI) 방법을 사용하여 hub 클러스터가 다음 플랫폼 중 하나에 설치되어 있고 플랫폼이 None 인 경우 다음 단계를 완료합니다. hub 클러스터가 다른 플랫폼에 있는 경우 이 단계를 건너뜁니다.

1. 다음 명령을 실행하여 Bare Metal Operator가 모든 네임스페이스를 조사할 수 있도록 프로비저닝 리소스를 수정합니다.

```
oc patch provisioning provisioning-configuration --type merge -p '{"spec":
{"watchAllNamespaces": true }}'
```

2. 연결이 끊긴 환경의 경우: 인프라 Operator와 *동일한 네임스페이스에* ConfigMap 을 생성하여 미러 레지스트리의 ca-bundle.crt 및 registries.conf 값을 지정합니다. 파일 ConfigMap 은 다음 예와 유사해야 합니다.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: <mirror-config>
  namespace: "<infrastructure-operator-namespace>"
  labels:
    app: assisted-service
data:
  ca-bundle.crt: |
    -----BEGIN CERTIFICATE-----
```

```
  certificate contents
  -----END CERTIFICATE-----

registries.conf: |
  unqualified-search-registries = ["registry.access.redhat.com", "docker.io"]

  [[registry]]
    prefix = ""
    location = "quay.io/edge-infrastructure"
    mirror-by-digest-only = false

    [[registry.mirror]]
    location = "mirror1.registry.corp.com:5000/edge-infrastructure"
```

1.5.2.1. *AgentServiceConfig* 사용자 정의 리소스 생성

다음 단계를 완료하여 **AgentServiceConfig** 사용자 지정 리소스를 생성합니다.

1. 연결이 끊긴 환경의 경우에만: **agent_service_config.yaml** 파일에 다음 **YAML** 콘텐츠를 저장하고 필요에 따라 값을 교체합니다.

```
apiVersion: agent-install.openshift.io/v1beta1
kind: AgentServiceConfig
metadata:
 name: agent
spec:
  databaseStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <db_volume_size>
  filesystemStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <fs_volume_size>
  mirrorRegistryRef:
    name: <mirror_config>
  unauthenticatedRegistries:
    - <unauthenticated_registry>
  imageStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <img_volume_size>
  osImages:
    - openshiftVersion: "<ocp_version>"
      version: "<ocp_release_version>"
```

```
    url: "<iso_url>"
    rootFSUrl: "<root_fs_url>"
    cpuArchitecture: "x86_64"
```

mirror_config 를 미러 레지스트리 구성 세부 정보가 포함된 **ConfigMap** 의 이름으로 교체합니다.

인증이 필요하지 않은 미러 레지스트리를 사용하는 경우 선택적 **unauthenticated_registry** 매개변수를 포함합니다. 이 목록의 항목은 검증되지 않거나 가져오기 시크릿에 항목이 있어야 합니다.

2. 연결된 환경의 경우에만: **agent_service_config.yaml** 파일에 다음 **YAML** 콘텐츠를 저장합니다.

```
apiVersion: agent-install.openshift.io/v1beta1
kind: AgentServiceConfig
metadata:
 name: agent
spec:
  databaseStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <db_volume_size>
  filesystemStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <fs_volume_size>
  imageStorage:
    accessModes:
    - ReadWriteOnce
    resources:
      requests:
        storage: <img_volume_size>
```

db_volume_size 를 **databaseStorage** 필드의 볼륨 크기로 교체합니다(예: **10G**). 이 값은 클러스터의 데이터베이스 테이블 및 데이터베이스 뷰와 같은 파일을 저장하기 위해 할당된 스토리지 양을 지정합니다. 클러스터가 많은 경우 더 높은 값을 사용해야 할 수 있습니다.

fs_volume_size 를 **filesystemStorage** 필드의 볼륨 크기로 교체합니다(예: 클러스터당 **200M**, 지원되는 **OpenShift Container Platform** 버전당 **2-3G**). 필요한 최소 값은 **100G** 입니다. 이 값은 클러스터의 로그, 매니페스트, **kubeconfig** 파일을 저장하기 위해 할당된 스토리지 양을 지정합니다. 클러스터가 많은 경우 더 높은 값을 사용해야 할 수 있습니다.

img_volume_size 를 **imageStorage** 필드의 볼륨 크기(예: 운영 체제 이미지당 **2G**)로 바꿉니다. 최소 크기는 **50G** 입니다. 이 값은 클러스터 이미지에 할당되는 스토리지 양을 지정합니다. 실행 중인 **Red Hat Enterprise Linux CoreOS** 인스턴스마다 **1GB**의 이미지 스토리지를 허용해야 합니다. **Red Hat Enterprise Linux CoreOS**의 여러 클러스터와 인스턴스가 있는 경우 더 높은 값을 사용해야 할 수 있습니다.

ocp_version 을 설치할 **OpenShift Container Platform** 버전으로 교체합니다(예: **4.9**).

ocp_release_version 을 특정 설치 버전 (예: **49.83.202103251640-0**)으로 바꿉니다.

iso_url 을 **ISO URL**로 교체합니다(예: **https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.10/4.10.3/rhcos-4.10.3-x86_64-live.x86_64.iso**). 다른 값은 **https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.10/4.10.3/** 에서 확인할 수 있습니다.

root_fs_url 을 루트 **FS** 이미지 **URL**로 교체합니다(예: **https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.10/4.10.3/rhcos-4.10.3-x86_64-live-rootfs.x86_64.img**). 다른 값은 **https://mirror.openshift.com/pub/openshift-v4/x86_64/dependencies/rhcos/4.10/4.10.3/** 에서 확인할 수 있습니다.

3. 다음 명령을 실행하여 **AgentServiceConfig** 사용자 지정 리소스를 생성합니다.

```
oc create -f agent_service_config.yaml
```

출력은 다음 예와 유사할 수 있습니다.

```
agentserviceconfig.agent-install.openshift.io/agent created
```

assisted-service 및 **assisted-image-service** 배포를 확인하고 해당 **Pod**가 준비되고 실행 중인지 확인할 수 있습니다. 콘솔을 사용하여 인프라 환경 생성을 계속합니다.

1.5.2.2. 프로비저닝 사용자 정의 리소스(CR)를 수동으로 생성

다음 명령을 사용하여 자동화된 프로비저닝 용 서비스를 활성화하는 프로비저닝 **CR**을 수동으로 생성합니다.

```
oc create -f provisioning-configuration.yaml
```

CR은 다음 샘플과 유사할 수 있습니다.

```
apiVersion: metal3.io/v1alpha1
kind: Provisioning
metadata:
  name: provisioning-configuration
spec:
  provisioningNetwork: Disabled
  watchAllNamespaces: true
```

1.5.2.3. Amazon Web Services에서 중앙 인프라 관리 활성화

Amazon Web Services에서 hub 클러스터를 실행하고 CIM 서비스를 활성화하려면 CIM 활성화 후 다음 추가 단계를 완료합니다.

1. 허브에 로그인했는지 확인하고 다음 명령을 실행하여 assisted-image-service 에서 구성된 고유 도메인을 찾습니다.

```
oc get routes --all-namespaces | grep assisted-image-service
```

도메인은 assisted-image-service-multicluster-engine.apps.<yourdomain>.com과 유사할 수 있습니다.

2. 허브에 로그인했는지 확인하고 NLB type 매개변수를 사용하여 고유한 도메인이 있는 새 IngressController 를 생성합니다. 다음 예제를 참조하십시오.

```
apiVersion: operator.openshift.io/v1
kind: IngressController
metadata:
  name: ingress-controller-with-nlb
  namespace: openshift-ingress-operator
spec:
  domain: nlb-apps.<domain>.com
  routeSelector:
     matchLabels:
       router-type: nlb
  endpointPublishingStrategy:
    type: LoadBalancerService
    loadBalancer:
      scope: External
      providerParameters:
```

```
    type: AWS
    aws:
      type: NLB
```

3. **nlb-apps.< domain >.com에서 <your domain >을 <yourdomain> 으로 교체하여 IngressController 의 domain 매개변수에 < yourdomain >을 추가합니다.**

4. 다음 명령을 사용하여 새 **IngressController** 를 적용합니다.

```
oc apply -f ingresscontroller.yaml
```

5. 다음 명령을 실행하여 **nlb-apps** 위치를 사용하도록 **assisted-image-service** 경로를 편집합니다.

```
oc edit route assisted-image-service -n <namespace>
```

팁: 기본 네임스페이스는 :mce:를 설치한 위치입니다.

6. **assisted-image-service** 경로에 다음 행을 추가합니다.

```
metadata:
  labels:
    router-type: nlb
  name: assisted-image-service
```

7. **assisted-image-service** 경로에서 **spec.host** 의 **URL** 값을 찾습니다. **URL**은 다음 예와 유사할 수 있습니다.

assisted-image-service-multicluster-engine.apps.<yourdomain>.com

8. 새 **IngressController** 에 구성된 도메인과 일치하도록 **URL**의 앱을 **nlb- apps** 로 바꿉니다.

Amazon Web Services에서 **CIM** 서비스가 활성화되어 있는지 확인하려면 다음 단계를 완료합니다.

1. 다음 명령을 실행하여 **Pod**가 정상인지 확인합니다.

```
oc get pods -n multicluster-engine | grep assist
```

2. 새 인프라 환경을 생성하고 다운로드 URL에서 새 nlb-apps URL을 사용하는지 확인합니다.

1.5.3. 콘솔을 사용하여 인프라 환경 생성

Red Hat Advanced Cluster Management 콘솔에서 인프라 환경을 생성하려면 다음 단계를 완료합니다.

1. 탐색 메뉴에서 인프라 > 인프라 환경으로 이동 하여 인프라 환경 생성을 클릭합니다.

2. 인프라 환경 설정에 다음 정보를 추가합니다.

- name: 사용자 환경의 고유 이름입니다.
- 네트워크 유형: 환경에 추가할 수 있는 호스트 유형을 지정합니다. 베어 메탈 호스트를 사용하는 경우에만 고정 IP 옵션을 사용할 수 있습니다.
- Location: 호스트의 지리적 위치를 지정합니다. 지리적 위치를 사용하여 클러스터를 생성할 때 클러스터의 데이터가 저장되는 위치를 쉽게 확인할 수 있습니다.
- 레이블: 인프라 환경에 레이블을 추가할 수 있는 선택적 필드로, 보다 쉽게 찾아 특성을 공유하는 다른 환경으로 환경을 그룹화할 수 있습니다. 네트워크 유형 및 위치에 대한 선택 사항이 레이블 목록에 자동으로 추가됩니다.
- 풀 시크릿: OpenShift Container Platform 리소스에 액세스할 수 있는 OpenShift Container Platform 풀 시크릿 입니다.
- SSH 공개 키: 호스트와의 보안 통신을 활성화하는 SSH 키입니다. 기본적으로 ~/.ssh/id_rsa.pub 파일에 있습니다.
- 모든 클러스터에서 프록시 설정을 활성화하려면 설정을 선택하여 활성화합니다. 이를 위해서는 다음 정보를 입력해야 합니다.

- **HTTP 프록시 URL:** 검색 서비스에 액세스할 때 사용해야 하는 **URL**입니다.

- **HTTPS 프록시 URL:** 검색 서비스에 액세스할 때 사용해야 하는 보안 프록시 **URL** 입니다. **https** 는 아직 지원되지 않으므로 **http** 형식이어야 합니다.

- 프록시 도메인 없음**:** 프록시를 바이패스해야 하는 쉼표로 구분된 도메인 목록입니다. 해당 도메인에 있는 모든 하위 도메인을 포함하려면 마침표로 도메인 이름을 시작합니다 **.** 모든 대상에 대한 프록시를 바이패스하려면 및 별표 * 를 추가합니다.

이제 인프라 환경에 호스트를 추가하여 계속할 수 있습니다.

인프라 환경에 액세스하려면 콘솔에서 **Infrastructure > Host inventory** 를 선택합니다. 목록에서 인프라 환경을 선택하여 해당 인프라 환경에 대한 세부 정보 및 호스트를 확인합니다.

1.5.4. 인프라 환경에 호스트 추가

Red Hat Advanced Cluster Management for Kubernetes 콘솔을 사용하여 인프라 환경에 호스트를 추가할 수 있습니다. 호스트를 추가하면 클러스터를 생성할 때 이미 구성된 호스트를 더 쉽게 선택할 수 있습니다.

호스트를 추가하려면 다음 단계를 완료합니다.

1. **Red Hat Advanced Cluster Management** 탐색에서 인프라 **>** 인프라 환경을 선택합니다.

2. 호스트를 추가하여 설정을 볼 인프라 환경을 선택합니다.

3. *호스트* 탭을 선택하여 이미 해당 환경에 추가된 호스트를 확인하고 호스트를 추가합니다. 사용 가능한 호스트가 테이블에 표시되는 데 몇 분이 걸릴 수 있습니다.

4. **Discovery ISO** 또는 **BMC(Baseboard Management Controller)** 를 선택하여 호스트 정보를 입력합니다.

5. **Discovery ISO** 옵션을 선택하는 경우 다음 단계를 완료합니다.

   a. 콘솔에 제공된 명령을 복사하여 **ISO**를 다운로드하거나 **Discovery ISO** 다운로드를 선택합니다.

   b. 부팅 가능한 장치에서 명령을 실행하여 각 호스트를 시작합니다.

   c. 보안을 강화하기 위해 검색된 각 호스트에 대해 **Approve host** 를 선택합니다. 이 추가 단계는 **ISO** 파일이 변경되어 인증되지 않은 사람이 실행하는 경우 일부 보호 조치를 제공합니다.

   d. 이름이 지정된 호스트, **localhost** 의 이름을 고유한 이름으로 변경합니다.

6. **BMC(Baseboard Management Controller)** 옵션을 선택하는 경우 다음 단계를 완료합니다.

   참고: 호스트를 추가하기 위한 **BMC** 옵션은 **Red Hat Advanced Cluster Management hub** 클러스터의 플랫폼이 베어 메탈, **Red Hat OpenStack Platform, VMware vSphere** 또는 사용자 프로비저닝 인프라(**UPI**) 방법을 사용하여 설치된 경우에만 사용할 수 있으며 플랫폼은 **None** 입니다.

   a. 호스트의 **BMC**에 대한 연결 세부 정보를 추가합니다.

   b. 부팅 프로세스를 시작하려면 호스트 추가 를 선택합니다. 호스트는 검색 **ISO** 이미지를 사용하여 자동으로 부팅되며 호스트 목록이 시작될 때 추가됩니다.

   BMC 옵션을 사용하여 호스트를 추가하면 호스트가 자동으로 승인됩니다.

이제 이 인프라 환경에서 온-프레미스 클러스터를 만들 수 있습니다. 클러스터 생성에 대한 자세한 내용은 온-프레미스 환경에서 클러스터 생성을 참조하십시오.

1.6. 클러스터 생성

Red Hat Advanced Cluster Management for Kubernetes를 사용하여 클라우드 공급자 전반에서 **Red Hat OpenShift Container Platform** 클러스터를 생성하는 방법을 알아보십시오.

멀티 클러스터 엔진에서는 **OpenShift Container Platform**과 함께 제공되는 **Hive Operator**를 사용하여 온-프레미스 클러스터 및 호스트된 컨트롤 플레인을 제외한 모든 공급자에 대한 클러스터를 프로비저닝합니다. 온프레미스 클러스터를 프로비저닝할 때 다중 클러스터 엔진에서는 **OpenShift Container Platform**과 함께 제공되는 **CIM(Central Infrastructure Management)** 및 지원 설치 관리자 기능을 사용합니다. 호스트된 컨트롤 플레인의 호스트 클러스터는 **HyperShift Operator**를 사용하여 프로비저닝됩니다.

- 클러스터 생성 중 추가 매니페스트 구성
- **Amazon Web Services**에서 클러스터 생성
- **Microsoft Azure**에서 클러스터 생성
- **Google Cloud Platform**에서 클러스터 생성
- **VMware vSphere**에서 클러스터 생성
- **Red Hat OpenStack Platform**에서 클러스터 생성
- **Red Hat Virtualization**에서 클러스터 생성
- 베어 메탈에서 클러스터 생성
- 온-프레미스 환경에서 클러스터 생성

1.6.1. 클러스터 생성 중 추가 매니페스트 구성

클러스터를 생성하는 설치 프로세스 중에 추가 **Kubernetes** 리소스 매니페스트를 구성할 수 있습니다. 이는 네트워킹 구성 또는 로드 밸런서 설정과 같은 시나리오에 대한 추가 매니페스트를 구성해야 하는 경우 도움이 될 수 있습니다.

클러스터를 생성하기 전에 추가 리소스 매니페스트가 포함된 **ConfigMap** 을 지정하는 **ClusterDeployment** 리소스에 대한 참조를 추가해야 합니다.

참고: ClusterDeployment 리소스와 ConfigMap 은 동일한 네임스페이스에 있어야 합니다. 다음 예제에서는 콘텐츠가 어떻게 표시되는지를 보여줍니다.

- 리소스 매니페스트가 있는 ConfigMap

  다른 ConfigMap 리소스가 있는 매니페스트가 포함된 ConfigMap 입니다. 리소스 매니페스트 ConfigMap 에는 data.<resource_name>\.yaml 패턴에 추가된 리소스 구성이 포함된 여러 키가 포함될 수 있습니다.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: <my-baremetal-cluster-install-manifests>
  namespace: <mynamespace>
data:
  99_metal3-config.yaml: |
    kind: ConfigMap
    apiVersion: v1
    metadata:
      name: metal3-config
      namespace: openshift-machine-api
    data:
      http_port: "6180"
      provisioning_interface: "enp1s0"
      provisioning_ip: "172.00.0.3/24"
      dhcp_range: "172.00.0.10,172.00.0.100"
      deploy_kernel_url: "http://172.00.0.3:6180/images/ironic-python-agent.kernel"
      deploy_ramdisk_url: "http://172.00.0.3:6180/images/ironic-python-agent.initramfs"
      ironic_endpoint: "http://172.00.0.3:6385/v1/"
      ironic_inspector_endpoint: "http://172.00.0.3:5150/v1/"
      cache_url: "http://192.168.111.1/images"
      rhcos_image_url: "https://releases-art-
rhcos.svc.ci.openshift.org/art/storage/releases/rhcos-
4.3/43.81.201911192044.0/x86_64/rhcos-43.81.201911192044.0-
openstack.x86_64.qcow2.gz"
```

- 리소스 매니페스트 ConfigMap 이 참조된 ClusterDeployment

  리소스 매니페스트 ConfigMap 은 spec.provisioning.manifestsConfigMapRef 에서 참조됩니다.

```
apiVersion: hive.openshift.io/v1
kind: ClusterDeployment
metadata:
  name: <my-baremetal-cluster>
```

```
  namespace: <mynamespace>
  annotations:
    hive.openshift.io/try-install-once: "true"
spec:
  baseDomain: test.example.com
  clusterName: <my-baremetal-cluster>
  controlPlaneConfig:
    servingCertificates: {}
  platform:
    baremetal:
      libvirtSSHPrivateKeySecretRef:
        name: provisioning-host-ssh-private-key
  provisioning:
    installConfigSecretRef:
      name: <my-baremetal-cluster-install-config>
    sshPrivateKeySecretRef:
      name: <my-baremetal-hosts-ssh-private-key>
    manifestsConfigMapRef:
      name: <my-baremetal-cluster-install-manifests>
    imageSetRef:
      name: <my-clusterimageset>
    sshKnownHosts:
    - "10.1.8.90 ecdsa-sha2-nistp256
AAAAE2VjZHNhLXvVVVKUYVkuyvkuygkuyTCYTytfkufTYAAAAIbmlzdHAyNTYAAABB
BKWjJRzeUVuZs4yxSy4eu45xiANFIlbwE3e1aPzGD58x/NX7Yf+S8eFKq4RrsfSaK2hVJyJ
jvVIhUsU9z2sBJP8="
  pullSecretRef:
    name: <my-baremetal-cluster-pull-secret>
```

1.6.2. Amazon Web Services에서 클러스터 생성

Kubernetes 콘솔을 위한 Red Hat Advanced Cluster Management for Kubernetes 콘솔을 사용하여 AWS(Amazon Web Services)에서 Red Hat OpenShift Container Platform 클러스터를 생성할 수 있습니다.

클러스터를 생성할 때 생성 프로세스는 Hive 리소스와 함께 OpenShift Container Platform 설치 프로그램을 사용합니다. 이 절차를 완료한 후 클러스터 생성에 대한 질문이 있는 경우 프로세스에 대한 자세한 내용은 OpenShift Container Platform 설명서의 AWS 에 설치를 참조하십시오.

- 사전 요구 사항
- 콘솔을 사용하여 클러스터 생성
- 기존 클러스터 세트에 클러스터 추가

1.6.2.1. 사전 요구 사항

AWS에서 클러스터를 생성하기 전에 다음 사전 요구 사항을 참조하십시오.

- 배포된 **Red Hat Advanced Cluster Management for Kubernetes hub** 클러스터가 있어야 합니다.
- **Amazon Web Services**에서 **Kubernetes** 클러스터를 생성할 수 있도록 **Red Hat Advanced Cluster Management for Kubernetes hub** 클러스터에 대한 인터넷 액세스가 필요합니다.
- **AWS** 인증 정보가 필요합니다. 자세한 내용은 **Amazon Web Services**에 대한 인증 정보 생성을 참조하십시오.
- **AWS**에서 구성된 도메인이 필요합니다. 도메인을 구성하는 방법에 대한 지침은 **AWS** 계정 구성을 참조하십시오.
- 사용자 이름, 암호, 액세스 키 **ID** 및 시크릿 액세스 키를 포함하는 **AWS(Amazon Web Services)** 로그인 키가 있어야 합니다. 보안 인증 정보 이해 및 가져오기 를 참조하십시오.
- **OpenShift Container Platform** 이미지 풀 시크릿이 있어야 합니다. 이미지 풀 시크릿 사용을 참조하십시오.

참고: 클라우드 공급자 액세스 키를 변경하는 경우 프로비저닝된 클러스터 액세스 키를 수동으로 업데이트해야 합니다. 자세한 내용은 알려진 문제에서 프로비저닝된 클러스터에 대한 자동 시크릿 업데이트가 지원되지 않음을 참조하십시오.

1.6.2.2. 콘솔을 사용하여 클러스터 생성

Red Hat Advanced Cluster Management 콘솔에서 클러스터를 생성하려면 **Infrastructure > Clusters** 로 이동합니다. *클러스터* 페이지에서 클러스터 생성을 클릭하고 콘솔의 단계를 완료합니다.

참고: 이 절차는 클러스터를 생성하기 위한 것입니다. 가져올 기존 클러스터가 있는 경우 해당 단계를 위해 대상 관리 클러스터 가져오기를 **hub** 클러스터로 참조하십시오.

인증 정보를 생성해야 하는 경우 자세한 내용은 Amazon Web Services의 인증 정보 생성을 참조하십시오.

클러스터의 이름은 클러스터의 호스트 이름에 사용됩니다.

중요: 클러스터를 생성할 때 Red Hat Advanced Cluster Management 컨트롤러에서 클러스터 및 해당 리소스의 네임스페이스를 생성합니다. 해당 네임스페이스에 해당 클러스터 인스턴스의 리소스만 포함해야 합니다. 클러스터를 삭제하면 네임스페이스와 그 안에 있는 모든 리소스가 삭제됩니다.

tips: 콘솔의 정보를 입력할 때 콘텐츠 업데이트를 보려면 YAML: on 을 선택합니다.

1.6.2.3. 기존 클러스터 세트에 클러스터 추가

기존 클러스터 세트에 클러스터를 추가하려면 클러스터에 대한 올바른 권한이 있어야 합니다. 클러스터를 생성할 때 cluster-admin 권한이 없는 경우 clusterset-admin 권한이 있는 클러스터를 선택해야 합니다. 지정된 클러스터 세트에 대한 올바른 권한이 없으면 클러스터 생성에 실패합니다. 선택할 클러스터 설정 옵션이 없는 경우 clusterset-admin 권한을 클러스터에 제공하기 위해 클러스터 관리자에게 문의하십시오.

관리되는 모든 클러스터는 관리형 클러스터 세트와 연결되어야 합니다. 관리 클러스터를 ManagedClusterSet 에 할당하지 않으면 기본 관리 클러스터 세트에 자동으로 추가됩니다.

AWS 계정으로 구성한 선택한 인증 정보와 연결된 기본 DNS 도메인이 이미 있는 경우 해당 값이 필드에 채워집니다. 값을 덮어쓰는 방식으로 변경할 수 있습니다. 이 이름은 클러스터의 호스트 이름에 사용됩니다. 자세한 내용은 AWS 계정 구성을 참조하십시오.

릴리스 이미지는 클러스터를 생성하는 데 사용되는 OpenShift Container Platform 이미지의 버전을 식별합니다. 사용하려는 버전이 사용 가능한 경우 이미지 목록에서 이미지를 선택할 수 있습니다. 사용하려는 이미지가 표준 이미지가 아닌 경우 사용하려는 이미지에 URL을 입력할 수 있습니다. 릴리스 이미지에 대한 자세한 내용은 릴리스 이미지를 참조하십시오.

노드 풀에는 컨트롤 플레인 풀과 작업자 풀이 포함됩니다. 컨트롤 플레인 노드는 클러스터 활동의 관리를 공유합니다. 정보에는 다음 필드가 포함됩니다.

- 아키텍처: 관리형 클러스터의 아키텍처 유형이 허브 클러스터의 아키텍처와 동일하지 않은 경우 풀에 있는 머신의 명령어 집합 아키텍처의 값을 입력합니다. 유효한 값은 *amd64,ppc64le,s390x*, cover *64* 입니다.

- **zones:** 컨트롤 플레인 풀을 실행할 위치를 지정합니다. 더 분산된 컨트롤 플레인 노드 그룹에 대해 리전 내에서 여러 영역을 선택할 수 있습니다. 더 가까운 영역은 더 빠른 성능을 제공할 수 있지만 더 멀리 떨어져 있는 영역이 더 분산될 수 있습니다.

- 인스턴스 유형: 컨트롤 플레인 노드의 인스턴스 유형을 지정합니다. 인스턴스 생성 후 인스턴스의 유형과 크기를 변경할 수 있습니다.

- 루트 스토리지: 클러스터에 할당할 루트 스토리지의 양을 지정합니다.

작업자 풀에 0개 이상의 작업자 노드를 생성하여 클러스터의 컨테이너 워크로드를 실행할 수 있습니다. 단일 작업자 풀에 있거나 여러 작업자 풀에 배포할 수 있습니다. 작업자 노드가 0개 지정되면 컨트롤 플레인 노드도 작업자 노드로 작동합니다. 선택적 정보에는 다음 필드가 포함됩니다.

- **zones:** 작업자 풀을 실행할 위치를 지정합니다. 더 분산된 노드 그룹에 대해 리전 내에서 여러 영역을 선택할 수 있습니다. 더 가까운 영역은 더 빠른 성능을 제공할 수 있지만 더 멀리 떨어져 있는 영역이 더 분산될 수 있습니다.

- 인스턴스 유형: 작업자 풀의 인스턴스 유형을 지정합니다. 인스턴스 생성 후 인스턴스의 유형과 크기를 변경할 수 있습니다.

- 노드 수: 작업자 풀의 노드 수를 지정합니다. 이 설정은 작업자 풀을 정의할 때 필요합니다.

- 루트 스토리지: 작업자 풀에 할당된 루트 스토리지의 양을 지정합니다. 이 설정은 작업자 풀을 정의할 때 필요합니다.

클러스터에 네트워킹 세부 정보가 필요하며 IPv6를 사용하는 데 여러 네트워크가 필요합니다. 네트워크 추가를 클릭하여 추가 네트워크를 추가할 수 있습니다.

인증 정보에서 제공되는 프록시 정보는 프록시 필드에 자동으로 추가됩니다. 정보를 그대로 사용하거나 덮어쓰거나 프록시를 활성화하려면 정보를 추가할 수 있습니다. 다음 목록에는 프록시 생성에 필요한 정보가 포함되어 있습니다.

- HTTP 프록시 URL: HTTP 트래픽의 프록시로 사용해야 하는 URL을 지정합니다.

-

HTTPS 프록시 **URL: HTTPS** 트래픽에 사용해야 하는 보안 프록시 **URL**을 지정합니다. 값을 제공하지 않으면 **HTTP** 및 **HTTPS** 모두에 대해 **HTTP** 프록시 **URL** 과 동일한 값이 사용됩니다.

- 프록시 도메인이 없음: 프록시를 바이패스해야 하는 쉼표로 구분된 도메인 목록입니다. 해당 도메인에 있는 모든 하위 도메인을 포함하려면 마침표로 도메인 이름을 시작합니다 . 모든 대상에 대한 프록시를 바이패스하려면 별표 * 를 추가합니다.

- 추가 신뢰 번들: 미러 레지스트리에 액세스하는 데 필요한 인증서 파일의 내용을 지정합니다.

클러스터를 생성하기 전에 정보를 검토하고 선택적으로 사용자 지정할 때 **YAML: On** 을 선택하여 패널에서 **install-config.yaml** 파일 콘텐츠를 볼 수 있습니다. 업데이트가 있는 경우 사용자 지정 설정으로 **YAML** 파일을 편집할 수 있습니다.

참고: 클러스터를 가져오기 위해 클러스터 세부 정보와 함께 제공된 **kubectl** 명령을 실행할 필요가 없습니다. 클러스터를 생성할 때 **Red Hat Advanced Cluster Management**의 관리 하에 자동으로 구성됩니다.

클러스터에 액세스하는 방법에 대한 지침을 보려면 클러스터에 계속 액세스합니다.

1.6.3. Microsoft Azure에서 클러스터 생성

Red Hat Advanced Cluster Management for Kubernetes 콘솔을 사용하여 **Microsoft Azure** 또는 **Microsoft Azure Government**에 **Red Hat OpenShift Container Platform** 클러스터를 배포할 수 있습니다.

클러스터를 생성할 때 생성 프로세스는 **Hive** 리소스와 함께 **OpenShift Container Platform** 설치 프로그램을 사용합니다. 이 절차를 완료한 후 클러스터 생성에 대한 질문이 있는 경우 프로세스에 대한 자세한 내용은 **OpenShift Container Platform** 설명서의 **Azure** 에 설치를 참조하십시오.

- 사전 요구 사항

- 콘솔을 사용하여 클러스터 생성

- 기존 클러스터 세트에 클러스터 추가

1.6.3.1. 사전 요구 사항

Azure에서 클러스터를 생성하기 전에 다음 사전 요구 사항을 참조하십시오.

- 배포된 **Red Hat Advanced Cluster Management for Kubernetes hub** 클러스터가 있어야 합니다.
- **Azure** 또는 **Azure Government**에서 **Kubernetes** 클러스터를 생성할 수 있도록 **Red Hat Advanced Cluster Management for Kubernetes Hub** 클러스터에 대한 인터넷 액세스가 필요합니다.
- **Azure** 인증 정보가 필요합니다. 자세한 내용은 **Microsoft Azure**에 대한 인증 정보 생성을 참조하십시오.
- **Azure** 또는 **Azure Government**에 구성된 도메인이 필요합니다. 도메인 구성 방법에 대한 지침은 **Azure** 클라우드 서비스의 사용자 정의 도메인 이름 구성을 참조하십시오.
- 사용자 이름과 암호를 포함하는 **Azure** 로그인 자격 증명이 필요합니다. **Microsoft Azure** 포털 을 참조하십시오.
- **clientId,clientSecret, tenantId** 를 포함하는 **Azure** 서비스 주체가 필요합니다. **azure.microsoft.com** 을 참조하십시오.
- **OpenShift Container Platform** 이미지 풀 시크릿이 필요합니다. 이미지 풀 시크릿 사용을 참조하십시오.

참고: 클라우드 공급자 액세스 키를 변경하는 경우 프로비저닝된 클러스터 액세스 키를 수동으로 업데이트해야 합니다. 자세한 내용은 알려진 문제에서 프로비저닝된 클러스터에 대한 자동 시크릿 업데이트가 지원되지 않음을 참조하십시오.

1.6.3.2. 콘솔을 사용하여 클러스터 생성

Kubernetes 콘솔용 **Red Hat Advanced Cluster Management**에서 클러스터를 생성하려면 **Infrastructure > Clusters** 로 이동합니다. *클러스터* 페이지에서 클러스터 생성을 클릭하고 콘솔의 단계를 완료합니다.

참고: 이 절차는 클러스터를 생성하기 위한 것입니다. 가져올 기존 클러스터가 있는 경우 해당 단계를 위해 대상 관리 클러스터 가져오기를 **hub** 클러스터로 참조하십시오.

자세한 내용은 **Microsoft Azure**에 대한 인증 정보 생성을 참조하십시오.

클러스터의 이름은 클러스터의 호스트 이름에 사용됩니다.

중요: 클러스터를 생성할 때 **Red Hat Advanced Cluster Management** 컨트롤러에서 클러스터 및 해당 리소스의 네임스페이스를 생성합니다. 해당 네임스페이스에 해당 클러스터 인스턴스의 리소스만 포함해야 합니다. 클러스터를 삭제하면 네임스페이스와 그 안에 있는 모든 리소스가 삭제됩니다.

tips: 콘솔의 정보를 입력할 때 콘텐츠 업데이트를 보려면 **YAML: on** 을 선택합니다.

1.6.3.3. 기존 클러스터 세트에 클러스터 추가

기존 클러스터 세트에 클러스터를 추가하려면 클러스터에 대한 올바른 권한이 있어야 합니다. 클러스터를 생성할 때 **cluster-admin** 권한이 없는 경우 **clusterset-admin** 권한이 있는 클러스터를 선택해야 합니다. 지정된 클러스터 세트에 대한 올바른 권한이 없으면 클러스터 생성에 실패합니다. 선택할 클러스터 설정 옵션이 없는 경우 **clusterset-admin** 권한을 클러스터에 제공하기 위해 클러스터 관리자에게 문의하십시오.

관리되는 모든 클러스터는 관리형 클러스터 세트와 연결되어야 합니다. 관리 클러스터를 **ManagedClusterSet** 에 할당하지 않으면 기본 관리 클러스터 세트에 자동으로 추가됩니다.

Azure 계정에 대해 구성한 선택한 인증 정보와 연결된 기본 **DNS** 도메인이 이미 있는 경우 해당 필드에 채워집니다. 값을 덮어쓰는 방식으로 변경할 수 있습니다. 자세한 내용은 **Azure** 클라우드 서비스의 사용자 정의 도메인 이름 구성을 참조하십시오. 이 이름은 클러스터의 호스트 이름에 사용됩니다.

릴리스 이미지는 클러스터를 생성하는 데 사용되는 **OpenShift Container Platform** 이미지의 버전을 식별합니다. 사용하려는 버전이 사용 가능한 경우 이미지 목록에서 이미지를 선택할 수 있습니다. 사용하려는 이미지가 표준 이미지가 아닌 경우 사용하려는 이미지에 **URL**을 입력할 수 있습니다. 릴리스 이미지에 대한 자세한 내용은 릴리스 이미지를 참조하십시오.

노드 풀에는 컨트롤 플레인 풀과 작업자 풀이 포함됩니다. 컨트롤 플레인 노드는 클러스터 활동의 관리를 공유합니다. 정보에는 다음과 같은 선택적 필드가 포함됩니다.

- **region:** 노드 풀을 실행할 리전을 지정합니다. 더 분산된 컨트롤 플레인 노드 그룹에 대해 리전 내에서 여러 영역을 선택할 수 있습니다. 더 가까운 영역은 더 빠른 성능을 제공할 수 있지만 더 멀리 떨어져 있는 영역이 더 분산될 수 있습니다.

- 아키텍처: 관리형 클러스터의 아키텍처 유형이 허브 클러스터의 아키텍처와 동일하지 않은 경우 풀에 있는 머신의 명령어 집합 아키텍처의 값을 입력합니다. 유효한 값은 ***amd64*,*ppc64le*,*s390x***, cover *64* 입니다.

- 컨트롤 플레인 풀에 대한 인스턴스 유형 및 루트 스토리지 할당(필수)입니다. 인스턴스 생성 후 인스턴스의 유형과 크기를 변경할 수 있습니다.

작업자 풀에서 하나 이상의 작업자 노드를 생성하여 클러스터의 컨테이너 워크로드를 실행할 수 있습니다. 단일 작업자 풀에 있거나 여러 작업자 풀에 배포할 수 있습니다. 작업자 노드가 **0**개 지정되면 컨트롤 플레인 노드도 작업자 노드로 작동합니다. 정보에는 다음 필드가 포함됩니다.

- **zones:** 작업자 풀을 실행할 여기에 지정합니다. 더 분산된 노드 그룹에 대해 리전 내에서 여러 영역을 선택할 수 있습니다. 더 가까운 영역은 더 빠른 성능을 제공할 수 있지만 더 멀리 떨어져 있는 영역이 더 분산될 수 있습니다.

- 인스턴스 유형: 인스턴스 생성 후 인스턴스의 유형과 크기를 변경할 수 있습니다.

네트워크 추가를 클릭하여 추가 네트워크를 추가할 수 있습니다. **IPv6** 주소를 사용하는 경우 네트워크가 두 개 이상 있어야 합니다.

인증 정보에서 제공되는 프록시 정보는 프록시 필드에 자동으로 추가됩니다. 정보를 그대로 사용하거나 덮어쓰거나 프록시를 활성화하려면 정보를 추가할 수 있습니다. 다음 목록에는 프록시 생성에 필요한 정보가 포함되어 있습니다.

- **HTTP** 프록시 **URL: HTTP** 트래픽의 프록시로 사용해야 하는 **URL**입니다.

- **HTTPS** 프록시 **URL: HTTPS** 트래픽에 사용해야 하는 보안 프록시 **URL**입니다. 값을 제공하지 않으면 **HTTP** 및 **HTTPS** 모두에 대해 **HTTP** 프록시 **URL** 과 동일한 값이 사용됩니다.

- 프록시 도메인이 없음: 프록시를 바이패스해야 하는 쉼표로 구분된 도메인 목록입니다. 해당 도메인에 있는 모든 하위 도메인을 포함하려면 마침표로 도메인 이름을 시작합니다 . 모든 대상에

대한 프록시를 바이패스하려면 별표 * 를 추가합니다.

- 추가 신뢰 번들: 미러 레지스트리에 액세스하는 데 필요한 인증서 파일의 콘텐츠입니다.

클러스터를 생성하기 전에 정보를 검토하고 선택적으로 사용자 지정할 때 **YAML** 스위치를 클릭하여 패널에서 **install-config.yaml** 파일 콘텐츠를 볼 수 있습니다. 업데이트가 있는 경우 사용자 지정 설정으로 **YAML** 파일을 편집할 수 있습니다.

참고: 클러스터를 가져오기 위해 클러스터 세부 정보와 함께 제공된 **kubectl** 명령을 실행할 필요가 없습니다. 클러스터를 생성할 때 **Red Hat Advanced Cluster Management**의 관리 하에 자동으로 구성됩니다.

클러스터에 액세스하는 방법에 대한 지침을 보려면 클러스터에 계속 액세스합니다.

1.6.4. Google Cloud Platform에서 클러스터 생성

GCP(Google Cloud Platform)에서 **Red Hat OpenShift Container Platform** 클러스터를 생성하는 절차를 따르십시오. **GCP**에 대한 자세한 내용은 **Google Cloud Platform** 을 참조하십시오.

클러스터를 생성할 때 생성 프로세스는 **Hive** 리소스와 함께 **OpenShift Container Platform** 설치 프로그램을 사용합니다. 이 절차를 완료한 후 클러스터 생성에 대한 질문이 있는 경우 프로세스에 대한 자세한 내용은 **OpenShift Container Platform** 설명서의 **GCP** 에 설치를 참조하십시오.

- 사전 요구 사항
- 콘솔을 사용하여 클러스터 생성
- 기존 클러스터 세트에 클러스터 추가

1.6.4.1. 사전 요구 사항

GCP에 클러스터를 생성하기 전에 다음 사전 요구 사항을 참조하십시오.

- 배포된 **Red Hat Advanced Cluster Management for Kubernetes hub** 클러스터가 있어야

합니다.

- GCP에서 Kubernetes 클러스터를 생성할 수 있도록 Red Hat Advanced Cluster Management for Kubernetes hub 클러스터에 대한 인터넷 액세스가 필요합니다.
- GCP 인증 정보가 있어야 합니다. 자세한 내용은 Google Cloud Platform에 대한 인증 정보 생성을 참조하십시오.
- GCP에 구성된 도메인이 있어야 합니다. 도메인 구성 방법에 대한 지침은 사용자 정의 도메인 설정을 참조하십시오.
- 사용자 이름과 암호를 포함하는 GCP 로그인 인증 정보가 필요합니다.
- OpenShift Container Platform 이미지 풀 시크릿이 있어야 합니다. 이미지 풀 시크릿 사용을 참조하십시오.

참고: 클라우드 공급자 액세스 키를 변경하는 경우 프로비저닝된 클러스터 액세스 키를 수동으로 업데이트해야 합니다. 자세한 내용은 알려진 문제에서 프로비저닝된 클러스터에 대한 자동 시크릿 업데이트가 지원되지 않음을 참조하십시오.

1.6.4.2. 콘솔을 사용하여 클러스터 생성

Kubernetes 콘솔용 Red Hat Advanced Cluster Management에서 클러스터를 생성하려면 Infrastructure > Clusters 로 이동합니다. *클러스터* 페이지에서 클러스터 생성을 클릭하고 콘솔의 단계를 완료합니다.

참고: 이 절차는 클러스터를 생성하기 위한 것입니다. 가져올 기존 클러스터가 있는 경우 해당 단계를 위해 대상 관리 클러스터 가져오기를 hub 클러스터로 참조하십시오.

인증 정보를 생성해야 하는 경우 자세한 내용은 Google Cloud Platform의 인증 정보 생성을 참조하십시오.

클러스터의 이름은 클러스터의 호스트 이름에 사용됩니다. GCP 클러스터 이름 지정에 적용되는 몇 가지 제한 사항이 있습니다. 이러한 제한에는 goog 으로 이름을 시작하지 않거나 이름의 모든 위치에서

Google 과 유사한 문자 및 숫자 그룹을 포함하지 않습니다. 전체 제한 목록은 **Bucket** 이름 지정 지침을 참조하십시오.

중요: 클러스터를 생성할 때 **Red Hat Advanced Cluster Management** 컨트롤러에서 클러스터 및 해당 리소스의 네임스페이스를 생성합니다. 해당 네임스페이스에 해당 클러스터 인스턴스의 리소스만 포함해야 합니다. 클러스터를 삭제하면 네임스페이스와 그 안에 있는 모든 리소스가 삭제됩니다.

tips: 콘솔의 정보를 입력할 때 콘텐츠 업데이트를 보려면 **YAML: on** 을 선택합니다.

1.6.4.3. 기존 클러스터 세트에 클러스터 추가

기존 클러스터 세트에 클러스터를 추가하려면 클러스터에 대한 올바른 권한이 있어야 합니다. 클러스터를 생성할 때 **cluster-admin** 권한이 없는 경우 **clusterset-admin** 권한이 있는 클러스터를 선택해야 합니다. 지정된 클러스터 세트에 대한 올바른 권한이 없으면 클러스터 생성에 실패합니다. 선택할 클러스터 설정 옵션이 없는 경우 **clusterset-admin** 권한을 클러스터에 제공하기 위해 클러스터 관리자에게 문의하십시오.

관리되는 모든 클러스터는 관리형 클러스터 세트와 연결되어야 합니다. 관리 클러스터를 **ManagedClusterSet** 에 할당하지 않으면 기본 관리 클러스터 세트에 자동으로 추가됩니다.

GCP 계정에 대해 선택한 인증 정보와 연결된 기본 **DNS** 도메인이 이미 있는 경우 해당 값이 필드에 채워집니다. 값을 덮어쓰는 방식으로 변경할 수 있습니다. 자세한 내용은 사용자 정의 도메인 설정을 참조하십시오. 이 이름은 클러스터의 호스트 이름에 사용됩니다.

릴리스 이미지는 클러스터를 생성하는 데 사용되는 **OpenShift Container Platform** 이미지의 버전을 식별합니다. 사용하려는 버전이 사용 가능한 경우 이미지 목록에서 이미지를 선택할 수 있습니다. 사용하려는 이미지가 표준 이미지가 아닌 경우 사용하려는 이미지에 **URL**을 입력할 수 있습니다. 릴리스 이미지에 대한 자세한 내용은 릴리스 이미지를 참조하십시오.

노드 풀에는 컨트롤 플레인 풀과 작업자 풀이 포함됩니다. 컨트롤 플레인 노드는 클러스터 활동의 관리를 공유합니다. 정보에는 다음 필드가 포함됩니다.

- **region:** 컨트롤 플레인 풀을 실행할 리전을 지정합니다. 더 가까운 지역이 더 빠른 성능을 제공할 수 있지만 더 멀리 있는 영역이 더 분산될 수 있습니다.
- 아키텍처: 관리형 클러스터의 아키텍처 유형이 허브 클러스터의 아키텍처와 동일하지 않은 경우 풀에 있는 머신의 명령어 집합 아키텍처의 값을 입력합니다. 유효한 값은

amd64,ppc64le,s390x, cover *64* 입니다.

- 인스턴스 유형: 인스턴스 생성 후 인스턴스의 유형과 크기를 변경할 수 있습니다.

작업자 풀에서 하나 이상의 작업자 노드를 생성하여 클러스터의 컨테이너 워크로드를 실행할 수 있습니다. 단일 작업자 풀에 있거나 여러 작업자 풀에 배포할 수 있습니다. 작업자 노드가 0개 지정되면 컨트롤 플레인 노드도 작업자 노드로 작동합니다. 정보에는 다음 필드가 포함됩니다.

- 인스턴스 유형: 인스턴스 생성 후 인스턴스의 유형과 크기를 변경할 수 있습니다.
- 노드 수: 이 설정은 작업자 풀을 정의할 때 필요합니다.

네트워킹 세부 정보가 필요하며 IPv6 주소를 사용하는 데 여러 네트워크가 필요합니다. 네트워크 추가를 클릭하여 추가 네트워크를 추가할 수 있습니다.

인증 정보에서 제공되는 프록시 정보는 프록시 필드에 자동으로 추가됩니다. 정보를 그대로 사용하거나 덮어쓰거나 프록시를 활성화하려면 정보를 추가할 수 있습니다. 다음 목록에는 프록시 생성에 필요한 정보가 포함되어 있습니다.

- HTTP 프록시 URL: HTTP 트래픽의 프록시로 사용해야 하는 URL입니다.
- HTTPS 프록시 URL: HTTPS 트래픽에 사용해야 하는 보안 프록시 URL입니다. 값을 제공하지 않으면 HTTP 및 HTTPS 모두에 대해 HTTP 프록시 URL 과 동일한 값이 사용됩니다.
- 프록시 도메인이 없음: 프록시를 바이패스해야 하는 쉼표로 구분된 도메인 목록입니다. 해당 도메인에 있는 모든 하위 도메인을 포함하려면 마침표로 도메인 이름을 시작합니다 . 모든 대상에 대한 프록시를 바이패스하려면 별표 * 를 추가합니다.
- 추가 신뢰 번들: 미러 레지스트리에 액세스하는 데 필요한 인증서 파일의 콘텐츠입니다.

클러스터를 생성하기 전에 정보를 검토하고 선택적으로 사용자 지정할 때 YAML: On 을 선택하여 패널에서 install-config.yaml 파일 콘텐츠를 볼 수 있습니다. 업데이트가 있는 경우 사용자 지정 설정으로 YAML 파일을 편집할 수 있습니다.

참고: 클러스터를 가져오기 위해 클러스터 세부 정보와 함께 제공된 **kubectl** 명령을 실행할 필요가 없습니다. 클러스터를 생성할 때 **Red Hat Advanced Cluster Management**의 관리 하에 자동으로 구성됩니다.

클러스터에 액세스하는 방법에 대한 지침을 보려면 클러스터에 계속 액세스합니다.

1.6.5. VMware vSphere에서 클러스터 생성

Red Hat Advanced Cluster Management for Kubernetes 콘솔을 사용하여 **VMware vSphere**에 **Red Hat OpenShift Container Platform** 클러스터를 배포할 수 있습니다.

클러스터를 생성할 때 생성 프로세스는 **Hive** 리소스와 함께 **OpenShift Container Platform** 설치 프로그램을 사용합니다. 이 절차를 완료한 후 클러스터 생성에 대한 질문이 있는 경우 프로세스에 대한 자세한 내용은 **OpenShift Container Platform** 설명서의 **vSphere** 에 설치를 참조하십시오.

- 사전 요구 사항
- 콘솔을 사용하여 클러스터 생성
- 기존 클러스터 세트에 클러스터 추가

1.6.5.1. 사전 요구 사항

vSphere에서 클러스터를 생성하기 전에 다음 사전 요구 사항을 참조하십시오.

- **OpenShift Container Platform** 버전 **4.6** 이상에 배포된 **Red Hat Advanced Cluster Management hub** 클러스터가 있어야 합니다.
- **vSphere**에서 **Kubernetes** 클러스터를 생성하려면 **Red Hat Advanced Cluster Management hub** 클러스터에 대한 인터넷 액세스가 필요합니다.
- **vSphere** 인증 정보가 필요합니다. 자세한 내용은 **VMware vSphere**에 대한 인증 정보 생성을 참조하십시오.

- OpenShift Container Platform 이미지 풀 시크릿이 필요합니다. 이미지 풀 시크릿 사용을 참조하십시오.

- 배포 중인 VMware 인스턴스에 대해 다음 정보가 있어야 합니다.

  - API 및 Ingress 인스턴스에 필요한 고정 IP 주소

  - 다음을 위한 DNS 레코드

    - api.<cluster_name>.<base_domain >은 정적 API VIP를 가리켜야 합니다.

    - *.apps.<cluster_name>.<base_domain >은 Ingress VIP의 고정 IP 주소를 가리켜야 합니다.

참고: VMware vSphere 또는 Red Hat OpenStack Platform 공급자 및 연결이 끊긴 설치 공급자를 사용하여 클러스터를 생성할 때 미러 레지스트리에 액세스하는 데 인증서가 필요한 경우 *연결이 끊긴 설치의 구성 섹션에* 인증 정보의 *추가 신뢰 번들* 필드에 입력해야 합니다. 클러스터 생성 콘솔 편집기에 입력할 수 없습니다.

1.6.5.2. 콘솔을 사용하여 클러스터 생성

Kubernetes 콘솔용 Red Hat Advanced Cluster Management에서 클러스터를 생성하려면 Infrastructure > Clusters 로 이동합니다. *클러스터* 페이지에서 클러스터 생성을 클릭하고 콘솔의 단계를 완료합니다.

참고: 이 절차는 클러스터를 생성하기 위한 것입니다. 가져올 기존 클러스터가 있는 경우 해당 단계를 위해 대상 관리 클러스터 가져오기를 hub 클러스터로 참조하십시오.

인증 정보를 생성해야 하는 경우 인증 정보 생성에 대한 자세한 내용은 VMware vSphere 의 인증 정보 생성을 참조하십시오.

클러스터의 이름은 클러스터의 호스트 이름에 사용됩니다.

중요: 클러스터를 생성할 때 Red Hat Advanced Cluster Management 컨트롤러에서 클러스터 및 해

당 리소스의 네임스페이스를 생성합니다. 해당 네임스페이스에 해당 클러스터 인스턴스의 리소스만 포함해야 합니다. 클러스터를 삭제하면 네임스페이스와 그 안에 있는 모든 리소스가 삭제됩니다.

tips: 콘솔의 정보를 입력할 때 콘텐츠 업데이트를 보려면 **YAML: on** 을 선택합니다.

1.6.5.3. 기존 클러스터 세트에 클러스터 추가

기존 클러스터 세트에 클러스터를 추가하려면 클러스터에 대한 올바른 권한이 있어야 합니다. 클러스터를 생성할 때 **cluster-admin** 권한이 없는 경우 **clusterset-admin** 권한이 있는 클러스터를 선택해야 합니다. 지정된 클러스터 세트에 대한 올바른 권한이 없으면 클러스터 생성에 실패합니다. 선택할 클러스터 설정 옵션이 없는 경우 **clusterset-admin** 권한을 클러스터에 제공하기 위해 클러스터 관리자에게 문의하십시오.

관리되는 모든 클러스터는 관리형 클러스터 세트와 연결되어야 합니다. 관리 클러스터를 **ManagedClusterSet** 에 할당하지 않으면 기본 관리 클러스터 세트에 자동으로 추가됩니다.

vSphere 계정에 대해 구성한 선택한 인증 정보와 연결된 기본 도메인이 이미 있는 경우 해당 값이 필드에 채워집니다. 값을 덮어쓰는 방식으로 변경할 수 있습니다. 자세한 내용은 사용자 지정으로 **vSphere** 에 클러스터 설치를 참조하십시오. 이 값은 사전 요구 사항 섹션에 나열된 **DNS** 레코드를 만드는 데 사용한 이름과 일치해야 합니다. 이 이름은 클러스터의 호스트 이름에 사용됩니다.

릴리스 이미지는 클러스터를 생성하는 데 사용되는 **OpenShift Container Platform** 이미지의 버전을 식별합니다. 사용하려는 버전이 사용 가능한 경우 이미지 목록에서 이미지를 선택할 수 있습니다. 사용하려는 이미지가 표준 이미지가 아닌 경우 사용하려는 이미지에 **URL**을 입력할 수 있습니다. 릴리스 이미지에 대한 자세한 내용은 릴리스 이미지를 참조하십시오.

참고: **OpenShift Container Platform** 버전 **4.5.x** 이상의 릴리스 이미지만 지원됩니다.

노드 풀에는 컨트롤 플레인 풀과 작업자 풀이 포함됩니다. 컨트롤 플레인 노드는 클러스터 활동의 관리를 공유합니다. 정보에는 *아키텍처* 필드가 포함됩니다. 다음 필드 설명을 확인합니다.

- 아키텍처: 관리형 클러스터의 아키텍처 유형이 허브 클러스터의 아키텍처와 동일하지 않은 경우 풀에 있는 머신의 명령어 집합 아키텍처의 값을 입력합니다. 유효한 값은 ***amd64,ppc64le,s390x***, cover ***64*** 입니다.

작업자 풀에서 하나 이상의 작업자 노드를 생성하여 클러스터의 컨테이너 워크로드를 실행할 수 있습니다. 단일 작업자 풀에 있거나 여러 작업자 풀에 배포할 수 있습니다. 작업자 노드가 **0**개 지정되면 컨트롤

플레인 노드도 작업자 노드로 작동합니다. 이 정보에는 *소켓당 코어*,*CPU*,*Memory_minMB*, *_Disk 크기* (GiB) 및 *노드 수가* 포함됩니다.

네트워킹 정보가 필요합니다. IPv6를 사용하려면 여러 네트워크가 필요합니다. 필수 네트워킹 정보 중 일부는 다음 필드를 포함합니다.

- vSphere 네트워크 이름: VMware vSphere 네트워크 이름을 지정합니다.
- API VIP: 내부 API 통신에 사용할 IP 주소를 지정합니다.

  참고: 이 값은 사전 요구 사항 섹션에 나열된 DNS 레코드를 만드는 데 사용한 이름과 일치해야 합니다. 제공되지 않는 경우 api. 가 올바르게 확인되도록 DNS를 사전 구성해야 합니다.
- Ingress VIP: 인그레스 트래픽에 사용할 IP 주소를 지정합니다.

  참고: 이 값은 사전 요구 사항 섹션에 나열된 DNS 레코드를 만드는 데 사용한 이름과 일치해야 합니다. 제공되지 않는 경우 test.apps 가 올바르게 확인되도록 DNS를 사전 구성해야 합니다.

네트워크 추가를 클릭하여 추가 네트워크를 추가할 수 있습니다. IPv6 주소를 사용하는 경우 네트워크가 두 개 이상 있어야 합니다.

인증 정보에서 제공되는 프록시 정보는 프록시 필드에 자동으로 추가됩니다. 정보를 그대로 사용하거나 덮어쓰거나 프록시를 활성화하려면 정보를 추가할 수 있습니다. 다음 목록에는 프록시 생성에 필요한 정보가 포함되어 있습니다.

- HTTP 프록시 URL: HTTP 트래픽의 프록시로 사용해야 하는 URL을 지정합니다.
- HTTPS 프록시 URL: HTTPS 트래픽에 사용해야 하는 보안 프록시 URL을 지정합니다. 값을 제공하지 않으면 HTTP 및 HTTPS 모두에 대해 HTTP 프록시 URL 과 동일한 값이 사용됩니다.
- 프록시 도메인이 없음: 프록시를 바이패스해야 하는 쉼표로 구분된 도메인 목록을 제공합니다. 해당 도메인에 있는 모든 하위 도메인을 포함하려면 마침표로 도메인 이름을 시작합니다 . 모든 대상에 대한 프록시를 바이패스하려면 별표 * 를 추가합니다.
- 추가 신뢰 번들: 미러 레지스트리에 액세스하는 데 필요한 인증서 파일의 내용을 지정합니

다.

설치 분리를 클릭하여 연결 해제된 설치 이미지를 정의할 수 있습니다. 클러스터 생성에 대한 연결되지 않은 설치 설정을 입력할 수 없거나 제한 사항에 대한 자세한 내용은 입력한 경우 무시됩니다.

자동화 템플릿 추가를 클릭하여 템플릿을 생성할 수 있습니다.

클러스터를 생성하기 전에 정보를 검토하고 선택적으로 사용자 지정할 때 **YAML** 스위치를 클릭하여 패널에서 **install-config.yaml** 파일 콘텐츠를 볼 수 있습니다. 업데이트가 있는 경우 사용자 지정 설정으로 **YAML** 파일을 편집할 수 있습니다.

참고: 클러스터를 가져오기 위해 클러스터 세부 정보와 함께 제공된 **kubectl** 명령을 실행할 필요가 없습니다. 클러스터를 생성할 때 **Red Hat Advanced Cluster Management**의 관리 하에 자동으로 구성됩니다.

클러스터에 액세스하는 방법에 대한 지침을 보려면 클러스터에 계속 액세스합니다.

1.6.6. Red Hat OpenStack Platform에서 클러스터 생성

Red Hat Advanced Cluster Management for Kubernetes 콘솔을 사용하여 **Red Hat OpenStack Platform**에 **Red Hat OpenShift Container Platform** 클러스터를 배포할 수 있습니다.

클러스터를 생성할 때 생성 프로세스는 **Hive** 리소스와 함께 **OpenShift Container Platform** 설치 프로그램을 사용합니다. 이 절차를 완료한 후 클러스터 생성에 대한 질문이 있는 경우 프로세스에 대한 자세한 내용은 **OpenShift Container Platform** 설명서의 **OpenStack** 에 설치를 참조하십시오.

- 사전 요구 사항
- 콘솔을 사용하여 클러스터 생성
- 기존 클러스터 세트에 클러스터 추가

1.6.6.1. 사전 요구 사항

Red Hat OpenStack Platform에서 클러스터를 생성하기 전에 다음 사전 요구 사항을 참조하십시오.

- OpenShift Container Platform 버전 4.6 이상에 배포된 Red Hat Advanced Cluster Management hub 클러스터가 있어야 합니다.
- Red Hat OpenStack Platform에서 Kubernetes 클러스터를 생성할 수 있도록 Red Hat Advanced Cluster Management hub 클러스터에 대한 인터넷 액세스가 필요합니다.
- Red Hat OpenStack Platform 인증 정보가 있어야 합니다. 자세한 내용은 Red Hat OpenStack Platform의 인증 정보 생성을 참조하십시오.
- OpenShift Container Platform 이미지 풀 시크릿이 필요합니다. 이미지 풀 시크릿 사용을 참조하십시오.
- 배포 중인 Red Hat OpenStack Platform 인스턴스에 대해 다음 정보가 필요합니다.
  - 컨트롤 플레인 및 작업자 인스턴스의 플레이버 이름입니다(예: m1.xlarge).
  - 유동 IP 주소를 제공하는 외부 네트워크의 네트워크 이름
  - API 및 Ingress 인스턴스에 필요한 부동 IP 주소
  - 다음을 위한 DNS 레코드
    - api.<cluster_name>.<base_domain > .이는 API의 부동 IP 주소를 가리켜야 합니다.
    - *.apps.<cluster_name>.<base_domain > , 수신의 부동 IP 주소를 가리켜야 합니다.

1.6.6.2. 콘솔을 사용하여 클러스터 생성

Kubernetes 콘솔용 **Red Hat Advanced Cluster Management**에서 클러스터를 생성하려면 **Infrastructure > Clusters** 로 이동합니다. *클러스터* 페이지에서 클러스터 생성을 클릭하고 콘솔의 단계를 완료합니다.

참고: 이 절차는 클러스터를 생성하기 위한 것입니다. 가져올 기존 클러스터가 있는 경우 해당 단계를 위해 대상 관리 클러스터 가져오기를 **hub** 클러스터로 참조하십시오.

인증 정보를 생성해야 하는 경우 자세한 내용은 **Red Hat OpenStack Platform**의 인증 정보 생성을 참조하십시오.

클러스터의 이름은 클러스터의 호스트 이름에 사용됩니다. 이름에는 **15**자 미만이 포함되어야 합니다. 이 값은 자격 증명 사전 요구 사항 섹션에 나열된 **DNS** 레코드를 생성하는 데 사용한 이름과 일치해야 합니다.

중요: 클러스터를 생성할 때 **Red Hat Advanced Cluster Management** 컨트롤러에서 클러스터 및 해당 리소스의 네임스페이스를 생성합니다. 해당 네임스페이스에 해당 클러스터 인스턴스의 리소스만 포함해야 합니다. 클러스터를 삭제하면 네임스페이스와 그 안에 있는 모든 리소스가 삭제됩니다.

tips: 콘솔의 정보를 입력할 때 콘텐츠 업데이트를 보려면 **YAML: on** 을 선택합니다.

1.6.6.3. 기존 클러스터 세트에 클러스터 추가

기존 클러스터 세트에 클러스터를 추가하려면 클러스터에 대한 올바른 권한이 있어야 합니다. 클러스터를 생성할 때 **cluster-admin** 권한이 없는 경우 **clusterset-admin** 권한이 있는 클러스터를 선택해야 합니다. 지정된 클러스터 세트에 대한 올바른 권한이 없으면 클러스터 생성에 실패합니다. 선택할 클러스터 설정 옵션이 없는 경우 **clusterset-admin** 권한을 클러스터에 제공하기 위해 클러스터 관리자에게 문의하십시오.

관리되는 모든 클러스터는 관리형 클러스터 세트와 연결되어야 합니다. 관리 클러스터를 **ManagedClusterSet** 에 할당하지 않으면 기본 관리 클러스터 세트에 자동으로 추가됩니다.

Red Hat OpenStack Platform 계정에 대해 구성한 선택한 인증 정보와 연결된 기본 **DNS** 도메인이 이미 있는 경우 해당 값이 필드에 채워집니다. 값을 덮어쓰는 방식으로 변경할 수 있습니다. 자세한 내용은 **Red** Hat **OpenStack Platform** 설명서의 도메인 관리를 참조하십시오. 이 이름은 클러스터의 호스트 이름에 사용됩니다.

릴리스 이미지는 클러스터를 생성하는 데 사용되는 **OpenShift Container Platform** 이미지의 버전을 식별합니다. 사용하려는 버전이 사용 가능한 경우 이미지 목록에서 이미지를 선택할 수 있습니다. 사용하

려는 이미지가 표준 이미지가 아닌 경우 사용하려는 이미지에 URL을 입력할 수 있습니다. 릴리스 이미지에 대한 자세한 내용은 릴리스 이미지를 참조하십시오. OpenShift Container Platform 버전 4.6.x 이상의 릴리스 이미지만 지원됩니다.

노드 풀에는 컨트롤 플레인 풀과 작업자 풀이 포함됩니다. 컨트롤 플레인 노드는 클러스터 활동의 관리를 공유합니다. 정보에는 다음 필드가 포함됩니다.

- 선택 사항: 관리형 클러스터의 아키텍처 유형이 hub 클러스터의 아키텍처와 동일하지 않은 경우 풀에 있는 머신의 명령어 집합 아키텍처의 값을 입력합니다. 유효한 값은 *amd64*,*ppc64le*,*s390x*, cover *64* 입니다.

- 컨트롤 플레인 풀의 인스턴스 유형: 인스턴스 생성 후 인스턴스의 유형 및 크기를 변경할 수 있습니다.

작업자 풀에서 하나 이상의 작업자 노드를 생성하여 클러스터의 컨테이너 워크로드를 실행할 수 있습니다. 단일 작업자 풀에 있거나 여러 작업자 풀에 배포할 수 있습니다. 작업자 노드가 0개 지정되면 컨트롤 플레인 노드도 작업자 노드로 작동합니다. 정보에는 다음 필드가 포함됩니다.

- 인스턴스 유형: 인스턴스 생성 후 인스턴스의 유형과 크기를 변경할 수 있습니다.

- 노드 수: 작업자 풀의 노드 수를 지정합니다. 이 설정은 작업자 풀을 정의할 때 필요합니다.

클러스터에 대한 네트워킹 세부 정보가 필요합니다. IPv4 네트워크의 하나 이상의 네트워크에 대한 값을 제공해야 합니다. IPv6 네트워크의 경우 둘 이상의 네트워크를 정의해야 합니다.

네트워크 추가를 클릭하여 추가 네트워크를 추가할 수 있습니다. IPv6 주소를 사용하는 경우 네트워크가 두 개 이상 있어야 합니다.

인증 정보에서 제공되는 프록시 정보는 프록시 필드에 자동으로 추가됩니다. 정보를 그대로 사용하거나 덮어쓰거나 프록시를 활성화하려면 정보를 추가할 수 있습니다. 다음 목록에는 프록시 생성에 필요한 정보가 포함되어 있습니다.

- HTTP 프록시 URL: HTTP 트래픽의 프록시로 사용해야 하는 URL을 지정합니다.

- HTTPS 프록시 URL: HTTPS 트래픽에 사용해야 하는 보안 프록시 URL입니다. 값을 제공하

지 않으면 **HTTP** 및 **HTTPS** 모두에 대해 **HTTP** 프록시 **URL** 과 동일한 값이 사용됩니다.

- 프록시 도메인이 없음: 프록시를 바이패스해야 하는 쉼표로 구분된 도메인 목록을 정의합니다. 해당 도메인에 있는 모든 하위 도메인을 포함하려면 마침표로 도메인 이름을 시작합니다 . 모든 대상에 대한 프록시를 바이패스하려면 별표 * 를 추가합니다.

- 추가 신뢰 번들: 미러 레지스트리에 액세스하는 데 필요한 인증서 파일의 내용을 지정합니다.

설치 분리를 클릭하여 연결 해제된 설치 이미지를 정의할 수 있습니다. 클러스터 생성에 대한 연결되지 않은 설치 설정을 입력할 수 없거나 제한 사항에 대한 자세한 내용은 입력한 경우 무시됩니다.

클러스터를 생성하기 전에 정보를 검토하고 선택적으로 사용자 지정할 때 **YAML** 스위치를 클릭하여 패널에서 **install-config.yaml** 파일 콘텐츠를 볼 수 있습니다. 업데이트가 있는 경우 사용자 지정 설정으로 **YAML** 파일을 편집할 수 있습니다.

참고: 클러스터를 가져오기 위해 클러스터 세부 정보와 함께 제공된 **kubectl** 명령을 실행할 필요가 없습니다. 클러스터를 생성할 때 **Red Hat Advanced Cluster Management**의 관리 하에 자동으로 구성됩니다.

클러스터에 액세스하는 방법에 대한 지침을 보려면 클러스터에 계속 액세스합니다.

1.6.7. Red Hat Virtualization에서 클러스터 생성

Red Hat Advanced Cluster Management for Kubernetes 콘솔을 사용하여 **Red Hat Virtualization** 에서 **Red Hat OpenShift Container Platform** 클러스터를 생성할 수 있습니다.

클러스터를 생성할 때 생성 프로세스는 **Hive** 리소스와 함께 **OpenShift Container Platform** 설치 프로그램을 사용합니다. 이 절차를 완료한 후 클러스터 생성에 대한 질문이 있는 경우 프로세스에 대한 자세한 내용은 **OpenShift Container Platform** 설명서의 **RHV** 에 설치를 참조하십시오.

- 사전 요구 사항

- 콘솔을 사용하여 클러스터 생성

- 기존 클러스터 세트에 클러스터 추가

1.6.7.1. 사전 요구 사항

Red Hat Virtualization에 클러스터를 생성하기 전에 다음 사전 요구 사항을 참조하십시오.

- 배포된 **Red Hat Advanced Cluster Management for Kubernetes hub** 클러스터가 있어야 합니다.
- **Red Hat Virtualization**에서 **Kubernetes** 클러스터를 생성할 수 있도록 **Red Hat Advanced Cluster Management for Kubernetes Hub** 클러스터에 대한 인터넷 액세스가 필요합니다.
- **Red Hat Virtualization** 인증 정보가 필요합니다. 자세한 내용은 **Red Hat Virtualization**에 대한 인증 정보 생성을 참조하십시오.
- **oVirt Engine** 가상 머신에 대해 구성된 도메인 및 가상 머신 프록시가 필요합니다. 도메인 구성 방법에 대한 자세한 내용은 **Red Hat OpenShift Container Platform** 설명서의 **RHV** 에 설치를 참조하십시오.
- **Red Hat** 고객 포털 사용자 이름 및 암호를 포함하는 **Red Hat Virtualization** 로그인 자격 증명이 있어야 합니다.
- **OpenShift Container Platform** 이미지 풀 시크릿이 필요합니다. 풀 시크릿은 **Pull secret**에서 다운로드할 수 있습니다. 가져오기 보안에 대한 자세한 내용은 이미지 풀 시크릿 사용을 참조하십시오.

참고: 클라우드 공급자 액세스 키를 변경하는 경우 프로비저닝된 클러스터 액세스 키를 수동으로 업데이트해야 합니다. 자세한 내용은 알려진 문제에서 프로비저닝된 클러스터에 대한 자동 시크릿 업데이트가 지원되지 않음을 참조하십시오.

1.6.7.2. 콘솔을 사용하여 클러스터 생성

Kubernetes 콘솔용 **Red Hat Advanced Cluster Management**에서 클러스터를 생성하려면 **Infrastructure > Clusters** 로 이동합니다. *클러스터* 페이지에서 클러스터 생성을 클릭하고 콘솔의 단계를 완료합니다.

참고: 이 절차는 클러스터를 생성하기 위한 것입니다. 가져올 기존 클러스터가 있는 경우 해당 단계를 위해 대상 관리 클러스터 가져오기를 hub 클러스터로 참조하십시오.

인증 정보를 생성해야 하는 경우 자세한 내용은 Red Hat Virtualization에 대한 인증 정보 생성을 참조하십시오.

클러스터의 이름은 클러스터의 호스트 이름에 사용됩니다.

중요: 클러스터를 생성할 때 Red Hat Advanced Cluster Management 컨트롤러에서 클러스터 및 해당 리소스의 네임스페이스를 생성합니다. 해당 네임스페이스에 해당 클러스터 인스턴스의 리소스만 포함해야 합니다. 클러스터를 삭제하면 네임스페이스와 그 안에 있는 모든 리소스가 삭제됩니다.

tips: 콘솔의 정보를 입력할 때 콘텐츠 업데이트를 보려면 YAML: on 을 선택합니다.

1.6.7.3. 기존 클러스터 세트에 클러스터 추가

기존 클러스터 세트에 클러스터를 추가하려면 클러스터에 대한 올바른 권한이 있어야 합니다. 클러스터를 생성할 때 cluster-admin 권한이 없는 경우 clusterset-admin 권한이 있는 클러스터를 선택해야 합니다. 지정된 클러스터 세트에 대한 올바른 권한이 없으면 클러스터 생성에 실패합니다. 선택할 클러스터 설정 옵션이 없는 경우 clusterset-admin 권한을 클러스터에 제공하기 위해 클러스터 관리자에게 문의하십시오.

관리되는 모든 클러스터는 관리형 클러스터 세트와 연결되어야 합니다. 관리 클러스터를 ManagedClusterSet 에 할당하지 않으면 기본 관리 클러스터 세트에 자동으로 추가됩니다.

Red Hat Virtualization 계정에 대해 구성한 선택한 인증 정보와 연결된 기본 DNS 도메인이 이미 있는 경우 해당 필드에 채워집니다. 값을 덮어쓰고 변경할 수 있습니다.

릴리스 이미지는 클러스터를 생성하는 데 사용되는 OpenShift Container Platform 이미지의 버전을 식별합니다. 사용하려는 버전이 사용 가능한 경우 이미지 목록에서 이미지를 선택할 수 있습니다. 사용하려는 이미지가 표준 이미지가 아닌 경우 사용하려는 이미지에 URL을 입력할 수 있습니다. 릴리스 이미지에 대한 자세한 내용은 릴리스 이미지를 참조하십시오.

노드 풀 정보에는 컨트롤 플레인 풀의 코어, 소켓, 메모리 및 디스크 크기가 포함됩니다. 세 개의 컨트롤 플레인 노드는 클러스터 활동 관리를 공유합니다. 정보에는 *아키텍처* 필드가 포함됩니다. 다음 필드 설명을 확인합니다.

- 아키텍처: 관리형 클러스터의 아키텍처 유형이 허브 클러스터의 아키텍처와 동일하지 않은 경우 풀에 있는 머신의 명령어 집합 아키텍처의 값을 입력합니다. 유효한 값은 *amd64*,*ppc64le*,*s390x*, cover *64* 입니다.

작업자 풀 정보에는 작업자 풀의 풀 이름, 코어 수, 메모리 할당, 디스크 크기 할당 및 노드 수가 필요합니다. 작업자 풀 내의 작업자 노드는 단일 작업자 풀에 있거나 여러 작업자 풀에 분산될 수 있습니다.

사전 구성된 oVirt 환경에서 다음 네트워킹 세부 정보가 필요합니다.

- ovirt 네트워크 이름
- API VIP: 내부 API 통신에 사용할 IP 주소를 지정합니다.

  참고: 이 값은 사전 요구 사항 섹션에 나열된 DNS 레코드를 만드는 데 사용한 이름과 일치해야 합니다. 제공되지 않는 경우 api. 가 올바르게 확인되도록 DNS를 사전 구성해야 합니다.
- Ingress VIP: 인그레스 트래픽에 사용할 IP 주소를 지정합니다.

  참고: 이 값은 사전 요구 사항 섹션에 나열된 DNS 레코드를 만드는 데 사용한 이름과 일치해야 합니다. 제공되지 않는 경우 test.apps 가 올바르게 확인되도록 DNS를 사전 구성해야 합니다.
- 네트워크 유형: 기본값은 OpenShiftSDN 입니다. OVNKubernetes는 IPv6 사용에 필요한 설정입니다.
- 클러스터 네트워크 CIDR: Pod IP 주소에 사용할 수 있는 IP 주소 수 및 목록입니다. 이 블록은 다른 네트워크 블록을 겹치지 않아야 합니다. 기본값은 10.128.0.0/14입니다.
- 네트워크 호스트 접두사: 각 노드의 서브넷 접두사 길이를 설정합니다. 기본값은 23입니다.
- 서비스 네트워크 CIDR: 서비스의 IP 주소 블록을 제공합니다. 이 블록은 다른 네트워크 블록을 겹치지 않아야 합니다. 기본값은 172.30.0.0/16입니다.
- Machine CIDR: OpenShift Container Platform 호스트에서 사용하는 IP 주소 블록을 제공합니다. 이 블록은 다른 네트워크 블록을 겹치지 않아야 합니다. 기본값은 10.0.0.0/16입니다.

네트워크 추가를 클릭하여 추가 네트워크를 추가할 수 있습니다. **IPv6** 주소를 사용하는 경우 네트워크가 두 개 이상 있어야 합니다.

인증 정보에서 제공되는 프록시 정보는 프록시 필드에 자동으로 추가됩니다. 정보를 그대로 사용하거나 덮어쓰거나 프록시를 활성화하려면 정보를 추가할 수 있습니다. 다음 목록에는 프록시 생성에 필요한 정보가 포함되어 있습니다.

- **HTTP 프록시 URL: HTTP** 트래픽의 프록시로 사용해야 하는 **URL**을 지정합니다.
- **HTTPS 프록시 URL: HTTPS** 트래픽에 사용해야 하는 보안 프록시 **URL**을 지정합니다. 값을 제공하지 않으면 **HTTP** 및 **HTTPS** 모두에 대해 **HTTP** 프록시 **URL** 과 동일한 값이 사용됩니다.
- 프록시 도메인이 없음: 프록시를 바이패스해야 하는 쉼표로 구분된 도메인 목록을 제공합니다. 해당 도메인에 있는 모든 하위 도메인을 포함하려면 마침표로 도메인 이름을 시작합니다 . 모든 대상에 대한 프록시를 바이패스하려면 별표 * 를 추가합니다.
- 추가 신뢰 번들: 미러 레지스트리에 액세스하는 데 필요한 인증서 파일의 내용을 지정합니다.

클러스터를 생성하기 전에 정보를 검토하고 선택적으로 사용자 지정할 때 **YAML** 스위치를 클릭하여 패널에서 **install-config.yaml** 파일 콘텐츠를 볼 수 있습니다. 업데이트가 있는 경우 사용자 지정 설정으로 **YAML** 파일을 편집할 수 있습니다.

참고: 클러스터를 가져오기 위해 클러스터 세부 정보와 함께 제공된 **kubectl** 명령을 실행할 필요가 없습니다. 클러스터를 생성할 때 **Red Hat Advanced Cluster Management**의 관리 하에 자동으로 구성됩니다.

클러스터에 액세스하는 방법에 대한 지침을 보려면 클러스터에 계속 액세스합니다.

1.6.8. 베어 메탈에서 클러스터 생성

Kubernetes 콘솔을 위한 **Red Hat Advanced Cluster Management**를 사용하여 베어 메탈 환경에서 **Red Hat OpenShift Container Platform** 클러스터를 생성할 수 있습니다.

사용 중단 알림: 베어 메탈 자산을 사용하여 베어 메탈 클러스터를 생성하는 절차는 더 이상 사용되지 않습니다. 베어 메탈 자산은 향후 릴리스에서 제거될 예정입니다.

클러스터를 생성할 때 생성 프로세스에서 **Hive** 리소스와 함께 **OpenShift Container Platform** 설치 프로그램을 사용합니다. 이 절차를 완료한 후 클러스터 생성에 대한 질문이 있는 경우 자세한 내용은 **OpenShift Container Platform** 설명서의 베어 메탈에 설치를 참조하십시오.

- 사전 요구 사항
- 베어 메탈 클러스터 생성

1.6.8.1. 사전 요구 사항

베어 메탈 환경에서 클러스터를 생성하기 전에 다음 사전 요구 사항을 참조하십시오.

- **OpenShift Container Platform** 버전 **4.6** 이상에 배포된 **Red Hat Advanced Cluster Management for Kubernetes** 허브 클러스터가 있어야 합니다.
- **Red Hat Advanced Cluster Management for Kubernetes hub cluster (connected)** 클러스터의 인터넷 액세스 또는 클러스터 생성에 필요한 이미지를 검색하려면 인터넷에 연결되지 않은 내부 또는 미러 레지스트리에 대한 연결이 필요합니다.
- **Hive** 클러스터를 생성하는 데 사용되는 부트스트랩 가상 머신을 실행하는 임시 외부 **KVM** 호스트가 필요합니다. 자세한 내용은 프로비저닝 호스트 준비를 참조하십시오.
- 배포된 **Red Hat Advanced Cluster Management for Kubernetes hub** 클러스터는 **provisioning** 네트워크로 라우팅할 수 있어야 합니다.
- 이전 항목의 부트스트랩 가상 머신의 **libvirt URI, SSH** 개인 키 및 **SSH** 알려진 호스트 목록을 포함하는 베어 메탈 서버 로그인 자격 증명이 필요합니다. 자세한 내용은 **OpenShift** 설치 환경 설정을 참조하십시오.
- 구성된 베어 메탈 인증 정보가 필요합니다. 자세한 내용은 베어 메탈의 인증 정보 생성을 참조하십시오.

- 사용자 이름, 암호 및 베이스 보드 관리 컨트롤러 주소가 포함된 베어 메탈 환경에 대한 로그인 인증 정보가 있어야 합니다.

- 인증서 확인을 사용하는 경우 구성된 베어 메탈 자산이 필요합니다. 자세한 내용은 베어 메탈 자산 생성 및 수정을 참조하십시오.

- **OpenShift Container Platform** 이미지 풀 시크릿이 필요합니다. 자세한 내용은 이미지 풀 시크릿 사용을 참조하십시오.

  참고:

  - 베어 메탈 자산, 관리형 베어 메탈 클러스터 및 관련 시크릿은 동일한 네임스페이스에 있어야 합니다.

  - 클라우드 공급자 액세스 키를 변경하는 경우 프로비저닝된 클러스터 액세스 키를 수동으로 업데이트해야 합니다. 자세한 내용은 알려진 문제에서 프로비저닝된 클러스터에 대한 자동 시크릿 업데이트가 지원되지 않음을 참조하십시오.

  - 베어 메탈 공급자 및 연결이 끊긴 설치를 사용하여 클러스터를 생성하는 경우 연결이 *끊긴 설치의 구성* 섹션에 모든 설정을 저장해야 합니다. 클러스터 생성 콘솔 편집기에 입력할 수 없습니다.

1.6.8.2. 베어 메탈 클러스터 생성

Kubernetes 콘솔용 **Red Hat Advanced Cluster Management**에서 클러스터를 생성하려면 **Infrastructure > Clusters** 로 이동합니다. *클러스터* 페이지에서 클러스터 생성을 클릭하고 콘솔의 단계를 완료합니다.

참고: 이 절차는 클러스터를 생성하기 위한 것입니다. 가져올 기존 클러스터가 있는 경우 해당 단계를 위해 대상 관리 클러스터 가져오기를 **hub** 클러스터로 참조하십시오.

인증 정보를 생성해야 하는 경우 인증 정보 생성에 대한 자세한 내용은 베어 메탈 의 인증 정보 생성을 참조하십시오.

베어 메탈 클러스터의 경우 클러스터 이름은 임의의 이름일 수 없습니다. 클러스터 **URL**과 연결되어 있습니다. 사용하는 클러스터 이름이 **DNS** 및 네트워크 설정과 일치하는지 확인합니다.

중요: 클러스터를 생성할 때 **Red Hat Advanced Cluster Management** 컨트롤러에서 클러스터 및 해당 리소스의 네임스페이스를 생성합니다. 해당 네임스페이스에 해당 클러스터 인스턴스의 리소스만 포함해야 합니다. 클러스터를 삭제하면 네임스페이스와 그 안에 있는 모든 리소스가 삭제됩니다.

tips: 콘솔의 정보를 입력할 때 콘텐츠 업데이트를 보려면 **YAML: on** 을 선택합니다.

기존 클러스터 세트에 클러스터를 추가하려면 클러스터에 대한 올바른 권한이 있어야 합니다. 클러스터를 생성할 때 **cluster-admin** 권한이 없는 경우 **clusterset-admin** 권한이 있는 클러스터를 선택해야 합니다. 지정된 클러스터 세트에 대한 올바른 권한이 없으면 클러스터 생성에 실패합니다. 선택할 클러스터 설정 옵션이 없는 경우 **clusterset-admin** 권한을 클러스터에 제공하기 위해 클러스터 관리자에게 문의하십시오.

관리되는 모든 클러스터는 관리형 클러스터 세트와 연결되어야 합니다. 관리 클러스터를 **ManagedClusterSet** 에 할당하지 않으면 기본 관리 클러스터 세트에 자동으로 추가됩니다.

공급자의 기본 도메인은 **Red Hat OpenShift Container Platform** 클러스터 구성 요소에 대한 경로를 생성하는 데 사용됩니다. 클러스터 공급자의 **DNS**에서 **SOA(Start of Authority)** 레코드로 구성됩니다. 이 이름은 클러스터의 호스트 이름에 사용됩니다.

베어 메탈 공급자 계정에 대해 구성한 선택한 인증 정보와 연결된 기본 **DNS** 도메인이 이미 있는 경우 해당 필드에 채워집니다. 값을 작성하여 변경할 수 있지만 클러스터를 생성한 후에는 이름을 변경할 수 없습니다. 자세한 내용은 **OpenShift Container Platform** 설명서의 베어 메탈에 설치를 참조하십시오.

릴리스 이미지는 클러스터를 생성하는 데 사용되는 **OpenShift Container Platform** 이미지의 버전을 식별합니다. 사용하려는 이미지가 표준 이미지가 아닌 경우 사용하려는 이미지에 **URL**을 입력할 수 있습니다. 릴리스 이미지에 대한 자세한 내용은 릴리스 이미지를 참조하십시오.

호스트 목록은 기존 베어 메탈 자산에서 컴파일되며 인증 정보와 연결됩니다. 베어 메탈 호스트에서 최신 펌웨어를 실행 중이거나 프로비저닝에 실패할 수 있는지 확인합니다. 하이퍼바이저와 동일한 브릿지 네트워크에 있는 최소 **3**개의 베어 메탈 자산을 선택해야 합니다. 생성된 베어 메탈 자산이 없는 경우 생성 프로세스를 계속하기 전에 자산 가져오기 를 선택하여 만들거나 가져올 수 있습니다. 베어 메탈 자산 생성에 대한 자세한 내용은 베어 메탈 자산 생성 및 수정을 참조하십시오. 또는 **Disable certificate verification** 을 선택하여 요구 사항을 바이패스할 수 있습니다.

다음 표에서는 네트워킹 옵션 및 해당 설명을 보여줍니다.

매개변수	설명	필수 또는 선택 사항
provisioning 네트워크 CIDR	프로비저닝에 사용할 네트워크의 CIDR입니다. 예제 형식은 172.30.0.0/16입니다.	필수 항목
프로비저닝 네트워크 인터페이스	provisioning 네트워크에 연결된 컨트롤 플레인 노드의 네트워크 인터페이스 이름입니다.	필수 항목
프로비저닝 네트워크 브리지	provisioning 네트워크에 연결된 하이퍼바이저의 브리지 이름입니다.	필수 항목
외부 네트워크 브리지	외부 네트워크에 연결된 하이퍼바이저의 브리지 이름입니다.	필수 항목
API VIP	내부 API 통신에 사용할 가상 IP입니다. **api.<cluster_name>.<Base DNS domain> 경로가 올바르게 확인되도록 DNS** 는 A/AAAA 또는 CNAME 레코드로 사전 구성해야 합니다.	필수 항목
Ingress VIP	Ingress 트래픽에 사용할 가상 IP입니다. ***.apps.<cluster_name>.<Base DNS domain> 경로가 올바르게 확인되도록 DNS** 는 A/AAAA 또는 CNAME 레코드로 사전 구성해야 합니다.	선택 사항
네트워크 유형	배포할 Pod 네트워크 공급자 플러그인입니다. OpenShift Container Platform 4.3에서는 OpenShiftSDN 플러그인만 지원됩니다. OVNKubernetes 플러그인은 OpenShift Container Platform 버전 4.3, 4.4 및 4.5에서 기술 프리뷰로 사용할 수 있습니다. 일반적으로 OpenShift Container Platform 버전 4.6 이상에서 사용할 수 있습니다. OVNKubernetes는 IPv6와 함께 사용해야 합니다. 기본값은 **OpenShiftSDN**입니다.	필수 항목
클러스터 네트워크 CIDR	Pod IP 주소가 할당되는 IP 주소 블록입니다. OpenShiftSDN 네트워크 플러그인은 여러 클러스터 네트워크를 지원합니다. 여러 클러스터 네트워크의 주소 블록이 겹치지 않아야 합니다. 예상 워크로드에 맞게 충분히 큰 주소 풀을 선택합니다. 기본값은 10.128.0.0/14입니다.	필수 항목

매개변수	설명	필수 또는 선택 사항
네트워크 호스트 접두사	개별 노드 각각에 할당할 서브넷 접두사 길이입니다. 예를 들어 hostPrefix를 23으로 설정하면 지정된 CIDR 이외 /23 서브넷이 각 노드에 할당되어 510(2^(32-23)-2) Pod IP 주소가 허용됩니다. 기본값은 23입니다.	필수 항목
서비스 네트워크 CIDR	서비스의 IP 주소 블록입니다. OpenShiftSDN은 하나의 serviceNetwork 블록만 허용합니다. 이 주소는 다른 네트워크 블록을 겹치지 않아야 합니다. 기본값은 172.30.0.0/16입니다.	필수 항목
Machine CIDR	OpenShift Container Platform 호스트에서 사용하는 IP 주소 블록입니다. 주소 블록은 다른 네트워크 블록을 겹치지 않아야 합니다. 기본값은 10.0.0.0/16입니다.	필수 항목

IPv6 주소를 사용하는 경우 네트워크가 두 개 이상 있어야 합니다.

인증 정보에서 제공되는 프록시 정보는 프록시 필드에 자동으로 추가됩니다. 정보를 그대로 사용하거나 덮어쓰거나 프록시를 활성화하려면 정보를 추가할 수 있습니다. 다음 목록에는 프록시 생성에 필요한 정보가 포함되어 있습니다.

- **HTTP 프록시 URL: HTTP 트래픽의 프록시로 사용해야 하는 URL입니다.**
- **HTTPS 프록시 URL: HTTPS 트래픽에 사용해야 하는 보안 프록시 URL입니다. 값을 제공하지 않으면 HTTP 및 HTTPS 모두에 대해 HTTP 프록시 URL 과 동일한 값이 사용됩니다.**
- **프록시 도메인이 없음: 프록시를 바이패스해야 하는 쉼표로 구분된 도메인 목록입니다. 해당 도메인에 있는 모든 하위 도메인을 포함하려면 마침표로 도메인 이름을 시작합니다 . 모든 대상에 대한 프록시를 바이패스하려면 별표 * 를 추가합니다.**
- **추가 신뢰 번들: 미러 레지스트리에 액세스하는 데 필요한 인증서 파일의 콘텐츠입니다.**

클러스터를 생성하기 전에 정보를 검토하고 선택적으로 사용자 지정할 때 YAML: On 을 선택하여 패

널에서 **install-config.yaml** 파일 콘텐츠를 볼 수 있습니다. 업데이트가 있는 경우 사용자 지정 설정으로 **YAML** 파일을 편집할 수 있습니다.

참고: 클러스터를 가져오기 위해 클러스터 세부 정보와 함께 제공된 **kubectl** 명령을 실행할 필요가 없습니다. 클러스터를 생성하면 **Red Hat Advanced Cluster Management** 관리로 자동 구성됩니다.

클러스터에 액세스하는 방법에 대한 지침을 보려면 클러스터에 계속 액세스합니다.

1.6.9. 온-프레미스 환경에서 클러스터 생성

Red Hat Advanced Cluster Management for Kubernetes 콘솔을 사용하여 온프레미스 **Red Hat OpenShift Container Platform** 클러스터를 생성할 수 있습니다. 이 프로세스는 더 이상 사용되지 않는 베어 메탈 프로세스 대신 권장됩니다.

모범 사례: 단일 노드 **OpenShift(SNO)** 클러스터를 생성하려면 다음 절차를 사용하십시오. **VMware vSphere, Red Hat OpenStack, Red Hat Virtualization Platform** 및 베어 메탈 환경에서 단일 노드 **OpenShift** 클러스터를 생성할 수 있습니다. 플랫폼 값이 **platform=none** 으로 설정되어 있으므로 클러스터를 설치하는 플랫폼과의 통합은 없습니다. 단일 노드 **OpenShift** 클러스터에는 컨트롤 플레인 서비스와 사용자 워크로드를 호스팅하는 단일 노드만 포함되어 있습니다. 이 구성은 클러스터의 리소스 풋프린트를 최소화하려는 경우 유용할 수 있습니다.

또한 **Red Hat OpenShift Container Platform**에서 사용할 수 있는 기술 프리뷰 기능인 제로 태그 프로비저닝 기능을 사용하여 엣지 리소스에 여러 단일 노드 **OpenShift** 클러스터를 프로비저닝하는 절차를 테스트할 수도 있습니다. 해당 절차에 대한 자세한 내용은 **OpenShift Container Platform** 설명서의 연결이 끊긴 환경에서 대규모로 분산 장치 배포를 참조하십시오.

- 사전 요구 사항
- 콘솔을 사용하여 클러스터 생성

1.6.9.1. 사전 요구 사항

온-프레미스 환경에서 클러스터를 만들기 전에 다음 사전 요구 사항을 참조하십시오.

- **OpenShift Container Platform** 버전 **4.9** 이상에 배포된 **Red Hat Advanced Cluster Management hub** 클러스터가 있어야 합니다.

- 구성된 호스트가 있는 구성된 인프라 환경이 필요합니다. 자세한 내용은 인프라 환경 생성을 참조하십시오.

- **Red Hat Advanced Cluster Management for Kubernetes hub** 클러스터(연결됨) 클러스터의 인터넷 액세스 또는 클러스터 생성에 필요한 이미지를 검색하는 데 인터넷(연결이 끊어짐) 이 인터넷에 연결되어 있어야 합니다.

- 구성된 온-프레미스 인증 정보가 필요합니다. 자세한 내용은 온-프레미스 환경에 대한 자격 증명 만들기를 참조하십시오.**See Creating a credential for an on-premises environment for more information.**

- **OpenShift Container Platform** 이미지 풀 시크릿이 필요합니다. 자세한 내용은 이미지 풀 시크릿 사용을 참조하십시오.

1.6.9.2. 콘솔을 사용하여 클러스터 생성

Kubernetes 콘솔용 **Red Hat Advanced Cluster Management**에서 클러스터를 생성하려면 **Infrastructure > Clusters** 로 이동합니다. *클러스터* 페이지에서 클러스터 생성을 클릭하고 콘솔의 단계를 완료합니다.

지원되는 설치에 다음 옵션을 사용할 수 있습니다.

- 기존 검색된 호스트 사용: 기존 인프라 환경에 있는 호스트 목록에서 호스트를 선택합니다.

- 새 호스트 검색: 기존 인프라 환경에 아직 없는 호스트를 검색합니다. 인프라 환경에 이미 있는 호스트를 사용하지 않고 자체 호스트를 검색합니다.

자세한 내용은 온-프레미스 환경에 대한 자격 증명 만들기를 참조하십시오.**If you need to create a credential, see Creating a credential for an on-premises environment for more information.**

클러스터의 이름은 클러스터의 호스트 이름에 사용됩니다.

중요: 클러스터를 생성할 때 **Red Hat Advanced Cluster Management** 컨트롤러에서 클러스터 및 해당 리소스의 네임스페이스를 생성합니다. 해당 네임스페이스에 해당 클러스터 인스턴스의 리소스만 포함

해야 합니다. 클러스터를 삭제하면 네임스페이스와 그 안에 있는 모든 리소스가 삭제됩니다.

tips: 콘솔의 정보를 입력할 때 콘텐츠 업데이트를 보려면 **YAML: on** 을 선택합니다.

기존 클러스터 세트에 클러스터를 추가하려면 클러스터에 대한 올바른 권한이 있어야 합니다. 클러스터를 생성할 때 **cluster-admin** 권한이 없는 경우 **clusterset-admin** 권한이 있는 클러스터를 선택해야 합니다. 지정된 클러스터 세트에 대한 올바른 권한이 없으면 클러스터 생성에 실패합니다. 선택할 클러스터 설정 옵션이 없는 경우 **clusterset-admin** 권한을 클러스터에 제공하기 위해 클러스터 관리자에게 문의하십시오.

관리되는 모든 클러스터는 관리형 클러스터 세트와 연결되어야 합니다. 관리 클러스터를 **ManagedClusterSet** 에 할당하지 않으면 기본 관리 클러스터 세트에 자동으로 추가됩니다.

공급자 계정에 대해 구성한 선택한 인증 정보와 연결된 기본 **DNS** 도메인이 이미 있는 경우 해당 필드에 채워집니다. 값을 수정하여 변경할 수 있지만 이 설정은 클러스터를 생성한 후에는 변경할 수 없습니다. 공급자의 기본 도메인은 **Red Hat OpenShift Container Platform** 클러스터 구성 요소에 대한 경로를 생성하는 데 사용됩니다. 클러스터 공급자의 **DNS**에서 **SOA(Start of Authority)** 레코드로 구성됩니다.

OpenShift 버전 은 클러스터를 생성하는 데 사용되는 **OpenShift Container Platform** 이미지의 버전을 식별합니다. 사용하려는 버전이 사용 가능한 경우 이미지 목록에서 이미지를 선택할 수 있습니다. 사용하려는 이미지가 표준 이미지가 아닌 경우 사용하려는 이미지에 **URL**을 입력할 수 있습니다. 릴리스 이미지에 대한 자세한 내용은 릴리스 이미지를 참조하십시오.

4.9 이상 **OpenShift** 버전을 선택하면 **Install single node OpenShift (SNO)** 를 선택하는 옵션이 표시됩니다. 단일 노드 **OpenShift** 클러스터에는 컨트롤 플레인 서비스와 사용자 워크로드를 호스팅하는 단일 노드가 포함되어 있습니다. 생성된 후에는 단일 노드 **OpenShift** 클러스터에 노드를 추가할 수 없습니다.

클러스터가 단일 노드 **OpenShift** 클러스터가 되도록 하려면 단일 노드 **OpenShift** 옵션을 선택합니다.

참고: 단일 노드 **OpenShift** 컨트롤 플레인에는 **8**개의 **CPU** 코어가 필요하지만 다중 노드 컨트롤 플레인 클러스터의 컨트롤 플레인 노드에는 **4**개의 **CPU** 코어만 필요합니다.

클러스터를 검토하고 저장하면 클러스터가 초안 클러스터로 저장됩니다. 클러스터 페이지에서 클러스터 이름을 선택하여 생성 프로세스를 종료하고 나중에 프로세스를 완료할 수 *있습니다*.

기존 호스트를 사용하는 경우 호스트를 직접 선택할지 또는 호스트를 자동으로 선택할지 선택합니다.

호스트 수는 선택한 노드 수를 기반으로 합니다. 예를 들어 SNO 클러스터에는 하나의 호스트만 필요하지만 표준 3-노드 클러스터에는 3개의 호스트가 필요합니다.

이 클러스터의 요구 사항을 충족하는 사용 가능한 호스트의 *위치는 호스트 위치* 목록에 표시됩니다. 호스트 배포 및 고가용성 구성의 경우 여러 위치를 선택합니다.

기존 인프라 환경 없이 새 호스트를 검색하는 경우 4단계로 시작하여 호스트를 정의하는 인프라 환경에 호스트 추가 단계를 완료합니다.

호스트가 바인딩되고 검증이 통과되면 다음 IP 주소를 추가하여 클러스터의 네트워킹 정보를 완료합니다.

- API VIP: 내부 API 통신에 사용할 IP 주소를 지정합니다.

  참고: 이 값은 사전 요구 사항 섹션에 나열된 DNS 레코드를 만드는 데 사용한 이름과 일치해야 합니다. 제공되지 않는 경우 api. 가 올바르게 확인되도록 DNS를 사전 구성해야 합니다.

- Ingress VIP: Ingress 트래픽에 사용할 IP 주소를 지정합니다.

  참고: 이 값은 사전 요구 사항 섹션에 나열된 DNS 레코드를 만드는 데 사용한 이름과 일치해야 합니다. 제공되지 않는 경우 test.apps 가 올바르게 확인되도록 DNS를 사전 구성해야 합니다.

클러스터 탐색 페이지에서 설치 상태를 볼 수 있습니다.

클러스터에 액세스하는 방법에 대한 지침을 보려면 클러스터에 계속 액세스합니다.

1.6.10. 생성된 클러스터 분리(기술 프리뷰)

Red Hat Advanced Cluster Management for Kubernetes를 사용하여 생성된 클러스터를 사용하여 리소스를 보존할 수 있습니다. 하이베이팅 클러스터에는 실행 중인 리소스보다 훨씬 적은 리소스가 필요하므로 하이베이팅 상태의 클러스터를 이동하거나 부족하여 공급자 비용을 절감할 수 있습니다. 이 기능은 다음 환경에서 Red Hat Advanced Cluster Management에서 생성한 클러스터에만 적용됩니다.

- Amazon Web Services

- **Microsoft Azure**

- **Google Cloud Platform**

1.6.10.1. 콘솔을 사용하여 클러스터의 iPXE

Red Hat Advanced Cluster Management 콘솔을 사용하여 Red Hat Advanced Cluster Management에서 생성한 클러스터를 사용하려면 다음 단계를 완료하십시오.

1. **Red Hat Advanced Cluster Management 탐색 메뉴에서 Infrastructure > Clusters 를 선택합니다. *Manage clusters* 탭이 선택되어 있는지 확인합니다.**

2. **클러스터 의 *옵션* 메뉴에서 iPXE 클러스터를 선택합니다. 참고: *iPXE 클러스터* 옵션을 사용할 수 없는 경우 클러스터를 사용할 수 없습니다. 이는 Red Hat Advanced Cluster Management에서 클러스터를 가져오지 않고 가져올 때 발생할 수 있습니다.**

클러스터 페이지의 클러스터 상태는 프로세스가 완료되면 Hibernating 입니다.

팁: 클러스터 페이지에서 클러스터를 선택하고 Actions > iPXE 클러스터를 선택하여 여러 클러스터를 할 수 있습니다.

선택한 클러스터가 hibernating입니다.

1.6.10.2. CLI를 사용하여 클러스터 way

CLI를 사용하여 Red Hat Advanced Cluster Management에서 생성한 클러스터를 사용하려면 다음 단계를 완료하십시오.

1. ***다음 명령을 입력하여 ECDHE할 클러스터의 설정을 편집합니다.***

```
oc edit clusterdeployment <name-of-cluster> -n <namespace-of-cluster>
```

name-of-cluster 를 ECDHE 클러스터의 이름으로 바꿉니다.

namespace-of-cluster 를 *ECDHE* 클러스터의 네임스페이스로 바꿉니다.

2. *spec.powerState* 의 값을 *Hibernating* 으로 변경합니다.

3. 다음 명령을 입력하여 클러스터 상태를 확인합니다.

```
oc get clusterdeployment <name-of-cluster> -n <namespace-of-cluster> -o yaml
```

name-of-cluster 를 *ECDHE* 클러스터의 이름으로 바꿉니다.

namespace-of-cluster 를 *ECDHE* 클러스터의 네임스페이스로 바꿉니다.

클러스터를 분리하는 프로세스가 완료되면 클러스터 유형 값이 *type=Hibernating* 입니다.

선택한 클러스터가 *hibernating*입니다.

1.6.10.3. 콘솔을 사용하여 임시 클러스터의 정상적인 작업 재시작

Red Hat Advanced Cluster Management 콘솔을 사용하여 상위 클러스터의 정상적인 작동을 다시 시작하려면 다음 단계를 완료하십시오.

1. *Red Hat Advanced Cluster Management* 탐색 메뉴에서 *Infrastructure > Clusters* 를 선택합니다. *Manage clusters* 탭이 선택되어 있는지 확인합니다.

2. 재개하려는 클러스터 의 옵션 메뉴에서 클러스터 다시 시작을 선택합니다.

프로세스가 완료되면 클러스터 페이지의 클러스터 상태가 *Ready* 입니다.

팁: 클러스터 페이지에서 재개할 클러스터를 선택하고 작업 *>* 클러스터 재시작을 선택하여 여러 클러스터를 다시 시작할 수 있습니다.

선택한 클러스터가 정상적인 작업을 다시 시작합니다.

1.6.10.4. CLI를 사용하여 상위 클러스터의 정상적인 작업 재시작

CLI를 사용하여 상위 클러스터의 정상적인 작동을 다시 시작하려면 다음 단계를 완료합니다.

1. 다음 명령을 입력하여 클러스터의 설정을 편집합니다.

```
oc edit clusterdeployment <name-of-cluster> -n <namespace-of-cluster>
```

   name-of-cluster 를 **ECDHE** 클러스터의 이름으로 바꿉니다.

   namespace-of-cluster 를 **ECDHE** 클러스터의 네임스페이스로 바꿉니다.

2. **spec.powerState** 의 값을 **Running** 으로 변경합니다.

3. 다음 명령을 입력하여 클러스터 상태를 확인합니다.

```
oc get clusterdeployment <name-of-cluster> -n <namespace-of-cluster> -o yaml
```

   name-of-cluster 를 **ECDHE** 클러스터의 이름으로 바꿉니다.

   namespace-of-cluster 를 **ECDHE** 클러스터의 네임스페이스로 바꿉니다.

   클러스터를 재시작하는 프로세스가 완료되면 클러스터 유형 값이 **type=Running** 입니다.

선택한 클러스터가 정상적인 작업을 다시 시작합니다.

1.7. 대상 관리 클러스터를 허브 클러스터로 가져오기

다른 **Kubernetes** 클라우드 공급자에서 클러스터를 가져올 수 있습니다. 가져온 후 대상 클러스터는 **Kubernetes** 허브 클러스터용 **Red Hat Advanced Cluster Management** 클러스터의 관리 클러스터가

됩니다. 별도로 지정하지 않는 한 허브 클러스터 및 대상 관리 클러스터에 액세스할 수 있는 어디에서나 가져오기 작업을 완료합니다.

허브 클러스터는 다른 허브 클러스터를 관리할 수 없지만 자체적으로 관리할 수 있습니다. **hub** 클러스터는 자동으로 가져오고 자체 관리되도록 구성됩니다. **hub** 클러스터를 수동으로 가져올 필요가 없습니다.

그러나 **hub** 클러스터를 제거하고 다시 가져오려는 경우 **local-cluster:true** 레이블을 추가해야 합니다.

콘솔 또는 **CLI**에서 관리되는 클러스터를 설정하려면 다음 지침에서 선택합니다.

필수 사용자 유형 또는 액세스 수준: 클러스터 관리자

- 콘솔을 사용하여 기존 클러스터 가져오기
- **CLI**를 사용하여 관리형 클러스터 가져오기
- 클러스터의 **klusterlet** 추가 기능 설정 수정

1.7.1. 콘솔을 사용하여 기존 클러스터 가져오기

Red Hat Advanced Cluster Management for Kubernetes를 설치한 후 관리할 클러스터를 가져올 준비가 된 것입니다. 콘솔과 **CLI**에서 모두 가져올 수 있습니다.

콘솔에서 가져오려면 다음 절차를 따르십시오. 이 절차 중에 인증을 위해 터미널이 필요합니다.

- 사전 요구 사항
- 클러스터 가져오기
- 클러스터 제거

1.7.1.1. 사전 요구 사항

- 배포된 **Kubernetes** 허브 클러스터용 **Red Hat Advanced Cluster Management** 클러스터가 필요합니다. 베어 메탈 클러스터를 가져오는 경우 **Red Hat OpenShift Container Platform** 버전 **4.8** 이상에 **hub** 클러스터가 설치되어 있어야 합니다.

- 관리하려는 클러스터와 인터넷 연결이 필요합니다.

- **kubectl** 을 설치합니다. **kubectl** 을 설치하려면 **Kubernetes** 문서에서 **kubectl** 설치 및 설정을 참조하십시오.

- **base64** 명령줄 도구가 필요합니다.

- 참고: **OpenShift Container Platform**에서 생성되지 않은 클러스터를 가져오는 경우 **multiclusterhub.spec.imagePullSecret** 이 정의되어 있어야 합니다. 이 시크릿은 **Red Hat Advanced Cluster Management**를 설치하면 생성될 수 있습니다.

  새 항목을 생성해야 하는 경우 다음 단계를 완료합니다.

  1. **cloud.redhat.com** 에서 **Kubernetes** 풀 시크릿을 다운로드합니다.
  2. **hub** 클러스터의 네임스페이스에 풀 시크릿을 추가합니다.
  3. 다음 명령을 실행하여 **hub** 클러스터의 네임스페이스에 새 보안을 생성합니다.

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

     open-cluster-management 를 **hub** 클러스터의 네임스페이스 이름으로 교체합니다. **hub** 클러스터의 기본 네임스페이스는 **open-cluster-management** 입니다.

     다운로드한 풀 시크릿 경로로 **path-to-pull-secret** 을 교체합니다.

시크릿을 가져오면 관리 클러스터에 자동으로 복사됩니다.

가져오기 보안에 대한 자세한 내용은 이미지 풀 시크릿 사용 또는 서비스 계정 이해 및 생성을 참조하십시오.

이 보안을 정의하는 방법에 대한 자세한 내용은 사용자 정의 이미지 가져오기 보안을 참조하십시오.

- 가져올 클러스터에서 에이전트가 삭제되었는지 확인합니다. 오류를 방지하려면 **open-cluster-management-agent** 및 **open-cluster-management-agent-addon** 네임스페이스를 제거해야 합니다.

- **Red Hat OpenShift Dedicated** 환경에서 가져오려면 다음 정보를 참조하십시오.

  - **Red Hat OpenShift Dedicated** 환경에 허브 클러스터가 배포되어 있어야 합니다.

  - **Red Hat OpenShift Dedicated**의 기본 권한은 **dedicated-admin**이지만 네임스페이스를 생성할 수 있는 권한은 모두 포함되어 있지 않습니다. **Kubernetes**용 **Red Hat Advanced Cluster Management**로 클러스터를 가져오고 관리하려면 **cluster-admin** 권한이 있어야 합니다.

필수 사용자 유형 또는 액세스 수준: 클러스터 관리자

1.7.1.2. 클러스터 가져오기

사용 가능한 각 클라우드 공급자에 대해 **Kubernetes** 콘솔용 **Red Hat Advanced Cluster Management**에서 기존 클러스터를 가져올 수 있습니다.

참고: 허브 클러스터는 다른 허브 클러스터를 관리할 수 없습니다. **hub** 클러스터는 자동으로 가져오고 관리하도록 설정되어 있으므로 자체적으로 관리하기 위해 **hub** 클러스터를 수동으로 가져올 필요가 없습니다.

1. 탐색 메뉴에서 인프라 **>** 클러스터를 선택합니다.

2. **Managed clusters** 탭에서 클러스터 가져오기 를 클릭합니다.

3. 클러스터 이름을 제공합니다. 기본적으로 네임스페이스는 클러스터 이름과 네임스페이스에 사용됩니다.

중요: 클러스터를 생성할 때 **Red Hat Advanced Cluster Management** 컨트롤러에서 클러스터 및 해당 리소스의 네임스페이스를 생성합니다. 해당 네임스페이스에 해당 클러스터 인스턴스의 리소스만 포함해야 합니다. 클러스터를 삭제하면 네임스페이스와 그 안에 있는 모든 리소스가 삭제됩니다.

1. **cluster-admin** 권한이 있는 기존 클러스터에 추가하려면 **Cluster** 세트를 지정합니다. 클러스터를 생성할 때 **cluster-admin** 권한이 없는 경우 **clusterset-admin** 권한이 있는 클러스터를 선택해야 합니다. 지정된 클러스터 세트에 대한 올바른 권한이 없으면 클러스터 생성에 실패합니다. 클러스터 관리자에게 문의하여 선택할 클러스터 설정 옵션이 없는 경우 **clusterset-admin** 권한을 클러스터에 제공합니다.

   관리되는 모든 클러스터는 관리형 클러스터 세트와 연결되어야 합니다. 관리 클러스터를 **ManagedClusterSet** 에 할당하지 않으면 기본 관리 클러스터 세트에 자동으로 추가됩니다.

2. 선택 사항: 추가 라벨을 추가합니다.

   참고: **Red Hat OpenShift Dedicated** 클러스터를 가져오고 **vendor=OpenShiftDedicated** 에 대한 레이블을 추가하거나 **vendor=auto-detect** 의 레이블을 추가하면 **managed-by=platform** 레이블이 클러스터에 자동으로 추가됩니다. 추가된 이 레이블을 사용하여 클러스터를 **Red Hat OpenShift Dedicated** 클러스터로 식별하고 그룹으로 **Red Hat OpenShift Dedicated** 클러스터를 검색할 수 있습니다.

3. 다음 옵션에서 가져올 클러스터를 식별하는 데 사용할 가져오기 모드를 선택합니다.

   - 가져오기 명령 실행 수동으로 실행 : 제공한 정보를 기반으로 복사 및 실행할 수 있는 가져오기 명령을 생성합니다. **Save import**를 클릭하고 코드를 생성 하여 **open-cluster-management-agent-addon** 을 배포하는 데 사용하는 명령을 생성합니다. 확인 메시지가 표시됩니다.

     a. 기존 클러스터 가져오기 창에서 복사 명령을 선택하여 생성된 명령과 토큰을 클립보드에 복사합니다.

     중요: 이 명령에는 가져온 각 클러스터에 복사되는 풀 시크릿 정보가 포함되어 있습니다. 가져온 클러스터에 액세스할 수 있는 모든 사용자는 풀 시크릿 정보도 볼 수 있습

니다. https://cloud.redhat.com/ 에서 보조 풀 시크릿을 생성하거나 서비스 계정을 생성하여 개인 인증 정보를 보호하는 것이 좋습니다.

b. 가져오려는 관리형 클러스터에 로그인합니다.

c. **Red Hat OpenShift Dedicated** 환경의 경우에만 해당: 다음 단계를 완료합니다.

i. 관리 클러스터에서 **open-cluster-management-agent** 및 **open-cluster-management** 네임스페이스 또는 프로젝트를 생성합니다.

ii. **OpenShift Container Platform** 카탈로그에서 **klusterlet Operator**를 찾습니다.

iii. 생성한 **open-cluster-management** 네임스페이스 또는 프로젝트에 설치합니다.

중요: **open-cluster-management-agent** 네임스페이스에 **Operator**를 설치하지 마십시오.

iv. 다음 단계를 완료하여 가져오기 명령에서 부트스트랩 보안을 추출합니다.

A. 가져오기 명령을 생성합니다.

I. **Red Hat Advanced Cluster Management** 콘솔의 기본 탐색에서 **인프라 > 클러스터**를 선택합니다.

II. **Add a cluster > Import an existing cluster** 를 선택합니다.

III. 클러스터 정보를 추가하고 **Save import and generate code** 를 선택합니다.

B. 가져오기 명령을 복사합니다.

C. *import-command 라는 파일에 가져오기 명령을 붙여넣습니다.*

D. *다음 명령을 실행하여 새 파일에 콘텐츠를 삽입합니다.*

```
cat import-command | awk '{split($0,a,"&&"); print a[3]}' | awk '{split($0,a,"|"); print a[1]}' | sed -e "s/^ echo //" | base64 -d
```

E. *출력에서 bootstrap-hub-kubeconfig 라는 이름으로 시크릿을 찾아서 복사합니다.*

F. *관리 클러스터의 open-cluster-management-agent 네임스페이스에 보안을 적용합니다.*

G. *설치된 Operator에서 예제를 사용하여 klusterlet 리소스를 생성합니다. clusterName은 가져오기 중에 설정된 클러스터 이름과 동일한 이름을 변경해야 합니다.*

참고: managedcluster 리소스가 허브에 성공적으로 등록되면 두 개의 klusterlet Operator가 설치됩니다. 하나의 klusterlet Operator는 open-cluster-management 네임스페이스에 있으며 다른 하나는 open-cluster-management-agent 네임스페이스에 있습니다. 여러 Operator가 klusterlet의 기능에는 영향을 미치지 않습니다.

d. *Red Hat OpenShift Dedicated 환경에 없는 클러스터 가져오기의 경우 다음 단계를 완료합니다.*

i. *필요한 경우 관리 클러스터에 대해 kubectl 명령을 구성합니다.*

kubectl 명령행 인터페이스를 구성하는 방법을 알아보려면 지원되는 공급자를 참조하십시오.

ii. *open-cluster-management-agent-addon 을 관리형 클러스터에 배포하려면 복사한 명령과 토큰을 실행합니다.*

e. *클러스터 보기 를 선택하여 개요 페이지에서 클러스터의 요약을 확인합니다.*

- 기존 클러스터의 서버 **URL** 및 **API** 토큰을 입력합니다. 가져올 클러스터 의 서버 **URL** 및 **API** 토큰을 제공합니다.

- **kubeconfig**: 가져올 클러스터의 **kubeconfig** 파일의 콘텐츠를 복사하여 붙여넣습니다.

4. 선택 사항: **oc get managedcluster** 명령을 실행할 때 테이블에 표시되는 **URL**을 구성하여 클러스터 세부 정보 페이지에 있는 클러스터 **API** 주소를 구성합니다.

   a. **cluster-admin** 권한이 있는 **ID**로 **hub** 클러스터에 로그인합니다.

   b. 대상 관리 클러스터에 대해 **kubectl** 을 구성합니다.

      kubectl 을 구성하는 방법을 알아보려면 지원되는 공급자를 참조하십시오.

   c. 다음 명령을 입력하여 가져올 클러스터의 관리형 클러스터 항목을 편집합니다.

      ```
      oc edit managedcluster <cluster-name>
      ```

      cluster-name 을 관리형 클러스터의 이름으로 변경합니다.

   d. 다음 예와 같이 **YAML** 파일의 **ManagedCluster ClientConfigs** 섹션을 **ManagedClusterClientConfigs** 섹션에 추가합니다.

      ```
      spec:
        hubAcceptsClient: true
        managedClusterClientConfigs:
        - url: https://multicloud-console.apps.new-managed.dev.redhat.com
      ```

      가져올 관리형 클러스터에 대한 외부 액세스를 제공하는 **URL**로 **URL** 값을 바꿉니다.

클러스터를 가져옵니다. 다른 가져오기를 선택하여 다른 가져오기 를 가져올 수 있습니다.

1.7.1.3. 가져온 클러스터 제거

가져온 클러스터와 관리 클러스터에서 생성된 **open-cluster-management-agent-addon** 을 제거하려면 다음 절차를 완료합니다.

클러스터 페이지에서 **Actions > Detach cluster** 를 클릭하여 클러스터를 관리에서 제거합니다.

참고: **local-cluster** 라는 **hub** 클러스터를 분리하려고 하면 **disableHubSelfManagement** 의 기본 설정은 **false** 입니다. 이 설정을 사용하면 허브 클러스터가 분리될 때 자체적으로 다시 가져오고 자체적으로 관리되고 **MultiClusterHub** 컨트롤러를 조정합니다. 허브 클러스터가 분리 프로세스를 완료하고 다시 가져오는 데 시간이 걸릴 수 있습니다. 프로세스가 완료될 때까지 기다리지 않고 **hub** 클러스터를 다시 가져오려면 다음 명령을 입력하여 **multiclusterhub-operator Pod**를 재시작하고 더 빨리 다시 가져올 수 있습니다.

```
oc delete po -n open-cluster-management `oc get pod -n open-cluster-management | grep multiclusterhub-operator| cut -d' ' -f1`
```

disableHubSelfManagement 값을 **true** 로 변경하여 **hub** 클러스터의 값을 자동으로 가져오지 않도록 변경할 수 있습니다. 자세한 내용은 **disableHubSelfManagement** 주제를 참조하십시오.

1.7.2. CLI를 사용하여 관리형 클러스터 가져오기

Red Hat Advanced Cluster Management for Kubernetes를 설치한 후 **Red Hat OpenShift Container Platform CLI**를 사용하여 관리할 클러스터를 가져올 준비가 된 것입니다. 가져오는 클러스터의 **kubeconfig** 파일을 사용하여 클러스터를 가져오거나 가져올 클러스터에서 가져오기 명령을 수동으로 실행할 수 있습니다. 두 절차를 모두 문서화합니다.

- 사전 요구 사항
- 지원되는 아키텍처
- 가져오기 준비
- 자동 가져오기 보안을 사용하여 클러스터 가져오기
- 수동 명령으로 클러스터 가져오기

- klusterlet 애드온 가져오기

중요: 허브 클러스터는 다른 허브 클러스터를 관리할 수 없습니다. hub 클러스터는 자체적으로 자동으로 가져오고 관리하도록 설정되어 있습니다. 자체적으로 관리하기 위해 허브 클러스터를 수동으로 가져올 필요는 없습니다.

그러나 hub 클러스터를 제거하고 다시 가져오려는 경우 local-cluster:true 레이블을 추가해야 합니다.

1.7.2.1. 사전 요구 사항

- 배포된 Kubernetes 허브 클러스터용 Red Hat Advanced Cluster Management 클러스터가 필요합니다. 베어 메탈 클러스터를 가져오는 경우 Red Hat OpenShift Container Platform 버전 4.6 이상에 hub 클러스터가 설치되어 있어야 합니다.

- 인터넷에 연결되어 있는 별도의 클러스터가 있어야 합니다.

- oc 명령을 실행하려면 Red Hat OpenShift Container Platform CLI 버전 4.6 이상이 필요합니다. Red Hat OpenShift Container Platform CLI의 설치 및 구성에 대한 정보는 OpenShift CLI 시작하기 를 참조하십시오.

- Kubernetes CLI, kubectl 을 설치해야 합니다. kubectl 을 설치하려면 Kubernetes 문서에서 kubectl 설치 및 설정을 참조하십시오.

  참고: 콘솔에서 CLI 툴용 설치 파일을 다운로드합니다.

- OpenShift Container Platform에서 생성되지 않은 클러스터를 가져오는 경우 multiclusterhub.spec.imagePullSecret 을 정의해야 합니다. 이 시크릿은 Kubernetes용 Red Hat Advanced Cluster Management를 설치할 때 생성될 수 있습니다. 시크릿 정의에 대한 자세한 내용은 사용자 정의 이미지 가져오기 보안을 참조하십시오.

1.7.2.2. 지원되는 아키텍처

- Linux (x86_64, s390x, ppc64le)

- macOS

1.7.2.3. 가져오기 준비

1. *다음 명령을 실행하여 **hub** 클러스터에 로그인합니다.*

```
oc login
```

2. ***hub** 클러스터에서 다음 명령을 실행하여 프로젝트와 네임스페이스를 생성합니다. 참고: **CLUSTER_NAME** 에 정의된 클러스터 이름도 **YAML** 파일 및 명령에서 클러스터 네임스페이스로 사용됩니다.*

```
oc new-project ${CLUSTER_NAME}
```

*중요: **cluster.open-cluster-management.io/managedCluster** 레이블이 관리형 클러스터 네임스페이스에 자동으로 추가 및 제거됩니다. 관리형 클러스터에서 수동으로 추가하거나 제거하지 마십시오.*

3. *다음 예제 콘텐츠를 사용하여 **managed-cluster.yaml** 이라는 파일을 생성합니다.*

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: ${CLUSTER_NAME}
  labels:
    cloud: auto-detect
    vendor: auto-detect
spec:
  hubAcceptsClient: true
```

*클라우드 및 공급 업체 의 값이 자동 감지 로 설정된 경우 **Red Hat Advanced Cluster Management**는 가져올 클러스터에서 클라우드 및 벤더 유형을 자동으로 탐지합니다. 자동 감지 값을 클러스터의 클라우드 및 벤더 값으로 선택적으로 교체할 수 있습니다. 다음 예제를 참조하십시오.*

```
cloud: Amazon
vendor: OpenShift
```

4. *다음 명령을 입력하여 **ManagedCluster** 리소스에 **YAML** 파일을 적용합니다.*

```
oc apply -f managed-cluster.yaml
```

자동 가져오기 보안을 사용하여 클러스터 가져오기를 계속하거나 수동 명령으로 클러스터를 가져옵

니다.

1.7.2.4. 자동 가져오기 보안을 사용하여 클러스터 가져오기

자동 가져오기 보안을 사용하여 가져오려면 클러스터의 **kubeconfig** 파일 또는 클러스터의 **kube API** 서버 및 토큰 쌍에 대한 참조가 포함된 시크릿을 생성해야 합니다.

1. 가져올 클러스터의 **kubeconfig** 파일 또는 **kube API** 서버 및 토큰을 검색합니다. **kubeconfig** 파일 또는 **kube API** 서버 및 토큰을 찾을 위치를 알아보려면 **Kubernetes** 클러스터 설명서를 참조하십시오.

2. **${CLUSTER_NAME}** 네임스페이스에 **auto-import-secret.yaml** 파일을 생성합니다.

   a. 다음 템플릿과 유사한 콘텐츠가 포함된 **auto-import-secret.yaml** 이라는 **YAML** 파일을 생성합니다.

```
apiVersion: v1
kind: Secret
metadata:
  name: auto-import-secret
  namespace: <cluster_name>
stringData:
  autoImportRetry: "5"
  # If you are using the kubeconfig file, add the following value for the kubeconfig file
  # that has the current context set to the cluster to import:
  kubeconfig: |- <kubeconfig_file>
  # If you are using the token/server pair, add the following two values instead of
  # the kubeconfig file:
  token: <Token to access the cluster>
  server: <cluster_api_url>
type: Opaque
```

   b. 다음 명령을 사용하여 **${CLUSTER_NAME}** 네임스페이스에 **YAML** 파일을 적용합니다.

```
oc apply -f auto-import-secret.yaml
```

참고: 기본적으로 자동 가져오기 보안은 한 번 사용되며 가져오기 프로세스가 완료되면 삭제됩니다. 자동 가져오기 보안을 유지하려면 **managedcluster-import-controller.open-cluster-management.io/keeping-auto-import-secret** 을 시크릿에 추가합니다. 다음 명령을 실행하여 추가할 수 있습니다.

```
oc -n <cluster_name> annotate secrets auto-import-secret managedcluster-import-
controller.open-cluster-management.io/keeping-auto-import-secret=""
```

3. 가져온 클러스터에 대한 **status ED** 및 **AVAILABLE** 상태를 검증합니다. **hub** 클러스터에서 다음 명령을 실행합니다.

```
oc get managedcluster ${CLUSTER_NAME}
```

4. 관리형 클러스터에서 다음 명령을 실행하여 관리형 클러스터에 로그인합니다.

```
oc login
```

5. 다음 명령을 실행하여 가져올 클러스터에서 **Pod** 상태를 확인합니다.

```
oc get pod -n open-cluster-management-agent
```

klusterlet 추가 기능 가져오기를 계속합니다.

1.7.2.5. 수동 명령으로 클러스터 가져오기

중요: 가져오기 명령에는 가져온 각 클러스터에 복사되는 풀 시크릿 정보가 포함되어 있습니다. 가져온 클러스터에 액세스할 수 있는 모든 사용자는 풀 시크릿 정보도 볼 수 있습니다.

1. 다음 명령을 실행하여 **hub** 클러스터의 가져오기 컨트롤러에서 생성한 **klusterlet-crd.yaml** 파일을 가져옵니다.

```
oc get secret ${CLUSTER_NAME}-import -n ${CLUSTER_NAME} -o jsonpath=
{.data.crds\\.yaml} | base64 --decode > klusterlet-crd.yaml
```

2. 다음 명령을 실행하여 **hub** 클러스터에서 가져오기 컨트롤러에서 생성된 **import.yaml** 파일을 가져옵니다.

```
oc get secret ${CLUSTER_NAME}-import -n ${CLUSTER_NAME} -o jsonpath=
{.data.import\\.yaml} | base64 --decode > import.yaml
```

가져올 클러스터에서 다음 단계를 진행합니다.

3. 다음 명령을 입력하여 가져올 관리형 클러스터에 로그인합니다.

```
oc login
```

4. 다음 명령을 실행하여 1단계에서 생성한 **klusterlet-crd.yaml** 을 적용합니다.

```
oc apply -f klusterlet-crd.yaml
```

5. 다음 명령을 실행하여 이전에 생성한 **import.yaml** 파일을 적용합니다.

```
oc apply -f import.yaml
```

6. 가져올 클러스터에 대한 **status** 및 **AVAILABLE** 상태를 확인합니다. **hub** 클러스터에서 다음 명령을 실행합니다.

```
oc get managedcluster ${CLUSTER_NAME}
```

klusterlet 추가 기능 가져오기를 계속합니다.

1.7.2.6. klusterlet 애드온 가져오기

다음 절차를 완료하여 **klusterlet** 추가 기능 구성 파일을 생성하고 적용할 수 있습니다.

1. 다음 예와 유사한 **YAML** 파일을 생성합니다.

```
apiVersion: agent.open-cluster-management.io/v1
kind: KlusterletAddonConfig
metadata:
  name: <cluster_name>
  namespace: <cluster_name>
spec:
  applicationManager:
    enabled: true
  certPolicyController:
    enabled: true
  iamPolicyController:
    enabled: true
  policyController:
```

```
    enabled: true
  searchCollector:
    enabled: true
```

2. *파일을 **klusterlet-addon-config.yaml** 로 저장합니다.*

3. *다음 명령을 실행하여 **YAML**을 적용합니다.*

```
oc apply -f klusterlet-addon-config.yaml
```

*ManagedCluster-Import-Controller는 **${CLUSTER_NAME}-import.** **${CLUSTER_NAME}-import** 시크릿에는 사용자가 **klusterlet**을 설치하기 위해 사용자가 관리형 클러스터에 적용하는 **import.yaml** 이 포함되어 있습니다.*

*애드온은 가져오는 클러스터 후 설치됩니다. **AVAILABLE** 입니다.*

4. *다음 명령을 실행하여 가져오는 클러스터에서 애드온의 **Pod** 상태를 확인합니다.*

```
oc get pod -n open-cluster-management-agent-addon
```

이제 클러스터를 가져옵니다.

1.7.2.7. CLI를 사용하여 가져온 클러스터 제거

클러스터를 제거하려면 다음 명령을 실행합니다.

```
oc delete managedcluster ${CLUSTER_NAME}
```

***cluster_name** 을 클러스터 이름으로 교체합니다.*

이제 클러스터가 제거되었습니다.

1.7.3. 사용자 정의 ManagedClusterImageRegistry CRD를 사용하여 클러스터 가져오기

가져올 관리형 클러스터에서 이미지 레지스트리를 재정의해야 하는 경우가 있을 수 있습니다.

ManagedClusterImageRegistry CRD(사용자 정의 리소스 정의)를 생성하여 이 작업을 수행할 수 있습니다.

ManagedClusterImageRegistry CRD는 네임스페이스 범위 리소스입니다.

ManagedClusterImageRegistry CRD는 선택할 배치의 관리 클러스터 세트를 지정하지만 사용자 정의 이미지 레지스트리의 다른 이미지가 필요합니다. 관리형 클러스터가 새 이미지로 업데이트되면 식별을 위해 각 관리 클러스터에 다음 레이블이 추가됩니다. open-cluster-management.io/image-registry=<namespace>.<managedClusterImageRegistryName > .

다음 예제는 ManagedClusterImageRegistry CRD를 보여줍니다.

```
apiVersion: imageregistry.open-cluster-management.io/v1alpha1
kind: ManagedClusterImageRegistry
metadata:
  name: <imageRegistryName>
  namespace: <namespace>
spec:
  placementRef:
    group: cluster.open-cluster-management.io
    resource: placements
    name: <placementName>
  pullSecret:
    name: <pullSecretName>
  registries:
  - mirror: <mirrored-image-registry-address>
    source: <image-registry-address>
  - mirror: <mirrored-image-registry-address>
    source: <image-registry-address>
```

spec 섹션에서 다음을 수행합니다.

- *placementName 을 관리 클러스터 세트를 선택하는 동일한 네임스페이스에 있는 배치 이름으로 교체합니다.*
- *pullSecretName 을 사용자 정의 이미지 레지스트리에서 이미지를 가져오는 데 사용되는 풀 시크릿의 이름으로 교체합니다.*
- *각 소스 및 미러 레지스트리의 값을 나열합니다. mirrored-image-registry-address 및 image-registry-address 를 각 미러 값과 레지스트리의 소스 값으로 교체합니다.*

- 예 1: registry.redhat.io/rhacm2 라는 소스 이미지 레지스트리를 localhost:5000/rhacm2 로 교체하고 registry.redhat.io/multicluster-engine 을 localhost:5000/multicluster-engine 으로 교체하려면 다음 예제를 사용합니다.

```
registries:
- mirror: localhost:5000/rhacm2/
    source: registry.redhat.io/rhacm2
- mirror: localhost:5000/multicluster-engine
    source: registry.redhat.io/multicluster-engine
```

- 예 2: 소스 이미지 registry.redhat.io/rhacm2/registration-rhel8-operator 를 localhost:5000/rhacm2-registration-rhel8-operator 로 교체하려면 다음 예제를 사용합니다.

```
registries:
- mirror: localhost:5000/rhacm2-registration-rhel8-operator
    source: registry.redhat.io/rhacm2/registration-rhel8-operator
```

1.7.3.1. ManagedClusterImageRegistry CRD를 사용하여 클러스터 가져오기

ManagedClusterImageRegistry CRD로 클러스터를 가져오려면 다음 단계를 완료합니다.

1. 클러스터를 가져오려는 네임스페이스에 풀 시크릿을 생성합니다. 이러한 단계에서는 myNamespace 입니다.

```
$ kubectl create secret docker-registry myPullSecret \
  --docker-server=<your-registry-server> \
  --docker-username=<my-name> \
  --docker-password=<my-password>
```

2. 생성한 네임스페이스에 배치를 생성합니다.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: myPlacement
  namespace: myNamespace
spec:
  clusterSets:
  - myClusterSet
  tolerations:
  - key: "cluster.open-cluster-management.io/unreachable"
    operator: Exists
```

참고: 배치에서 클러스터를 선택하는 데 연결할 수 없는 허용 오차가 필요합니다.

3. ManagedClusterSet 리소스를 생성하여 네임스페이스에 바인딩합니다.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: ManagedClusterSet
metadata:
  name: myClusterSet

---
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: ManagedClusterSetBinding
metadata:
  name: myClusterSet
  namespace: myNamespace
spec:
  clusterSet: myClusterSet
```

4. 네임스페이스에 ManagedClusterImageRegistry CRD를 생성합니다.

```
apiVersion: imageregistry.open-cluster-management.io/v1alpha1
kind: ManagedClusterImageRegistry
metadata:
  name: myImageRegistry
  namespace: myNamespace
spec:
  placementRef:
    group: cluster.open-cluster-management.io
    resource: placements
    name: myPlacement
  pullSecret:
    name: myPullSecret
  registry: myRegistryAddress
```

5. Red Hat Advanced Cluster Management 콘솔에서 관리형 클러스터를 가져와서 관리형 클러스터 세트에 추가합니다.

6. open-cluster-management.io/image-registry=myNamespace.myImageRegistry 레이블이 관리형 클러스터에 추가된 후 관리 클러스터에서 가져오기 명령을 복사하고 실행합니다.

1.7.4. 클러스터의 klusterlet 추가 기능 설정 수정

KlusterletAddonConfig 의 설정을 수정하여 hub 클러스터를 사용하여 구성을 변경할 수 있습니다.

KlusterletAddonConfig 컨트롤러는 klusterletaddonconfigs.agent.open-cluster-management.io Kubernetes 리소스의 설정에 따라 활성화 및 비활성화된 기능을 관리합니다. KlusterletAddonConfig 의 다음 예제를 봅니다.

```
apiVersion: agent.open-cluster-management.io/v1
kind: KlusterletAddonConfig
metadata:
  name: <cluster-name>
  namespace: <cluster-name>
spec:
  clusterName: <cluster-name>
  clusterNamespace: <cluster-name>
  clusterLabels:
    cloud: auto-detect
    vendor: auto-detect
  applicationManager:
    enabled: true
  certPolicyController:
    enabled: true
  iamPolicyController:
    enabled: true
  policyController:
    enabled: true
  searchCollector:
    enabled: false
  version: 2.5.0
```

1.7.4.1. klusterlet 애드온 설정 설명

klusterletaddonconfigs.agent.open-cluster-management.io Kubernetes 리소스에서 다음 설정을 업데이트할 수 있습니다.

표 1.2. klusterlet 애드온 설정 테이블 목록

이름 설정	현재의	설명
applicationmanager	**true** 또는 **false**	이 컨트롤러는 관리형 클러스터의 애플리케이션 서브스크립션 라이프사이클을 관리합니다.
certPolicyController	**true** 또는 **false**	이 컨트롤러는 관리형 클러스터에 인증서 기반 정책을 적용합니다.
iamPolicyController	**true** 또는 **false**	이 컨트롤러는 관리 클러스터에 IAM 기반 정책 라이프사이클을 적용합니다.
policyController	**true** 또는 **false**	이 컨트롤러는 관리형 클러스터에 다른 모든 정책 규칙을 적용합니다.

이름 설정	현재의	설명
searchCollector	**true** 또는 **false**	이 컨트롤러는 리소스 인덱스 데이터를 hub 클러스터로 주기적으로 푸시하는 데 사용됩니다.

1.7.4.2. hub 클러스터에서 콘솔을 사용하여 수정

*hub 클러스터를 사용하여 **klusterletaddonconfigs.agent.open-cluster-management.io** 리소스의 설정을 수정할 수 있습니다. 설정을 변경하려면 다음 단계를 완료합니다.*

1. *허브 클러스터의 **Kubernetes** 콘솔용 **Red Hat Advanced Cluster Management**에 로그인합니다.*
2. *허브 클러스터 콘솔의 헤더 메뉴에서 검색 아이콘을 선택합니다.*
3. *검색 매개변수에 다음 값을 입력합니다. **kind:klusteraddonconfigs***
4. *업데이트할 끝점 리소스를 선택합니다.*
5. ***spec** 섹션을 찾아 **Edit** 를 선택하여 콘텐츠를 편집합니다.*
6. *설정을 수정합니다.*
7. *저장 을 선택하여 변경 사항을 적용합니다.*

1.7.4.3. hub 클러스터의 명령줄을 사용하여 수정

*hub 클러스터를 사용하여 설정을 수정하려면 **cluster-name** 네임스페이스에 액세스할 수 있어야 합니다. 다음 단계를 완료합니다.*

1. ***hub** 클러스터에 로그인합니다.*

2. 다음 명령을 입력하여 리소스를 편집합니다.

```
kubectl edit klusterletaddonconfigs.agent.open-cluster-management.io <cluster-name> -n <cluster-name>
```

3. **spec** 섹션을 찾습니다.

4. 필요에 따라 설정을 수정합니다.

1.8. 클러스터에 액세스

Kubernetes용 **Red Hat Advanced Cluster Management**에서 생성 및 관리하는 **Red Hat OpenShift Container Platform** 클러스터에 액세스하려면 다음 단계를 완료하십시오.

1. **Red Hat Advanced Cluster Management for Kubernetes** 탐색 메뉴에서 **Infrastructure > Clusters** 로 이동하여 생성했거나 액세스하려는 클러스터 이름을 선택합니다.

2. **Reveal credentials** 를 선택하여 클러스터의 사용자 이름과 암호를 확인합니다. 클러스터에 로그인할 때 사용할 다음 값을 기록해 두십시오.

   참고: 가져온 클러스터에서는 **Reveal credentials** 옵션을 사용할 수 없습니다.

3. 클러스터에 연결할 콘솔 **URL** 을 선택합니다.

4. **3**단계에서 찾은 사용자 **ID**와 암호를 사용하여 클러스터에 로그인합니다.

1.9. 프록시 환경에서 클러스터 생성

프록시 서버를 통해 허브 클러스터가 연결된 경우 **Red Hat OpenShift Container Platform** 클러스터를 생성할 수 있습니다.

클러스터 생성에 성공하려면 다음 상황 중 하나가 **true**여야 합니다.

-

Red Hat Advanced Cluster Management for Kubernetes에는 생성 중인 관리형 클러스터와의 사설 네트워크 연결이 있지만, Red Hat Advanced Cluster Management 및 관리형 클러스터는 프록시를 사용하여 인터넷에 액세스합니다.

- *관리 클러스터는 인프라 공급자에 있지만 방화벽 포트는 관리 클러스터에서 허브 클러스터로 통신할 수 있습니다.*

프록시로 구성된 클러스터를 생성하려면 다음 단계를 완료합니다.

1. *install-config.yaml 파일에 다음 정보를 추가하여 hub 클러스터에서 cluster-wide-proxy 설정을 구성합니다.*

```
apiVersion: v1
kind: Proxy
baseDomain: <domain>
proxy:
  httpProxy: http://<username>:<password>@<proxy.example.com>:<port>
  httpsProxy: https://<username>:<password>@<proxy.example.com>:<port>
  noProxy: <wildcard-of-domain>,<provisioning-network/CIDR>,<BMC-address-range/CIDR>
```

username 을 프록시 서버의 사용자 이름으로 바꿉니다.

프록시 서버에 액세스하려면 password 를 암호로 바꿉니다.

proxy.example.com 을 프록시 서버의 경로로 바꿉니다.

포트를 프록시 서버와 통신 포트로 바꿉니다.

wildcard-of-domain 을 프록시를 바이패스해야 하는 도메인의 항목으로 바꿉니다.

CIDR 표기법에서 provisioning-network/CIDR 를 provisioning 네트워크의 IP 주소 및 할당된 IP 주소 수로 바꿉니다.

CIDR 표기법에서 BMC-address-range/CIDR 를 BMC 주소 및 주소 수로 바꿉니다.

이전 값을 추가한 후 설정이 클러스터에 적용됩니다.

2. 클러스터 생성 절차를 완료하여 클러스터를 프로비저닝합니다. 공급자를 선택하려면 클러스터 생성을 참조하십시오.

1.9.1. 기존 클러스터 애드온에서 클러스터 전체 프록시 활성화

hub 클러스터에서 관리하는 Red Hat OpenShift Container Platform 클러스터의 모든 klusterletlet add-on Pod에 프록시 환경 변수를 추가하도록 클러스터 네임스페이스에서 KlusterletAddonConfig 를 구성할 수 있습니다.

klusterletAddonConfig의 Pod에 3개의 환경 변수를 추가하도록 KlusterletAddonConfig 를 구성하려면 다음 단계를 완료합니다.

1. 프록시를 추가해야 하는 클러스터의 네임스페이스에 있는 KlusterletAddonConfig 파일을 엽니다.

2. 다음 예와 같이 파일의 .spec.proxyConfig 섹션을 편집합니다.

```
spec
  proxyConfig:
    httpProxy: "<proxy_not_secure>"
    httpsProxy: "<proxy_secure>"
    noProxy: "<no_proxy>"
```

proxy_not_secure 를 http 요청에 대해 프록시 서버의 주소로 바꿉니다. 예: http://192.168.123.145:3128.

proxy_secure 를 https 요청에 대해 프록시 서버의 주소로 바꿉니다. 예: https://192.168.123.145:3128.

no_proxy 를 프록시를 통해 트래픽을 라우팅하지 않는 IP 주소, 호스트 이름 및 도메인 이름 목록으로 바꿉니다. 예: .cluster.local,.svc,10.128.0.0/14,example.com.

spec.proxyConfig 는 선택적 섹션입니다. Red Hat Advanced Cluster Management hub 클러스터에 구성된 클러스터 전체 프록시를 사용하여 OpenShift Container Platform 클러스터

클러스터가 생성되는 경우 다음 조건이 충족될 때 **klusterlet** 애드온의 **Pod**에 클러스터 전체 프록시 구성 값이 추가됩니다.

- **addon** 섹션의 **.spec.policyController.proxyPolicy** 가 활성화되고 **OCPGlobalProxy**로 설정됩니다.
- **.spec.applicationManager.proxyPolocy** 가 활성화되고 **CustomProxy** 로 설정됩니다.

  참고: 애드온 섹션의 **proxyPolicy** 기본값은 **Disabled** 입니다.

  다음 예제를 참조하십시오.

```
apiVersion: agent.open-cluster-management.io/v1
  kind: KlusterletAddonConfig
  metadata:
    name: clusterName
    namespace: clusterName
  spec:
    proxyConfig:
      httpProxy: http://pxuser:12345@10.0.81.15:3128
      httpsProxy: http://pxuser:12345@10.0.81.15:3128
      noProxy: .cluster.local,.svc,10.128.0.0/14, example.com
    applicationManager:
      enabled: true
      proxyPolicy: CustomProxy
    policyController:
      enabled: true
      proxyPolicy: OCPGlobalProxy
    searchCollector:
      enabled: true
      proxyPolicy: Disabled
    certPolicyController:
      enabled: true
      proxyPolicy: Disabled
    iamPolicyController:
      enabled: true
      proxyPolicy: Disabled
```

프록시는 클러스터 애드온에 구성됩니다.

중요: 글로벌 프록시 설정은 경고 전달에 영향을 미치지 않습니다. 클러스터 전체 프록시를 사용하여 **Red Hat Advanced Cluster Management hub** 클러스터에 대한 경고 전달을 설정하려면 자세한 내용은 경고 전달을 참조하십시오.

1.10. 클러스터 프록시 애드온 활성화

일부 환경에서는 관리형 클러스터가 방화벽 뒤에 있으며 **hub** 클러스터에서 직접 액세스할 수 없습니다. 액세스할 수 있도록 관리형 클러스터의 **kube-api** 서버에 액세스하도록 프록시 애드온을 설정하여 보다 안전한 연결을 제공할 수 있습니다.

필수 액세스: 편집기

hub 클러스터 및 관리 클러스터에 대한 클러스터 프록시 애드온을 구성하려면 다음 단계를 완료하십시오.

1. **Red Hat Advanced Cluster Management for Kubernetes hub** 클러스터에서 클러스터 프록시 애드온을 활성화합니다. 자세한 내용은 고급 구성을 참조하십시오.

2. 다음 단계를 완료하여 관리 클러스터 **kube-apiserver** 에 액세스하도록 **kubeconfig** 파일을 구성합니다.

   a. 관리 클러스터에 유효한 액세스 토큰을 제공합니다. 기본 서비스 계정이 **default** 네임스페이스에 있다고 가정하여 서비스 계정의 해당 토큰을 사용할 수 있습니다.

      i. 관리 클러스터의 컨텍스트를 사용하고 있는지 확인합니다. **managed-cluster.kubeconfig** 라는 파일이 관리 클러스터의 **kubeconfig** 파일이라고 가정합니다. 팁: **--kubeconfig=managed-cluster.kubeconfig** 가 있는 명령은 관리형 클러스터에서 실행되며 이 프로세스의 모든 명령은 동일한 콘솔에서 실행되어야 합니다. 다른 콘솔에서 명령을 실행하지 마십시오.

      ii. 다음 명령을 실행하여 **Pod**에 액세스할 수 있는 서비스 계정에 역할을 추가합니다.

```
oc create role -n default test-role --verb=list,get --resource=pods --
kubeconfig=managed-cluster.kubeconfig
oc create rolebinding -n default test-rolebinding --serviceaccount=default:default --
role=test-role --kubeconfig=managed-cluster.kubeconfig
```

      iii. 다음 명령을 실행하여 서비스 계정 토큰의 보안을 찾습니다.

```
oc get secret -n default --kubeconfig=managed-cluster.kubeconfig | grep default-
token
```

iv. *다음 명령을 실행하여 토큰을 복사합니다.*

```
export MANAGED_CLUSTER_TOKEN=$(kubectl --kubeconfig=managed-cluster.kubeconfig -n default get secret <default-token> -o jsonpath={.data.token} | base64 -d)
```

***default-token** 을 보안 이름으로 교체합니다.*

b. ***Red Hat Advanced Cluster Management hub** 클러스터에서 **kubeconfig** 파일을 구성합니다.*

i. *다음 명령을 실행하여 **hub** 클러스터에서 현재 **kubeconfig** 파일을 내보냅니다.*

```
oc config view --minify --raw=true > cluster-proxy.kubeconfig
```

ii. *편집기로 서버 파일을 수정합니다. 이 예제에서는 **sed** 를 사용할 때 명령을 사용합니다. **OSX**를 사용하는 경우 별칭 **sed=gsed** 를 실행합니다.*

```
export TARGET_MANAGE_CLUSTER=<cluster1>

export NEW_SERVER=https://$(oc get route -n open-cluster-management cluster-proxy-addon-user -o=jsonpath='{.spec.host}')/$TARGET_MANAGE_CLUSTER

sed -i'' -e '/server:/c\    server: '"$NEW_SERVER"'' cluster-proxy.kubeconfig

export CADATA=$(oc get configmap -n openshift-service-ca kube-root-ca.crt -o=go-template='{{index .data "ca.crt"}}' | base64)

sed -i'' -e '/certificate-authority-data:/c\    certificate-authority-data: '"$CADATA"'' cluster-proxy.kubeconfig
```

***cluster1** 을 액세스하려는 관리형 클러스터 이름으로 교체합니다.*

iii. *다음 명령을 입력하여 원래 사용자 자격 증명을 삭제합니다.*

```
sed -i'' -e '/client-certificate-data/d' cluster-proxy.kubeconfig
sed -i'' -e '/client-key-data/d' cluster-proxy.kubeconfig
sed -i'' -e '/token/d' cluster-proxy.kubeconfig
```

iv. 서비스 계정의 토큰을 추가합니다.

```
sed -i'' -e '$a\    token: "'$MANAGED_CLUSTER_TOKEN'"' cluster-proxy.kubeconfig
```

3. 다음 명령을 실행하여 대상 관리 클러스터의 대상 네임스페이스에 있는 모든 **Pod**를 나열합니다.

```
oc get pods --kubeconfig=cluster-proxy.kubeconfig -n <default>
```

default 네임스페이스를 사용하려는 네임스페이스로 바꿉니다.

이제 **hub** 클러스터가 관리 클러스터의 **kube-api** 와 통신합니다.

1.11. 특정 클러스터 관리 역할 구성

Kubernetes용 **Red Hat Advanced Cluster Management**를 설치할 때 기본 구성은 **Red Hat Advanced Cluster Management hub** 클러스터에 대한 **cluster-admin** 역할을 제공합니다. 이 권한을 사용하면 **hub** 클러스터에서 관리 클러스터를 생성, 관리 및 가져올 수 있습니다. 어떤 경우에는 **hub** 클러스터의 모든 관리형 클러스터에 대한 액세스 권한을 제공하는 대신 **hub** 클러스터에서 관리하는 특정 관리 클러스터에 대한 액세스를 제한해야 할 수 있습니다.

클러스터 역할을 정의하고 사용자 또는 그룹에 적용하여 특정 관리 클러스터에 대한 액세스를 제한할 수 있습니다. 역할을 구성하고 적용하려면 다음 단계를 완료합니다.

1. 다음 콘텐츠로 **YAML** 파일을 생성하여 클러스터 역할을 정의합니다.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: <clusterrole-name>
rules:
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  resourceNames:
  - <managed-cluster-name>
  verbs:
  - get
  - list
  - watch
```

```
  - update
  - delete
  - deletecollection
  - patch
- apiGroups:
  - cluster.open-cluster-management.io
  resources:
  - managedclusters
  verbs:
  - create
- apiGroups:
  - ""
  resources:
  - namespaces
  resourceNames:
  - <managed-cluster-name>
  verbs:
  - create
  - get
  - list
  - watch
  - update
  - delete
  - deletecollection
  - patch
- apiGroups:
  - register.open-cluster-management.io
  resources:
  - managedclusters/accept
  resourceNames:
  - <managed-cluster-name>
  verbs:
  - update
```

clusterrole-name 을 생성 중인 클러스터 역할의 이름으로 바꿉니다.

사용자가 액세스할 수 있도록 **managed-cluster-name** 을 관리 클러스터의 이름으로 교체합니다.

2. 다음 명령을 입력하여 **clusterrole** 정의를 적용합니다.

```
oc apply <filename>
```

파일 이름을 이전 단계에서 생성한 **YAML** 파일의 이름으로 바꿉니다.

3. 다음 명령을 입력하여 **clusterrole** 을 지정된 사용자 또는 그룹에 바인딩합니다.

```
oc adm policy add-cluster-role-to-user <clusterrole-name> <username>
```

clusterrole-name 을 이전 단계에서 적용한 클러스터 역할의 이름으로 교체합니다. **username** 을 클러스터 역할을 바인딩할 사용자 이름으로 바꿉니다.

1.12. 클러스터 라벨 관리

클러스터에 레이블을 추가하여 그룹 리소스를 선택합니다. 자세한 내용은 레이블 및 선택기를 참조하십시오.

새 레이블을 추가하고, 기존 레이블을 제거하고, 클러스터의 기존 라벨을 편집할 수 있습니다.

레이블을 관리하려면 **Infrastructure > Clusters** 로 이동하여 **Clusters** 테이블에서 클러스터를 찾습니다. 클러스터의 옵션 메뉴를 사용하여 라벨 편집을 선택합니다.

- 새 레이블을 추가하려면 라벨 편집 대화 상자에 레이블을 입력합니다. 입력한 항목은 **Key=Value** 형식이어야 합니다. 레이블을 여러 개 추가하는 경우 를 입력하거나, 쉼표를 추가하거나, 레이블 사이에 공백을 추가하여 라벨을 구분합니다.

  레이블은 저장 을 클릭한 후에만 저장됩니다.

- 기존 레이블을 제거하려면 목록에서 제거할 레이블의 제거 아이콘을 클릭합니다.

- 기존 레이블을 업데이트하려면 다른 값이 있는 동일한 키를 사용하여 새 레이블을 추가하여 해당 키를 새 값에 다시 할당할 수 있습니다. 예를 들어 **Key=NewValue** 를 입력하여 **Key=Value** 값을 업데이트하여 **Key =Value**를 변경할 수 있습니다.

팁: 클러스터 세부 정보 페이지에서 클러스터 레이블을 편집할 수도 있습니다. 탐색 메뉴에서 인프라 > 클러스터를 클릭합니다. 클러스터 페이지에서 클러스터 이름을 클릭하여 클러스터의 세부 정보 페이지에 액세스합니다. 라벨 섹션에서 편집 아이콘을 선택합니다. 라벨 편집 대화 상자가 표시됩니다.

1.13. 관리 클러스터에서 실행되도록 ANSIBLE TOWER 작업 구성

Red Hat Advanced Cluster Management는 **Ansible Tower** 자동화와 통합되어 클러스터를 생성하거나 업그레이드하기 전이나 후에 발생하는 **prehook** 및 **posthook AnsibleJob** 인스턴스를 생성할 수 있습

니다. 클러스터 제거에 대한 **prehook** 및 **posthook** 작업을 구성하고 클러스터 스케일링 작업은 지원되지 않습니다.

필수 액세스 권한: 클러스터 관리자

- 사전 요구 사항
- 콘솔을 사용하여 클러스터에서 실행되도록 **AnsibleJob** 템플릿 구성
- **AnsibleJob** 템플릿 생성
- 라벨을 사용하여 관리 클러스터에서 실행되도록 **AnsibleJob** 템플릿 구성
- **Ansible** 작업 상태 보기

1.13.1. 사전 요구 사항

Red Hat Advanced Cluster Management 클러스터에서 **Ansible** 템플릿을 실행하려면 다음 사전 요구 사항을 충족해야 합니다.

- **OpenShift Container Platform 4.6** 이상
- **Ansible Automation Platform Resource Operator**를 설치하여 **Ansible** 작업을 **Git** 서브스크립션 라이프사이클에 연결합니다. **AnsibleJob**을 사용하여 **Ansible Tower** 작업을 시작할 때 최상의 결과를 얻으려면 **Ansible Tower** 작업 템플릿이 멱등이어야 합니다. **OpenShift Container Platform OperatorHub** 에서 **Ansible Automation Platform Resource Operator**를 찾을 수 있습니다.

Ansible Tower 자동화 설치 및 구성에 대한 자세한 내용은 **Ansible** 작업 설정을 참조하십시오.

1.13.2. 콘솔을 사용하여 클러스터에서 실행되도록 AnsibleJob 템플릿 구성

클러스터를 생성할 때 클러스터에 사용할 **Ansible** 작업 템플릿을 지정해야 합니다. 클러스터를 생성할 때 템플릿을 지정하려면 자동화 단계에서 클러스터에 적용할 **Ansible** 템플릿을 선택합니다. **Ansible** 템

플릿이 없는 경우 자동화 템플릿 추가를 클릭하여 템플릿을 생성합니다.

1.13.3. AnsibleJob 템플릿 생성

*클러스터 설치 또는 업그레이드를 사용하여 **Ansible** 작업을 시작하려면 작업을 실행할 시기를 지정할 **Ansible** 작업 템플릿을 생성해야 합니다. 클러스터 설치 또는 업그레이드 전후에 실행되도록 구성할 수 있습니다.*

*템플릿을 생성하는 동안 **Ansible** 템플릿 실행에 대한 세부 정보를 지정하려면 콘솔의 단계를 완료합니다.*

1. ***Red Hat Advanced Cluster Management** 탐색에서 **Infrastructure > Automation** 을 선택합니다.*

2. *상황에 맞는 경로를 선택합니다.*

   - *새 템플릿을 생성하려면 **Create Ansible template** 을 클릭하고 **3**단계를 계속합니다.*
   - *기존 템플릿을 수정하려면 수정하려는 템플릿의 옵션 메뉴에서 템플릿 편집 을 클릭하고 **5**단계로 진행합니다.*

3. *템플릿의 고유 이름을 입력합니다. 소문자 영숫자 또는 하이픈(-)을 포함합니다.*

4. *새 템플릿에 사용할 인증 정보를 선택합니다. **Ansible** 자격 증명을 **Ansible** 템플릿에 연결하려면 다음 단계를 완료합니다.*

   a. ***Red Hat Advanced Cluster Management** 탐색에서 자동화 를 선택합니다. 인증 정보에 연결되지 않은 템플릿 목록에 있는 템플릿에는 템플릿을 기존 인증 정보에 연결하는 데 사용할 수 있는 인증 정보 링크 아이콘이 포함되어 있습니다. 템플릿과 동일한 네임스페이스에 있는 인증 정보만 표시됩니다.*

   b. *선택할 수 있는 인증 정보가 없거나 기존 인증 정보를 사용하지 않으려면 연결할 템플릿의 옵션 메뉴에서 템플릿 편집 을 선택합니다.*

   c. *인증 정보 추가 를 클릭하고 인증 정보를 생성해야 하는 경우 **Ansible Automation***

Platform에 대한 인증 정보 생성 절차를 완료합니다.

d. *템플릿과 동일한 네임스페이스에 인증 정보를 생성한 후 템플릿을 편집할 때 Ansible Automation Platform 인증 정보 필드에서 인증 정보를 선택합니다.*

5. *클러스터를 설치하기 전에 Ansible 작업을 시작하려면 Pre-install Ansible 작업 템플릿 섹션에서 Add an Ansible 작업 템플릿 섹션을 선택합니다.*

6. *클러스터 설치 또는 업그레이드에 추가할 prehook 및 posthook Ansible 작업의 이름을 선택하거나 입력합니다.*

   참고: Ansible 작업 템플릿 이름은 Ansible Tower의 Ansible 작업 이름과 일치해야 합니다.

7. *필요한 경우 Ansible 작업을 끌어서 순서를 변경합니다.*

8. *클러스터가 설치 후 Ansible 작업 템플릿 섹션, Pre -upgrade Ansible 작업 템플릿 섹션 및 업그레이드 Ansible 작업 템플릿 섹션에 설치 하려는 모든 Ansible 작업 템플릿에 대해 5 - 7단계를 반복합니다.*

Ansible 템플릿은 지정된 작업이 발생할 때 이 템플릿을 지정하는 클러스터에서 실행되도록 구성됩니다.

1.13.4. 라벨을 사용하여 관리 클러스터에서 실행되도록 AnsibleJob 템플릿 구성

Red Hat Advanced Cluster Management for Kubernetes에서 클러스터를 생성하거나 라벨을 사용하여 Red Hat Advanced Cluster Management로 가져올 때 클러스터에 바인딩되는 AnsibleJob 을 생성할 수 있습니다.

Ansible 작업을 생성하고 Red Hat Advanced Cluster Management에서 아직 관리하지 않은 클러스터로 구성하려면 다음 절차를 완료합니다.

1. *애플리케이션 함수에서 지원하는 채널 중 하나에서 Ansible 작업에 대한 정의 파일을 생성합니다. Git 채널만 지원됩니다.*

   정의에서 kind 값으로 AnsibleJob 을 사용합니다.

정의 파일 내용은 다음 예와 유사합니다.

```
apiVersion: apiVersion: tower.ansible.com/v1alpha1
kind: AnsibleJob
metadata:
  name: hive-cluster-gitrepo
spec:
  tower_auth_secret: my-toweraccess
  job_template_name: my-tower-template-name
  extra_vars:
    variable1: value1
    variable2: value2
```

prehook 또는 **posthook** 디렉터리에 파일을 저장하면 배치 규칙과 일치하는 클러스터 이름 목록이 생성됩니다. 클러스터 이름 목록은 **AnsibleJob** 종류 리소스에 **extra_vars** 값으로 전달할 수 있습니다. 이 값이 **AnsibleJob** 리소스에 전달되면 **Ansible** 작업에서 새 클러스터 이름을 확인하고 자동화에 사용할 수 있습니다.

2. **Red Hat Advanced Cluster Management hub** 클러스터에 로그인합니다.

3. **Red Hat Advanced Cluster Management** 콘솔을 사용하여 방금 생성한 정의 파일을 저장한 채널을 참조하는 **Git** 서브스크립션이 있는 애플리케이션을 생성합니다. 애플리케이션 및 서브스크립션 생성에 대한 자세한 내용은 애플리케이션 리소스 관리를 참조하십시오.

서브스크립션을 생성할 때 이 서브스크립션을 클러스터와 연결하기 위해 생성하거나 나중에 가져오는 클러스터에 추가할 수 있는 레이블을 지정합니다. **vendor=OpenShift** 와 같은 기존 레이블 또는 생성 및 정의하는 고유한 레이블일 수 있습니다.

참고: 이미 사용 중인 레이블을 선택하면 **Ansible** 작업이 자동으로 실행됩니다. **prehooks** 또는 **posthooks**의 일부가 아닌 애플리케이션에 리소스를 포함하는 것이 좋습니다.

기본 배치 규칙은 **AnsibleJob** 의 레이블과 일치하는 라벨을 사용하여 클러스터를 감지할 때 작업을 실행합니다. 허브 클러스터에서 관리하는 실행 중인 모든 클러스터에서 자동화를 실행하려면 배치 규칙에 다음 내용을 추가합니다.

```
clusterConditions:
  - type: ManagedClusterConditionAvailable
    status: "True"
```

배치 규칙의 **YAML** 콘텐츠에 붙여넣거나 **Red Hat Advanced Cluster Management** 콘솔의 애플리케이션 생성 페이지의 모든 온라인 클러스터 및 로컬 클러스터에 배포하는 옵션을 선택할

수 있습니다.

4. 클러스터 생성 또는 대상 관리 클러스터를 각각 허브 클러스터로 가져오기의 지침에 따라 클러스터를 만들거나 가져옵니다.

클러스터를 생성하거나 가져올 때 서브스크립션을 만들 때 사용한 것과 동일한 레이블을 사용하며 **AnsibleJob** 은 클러스터에서 실행되도록 자동으로 구성됩니다.

Red Hat Advanced Cluster Management는 **AnsibleJob.extra_vars.target_clusters** 경로에 클러스터 이름을 자동으로 삽입합니다. 클러스터 이름을 정의에 동적으로 삽입할 수 있습니다. **AnsibleJob**을 생성하고 **Red Hat Advanced Cluster Management**에서 이미 관리하는 클러스터로 구성하려면 다음 절차를 완료합니다.

1. **Git** 채널의 **prehook** 또는 **posthook** 디렉터리에서 **AnsibleJob**에 대한 정의 파일을 생성합니다.

정의에서 **kind** 값으로 **AnsibleJob** 을 사용합니다.

정의 파일 내용은 다음 예와 유사합니다.

```
apiVersion: tower.ansible.com/v1alpha1
kind: AnsibleJob
metadata:
  name: hive-cluster-gitrepo
spec:
  tower_auth_secret: my-toweraccess
  job_template_name: my-tower-template-name
  extra_vars:
    variable1: value1
    variable2: value2
```

my-toweraccess 를 인증 시크릿으로 교체하여 **Ansible Tower**에 액세스합니다.

my-tower-template-name 을 **Ansible Tower**의 템플릿 이름으로 교체합니다.

Ansible 작업에서 제어되는 클러스터가 제거 또는 추가될 때마다 **AnsibleJob**은 **extra_vars.target_clusters** 변수를 자동으로 실행하고 업데이트합니다. 이번 업데이트를 통해 특정 자동화로 클러스터 이름을 지정하거나 클러스터 그룹에 자동화를 적용할 수 있습니다.

1.13.5. Ansible 작업 상태 보기

실행 중인 **Ansible** 작업의 상태를 보고 시작했으며 성공적으로 실행되고 있는지 확인할 수 있습니다. 실행 중인 **Ansible** 작업의 현재 상태를 보려면 다음 단계를 완료합니다.

1. **Red Hat Advanced Cluster Management** 메뉴에서 **Infrastructure > Clusters** 를 선택하여 클러스터 페이지에 액세스합니다.

2. 클러스터 이름을 선택하여 세부 정보를 확인합니다.

3. 클러스터 정보에 대해 **Ansible** 작업의 마지막 실행 상태를 봅니다. 이 항목은 다음 상태 중 하나를 보여줍니다.

   - 설치 **prehook** 또는 **posthook** 작업이 실패하면 클러스터 상태에 **Failed** 가 표시됩니다.

   - 업그레이드 **prehook** 또는 **posthook** 작업이 실패하면 업그레이드에 실패한 배포 필드에 경고가 표시됩니다.

     팁: 클러스터 **prehook** 또는 **posthook** 가 실패한 경우 클러스터 페이지에서 업그레이드를 재시도할 수 있습니다.

1.14. MANAGEDCLUSTERSETS 생성 및 관리

ManagedClusterSet 은 관리형 클러스터 그룹입니다. 관리형 클러스터 세트를 사용하면 그룹의 모든 관리형 클러스터에 대한 액세스를 함께 관리할 수 있습니다. **ManagedClusterSetBinding** 리소스를 생성하여 **ManagedClusterSet** 리소스를 네임스페이스에 바인딩할 수도 있습니다.

각 관리 클러스터는 **ManagedClusterSet** 의 멤버여야 합니다. **hub** 클러스터를 설치하면 **default** 라고 하는 기본 **ManagedClusterSet** 이 생성됩니다. 관리형 클러스터 세트에 특별히 할당되지 않은 모든 관리형 클러스터는 기본 관리 클러스터 세트에 자동으로 할당됩니다. 기본 관리 클러스터 세트를 항상 사용할 수 있도록 하려면 기본 관리 클러스터 세트를 삭제하거나 업데이트할 수 없습니다.

참고: **ManagedClusterSet**에 특별히 추가되지 않은 클러스터 풀은 기본 **ManagedClusterSet** 에 추가되지 않습니다. 관리형 클러스터가 클러스터 풀에서 요청되면 다른 **ManagedClusterSet** 에 특별히 추가되지 않는 경우 기본 **ManagedClusterSet** 에 추가됩니다.

- ManagedClusterSet 생성
- ManagedClusterSet에 사용자 또는 그룹 역할 기반 액세스 제어 권한 할당
- ManagedClusterSetBinding 리소스 생성
- ManagedClusterSet에 클러스터 추가
- ManagedClusterSet에서 클러스터 제거

1.14.1. ManagedClusterSet 생성

관리형 클러스터에서 함께 관리 클러스터를 그룹화하여 관리형 클러스터에서 사용자 액세스를 제한할 수 있습니다.

필수 액세스: 클러스터 관리자

ManagedClusterSet 은 클러스터 범위 리소스이므로 **ManagedClusterSet** 을 생성하는 클러스터의 클러스터 관리 권한이 있어야 합니다. 관리 클러스터는 둘 이상의 **ManagedClusterSet** 에 포함할 수 없습니다. **Kubernetes** 콘솔의 **Red Hat Advanced Cluster Management** 또는 명령줄 인터페이스에서 관리형 클러스터 세트를 생성할 수 있습니다.

1.14.1.1. 콘솔을 사용하여 ManagedClusterSet 생성

Red Hat Advanced Cluster Management 콘솔을 사용하여 설정된 관리형 클러스터를 생성하려면 다음 단계를 완료합니다.

1. 기본 콘솔 탐색에서 **인프라 > 클러스터**를 선택하고 **클러스터 설정** 탭이 선택되었는지 확인합니다.
2. **클러스터 세트 만들기** 를 선택하고 클러스터 세트의 이름을 입력합니다.

1.14.1.2. 명령줄을 사용하여 ManagedClusterSet 생성

관리형 클러스터 세트의 다음 정의를 **yaml** 파일에 추가하여 명령줄을 사용하여 관리되는 클러스터 세트를 생성합니다.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: ManagedClusterSet
metadata:
  name: <clusterset1>
```

clusterset1 을 관리 클러스터 세트의 이름으로 교체합니다.

1.14.2. ManagedClusterSet에 사용자 또는 그룹 역할 기반 액세스 제어 권한 할당

hub 클러스터에서 구성된 **ID** 공급자가 제공하는 사용자 또는 그룹을 클러스터에 할당할 수 있습니다.

필수 액세스 권한: 클러스터 관리자

ManagedClusterSet API는 두 가지 수준의 **RBAC** 권한을 제공합니다.

- 클러스터 세트 관리자
  - 관리 클러스터 세트에 할당된 모든 클러스터 및 클러스터 풀 리소스에 대한 전체 액세스 권한입니다.
  - 클러스터를 생성하고, 클러스터를 가져오고, 클러스터 풀을 생성할 수 있는 권한입니다. 관리 클러스터 세트가 생성될 때 권한을 관리 클러스터에 할당해야 합니다.
- 클러스터 세트 보기
  - 관리 클러스터 세트에 할당된 모든 클러스터 및 클러스터 풀 리소스에 대한 권한만 읽습니다.
  - 클러스터를 생성하거나 클러스터를 가져오거나 클러스터 풀을 생성할 수 있는 권한이 없습니다.

Red Hat Advanced Cluster Management 콘솔에서 관리형 클러스터에 사용자 또는 그룹을 할당하려면 다음 단계를 완료합니다.

1. 콘솔의 메인 탐색 메뉴에서 인프라 > 클러스터를 선택합니다.
2. 클러스터 세트 탭을 선택합니다.
3. 대상 클러스터 세트를 선택합니다.
4. 액세스 관리 탭을 선택합니다.
5. 사용자 또는 그룹 추가를 선택합니다.
6. 액세스 권한을 검색하고 제공할 사용자 또는 그룹을 선택합니다.
7. Cluster set admin 또는 Cluster set view 역할을 선택하여 선택한 사용자 또는 사용자 그룹에 제공합니다. 역할 권한에 대한 자세한 내용은 역할 개요 를 참조하십시오.
8. 추가 를 선택하여 변경 사항을 제출합니다.

사용자 또는 그룹이 테이블에 표시됩니다. 모든 관리 클러스터 세트 리소스가 사용자 또는 그룹에 전파되도록 권한 할당에 대해 권한 할당이 몇 초 정도 걸릴 수 있습니다.

역할 기반 작업에 대한 자세한 내용은 역할 기반 액세스 제어를 참조하십시오.

배치 정보는 ManagedClusterSets를 배치와 함께 사용을 참조하십시오.

1.14.2.1. ManagedClusterSetBinding 리소스 생성

ManagedClusterSetBinding 리소스를 생성하여 ManagedClusterSet 리소스를 네임스페이스에 바인딩합니다. 동일한 네임스페이스에서 생성되는 애플리케이션 및 정책은 바인딩된 관리형 클러스터 세트

리소스에 포함된 관리형 클러스터에만 액세스할 수 있습니다.

네임스페이스에 대한 액세스 권한은 해당 네임스페이스에 바인딩된 관리형 클러스터 세트에 자동으로 적용됩니다. 관리 클러스터 세트가 바인딩된 네임스페이스에 액세스할 수 있는 액세스 권한이 있는 경우 해당 네임스페이스에 바인딩된 모든 관리 클러스터 세트에 액세스할 수 있는 권한이 자동으로 부여됩니다. 그러나 관리형 클러스터 세트에 액세스할 수 있는 권한만 있는 경우 해당 네임스페이스의 다른 관리 클러스터 세트에 액세스할 수 있는 권한이 자동으로 없습니다. 관리형 클러스터 세트가 표시되지 않으면 이를 확인하는 데 필요한 권한이 없을 수 있습니다.

콘솔 또는 명령줄을 사용하여 관리형 클러스터 세트 바인딩을 생성할 수 있습니다.

1.14.2.1.1. 콘솔을 사용하여 ManagedClusterSetBinding 생성

Red Hat Advanced Cluster Management 콘솔을 사용하여 설정된 관리형 클러스터에서 클러스터를 제거하려면 다음 단계를 완료합니다.

1. *기본 탐색에서 인프라 > 클러스터를 선택하고 클러스터 세트 탭을 선택하여 클러스터 페이지에 액세스합니다.*
2. *클러스터 세트 세부 정보를 보려면 바인딩을 생성할 클러스터 세트의 이름을 선택합니다.*
3. *작업 > 네임스페이스 바인딩 편집 을 선택합니다.*
4. *네임스페이스 바인딩 편집 페이지의 드롭다운 메뉴에서 클러스터 세트를 바인딩할 네임스페이스를 선택합니다. 클러스터 세트에 대한 바인딩이 있는 기존 네임스페이스가 이미 선택되어 있습니다.*

1.14.2.1.2. 명령줄을 사용하여 ManagedClusterSetBinding 생성

명령줄을 사용하여 관리형 클러스터 세트 바인딩을 생성하려면 다음 단계를 완료합니다.

1. *yaml 파일에 ManagedClusterSetBinding 리소스를 생성합니다. 관리형 클러스터 세트 바인딩을 생성할 때 관리형 클러스터 세트 바인딩의 이름이 바인딩되도록 관리형 클러스터 세트의 이름과 일치해야 합니다. ManagedClusterSetBinding 리소스는 다음 정보와 유사합니다.*

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: ManagedClusterSetBinding
```

```
metadata:
  namespace: project1
  name: clusterset1
spec:
  clusterSet: clusterset1
```

2. 대상 관리 클러스터에 대한 바인딩 권한이 설정되어 있는지 확인합니다. 사용자가 **clusterset1** 에 바인딩할 수 있는 규칙이 포함된 **ClusterRole** 리소스의 다음 예제를 봅니다.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: clusterrole1
rules:
  - apiGroups: ["cluster.open-cluster-management.io"]
    resources: ["managedclustersets/bind"]
    resourceNames: ["clusterset1"]
    verbs: ["create"]
```

1.14.3. ManagedClusterSet에 클러스터 추가

ManagedClusterSet 을 생성한 후 하나 이상의 관리 클러스터를 추가해야 합니다. 콘솔 또는 명령줄을 사용하여 설정된 관리형 클러스터를 관리 클러스터에 추가할 수 있습니다.

1.14.3.1. 콘솔을 사용하여 ManagedClusterSet에 클러스터 추가

Red Hat Advanced Cluster Management 콘솔을 사용하여 설정된 관리형 클러스터에 클러스터를 추가하려면 다음 단계를 완료합니다.

1. 관리형 클러스터 세트를 방금 만든 경우 리소스 할당 관리를 선택하여 리소스 할당 관리 페이지로 직접 이동합니다. 이 절차의 **6**단계를 계속 진행합니다.

2. 클러스터가 이미 존재하는 경우 기본 탐색에서 인프라 **>** 클러스터를 선택하여 클러스터 페이지에 액세스합니다.

3. 사용 가능한 클러스터 세트를 보려면 클러스터 세트 탭을 선택합니다.

4. 클러스터 세트 세부 정보를 보려면 관리형 클러스터 세트에 추가할 클러스터 세트의 이름을 선택합니다.

-

5. *작업 > 리소스 할당 관리를 선택합니다.*

6. *리소스 할당 관리 페이지에서 클러스터 세트에 추가할 리소스의 확인란을 선택합니다.*

7. *검토를 선택하여 변경 사항을 검토합니다.*

8. *저장 을 선택하여 변경 사항을 저장합니다.*

참고: 관리 대상 클러스터에서 다른 클러스터로 설정된 관리형 클러스터를 이동하는 경우 두 관리 클러스터 세트 모두에서 필요한 RBAC 권한을 사용할 수 있어야 합니다.

1.14.3.2. 명령줄을 사용하여 ManagedClusterSet에 클러스터 추가

명령줄을 사용하여 설정된 관리형 클러스터에 클러스터를 추가하려면 다음 단계를 완료합니다.

1. *managedclustersets/join 의 가상 하위 리소스에서 생성할 수 있는 RBAC ClusterRole 항목이 있는지 확인합니다. 이 권한이 없으면 관리 클러스터를 ManagedClusterSet 에 할당할 수 없습니다.*

   이 항목이 없으면 yaml 파일에 추가합니다. 샘플 항목은 다음 내용과 유사합니다.

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: clusterrole1
rules:
  - apiGroups: ["cluster.open-cluster-management.io"]
    resources: ["managedclustersets/join"]
    resourceNames: ["<clusterset1>"]
    verbs: ["create"]
```

   clusterset1 을 ManagedClusterSet 의 이름으로 교체합니다.

   참고: 관리 클러스터를 하나의 ManagedClusterSet 에서 다른 클러스터로 이동하는 경우 두 관리 클러스터 세트 모두에서 사용 가능한 권한이 있어야 합니다.

2. *yaml 파일에서 관리 클러스터의 정의를 찾습니다. 레이블을 추가하는 관리형 클러스터 정의*

의 섹션은 다음 내용과 유사합니다.

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
spec:
  hubAcceptsClient: true
```

이 예제에서 **cluster1** 은 관리 클러스터의 이름입니다.

3. **ManagedClusterSet** 의 이름을 **cluster.open-cluster-management.io/clusterset: clusterset1** 형식으로 지정하는 레이블을 추가합니다.

코드는 다음 예와 유사합니다.

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
  labels:
    cluster.open-cluster-management.io/clusterset: clusterset1
spec:
  hubAcceptsClient: true
```

이 예에서 **cluster1** 은 관리형 클러스터 세트 이름 **clusterset1** 에 추가된 클러스터입니다.

참고: 관리형 클러스터가 삭제된 관리형 클러스터 세트에 이전에 할당된 경우 관리 클러스터에 존재하지 않는 클러스터 세트에 이미 지정된 관리형 클러스터 세트가 있을 수 있습니다. 이 경우 이름을 새 이름으로 바꿉니다.

1.14.4. ManagedClusterSet에서 관리형 클러스터 제거

관리형 클러스터 세트에서 관리 클러스터를 제거하여 다른 관리 클러스터 세트로 이동하거나 세트의 관리 설정에서 제거할 수 있습니다. 콘솔 또는 명령줄 인터페이스를 사용하여 설정된 관리형 클러스터에서 관리형 클러스터를 제거할 수 있습니다.

참고: 관리되는 모든 클러스터는 관리형 클러스터 세트에 할당해야 합니다. **ManagedClusterSet** 에서 관리 클러스터를 제거하고 다른 **ManagedClusterSet** 에 할당하지 않으면 기본 관리형 클러스터 세트에 자동으로 추가됩니다.

1.14.4.1. 콘솔을 사용하여 ManagedClusterSet에서 관리 클러스터 제거

Red Hat Advanced Cluster Management 콘솔을 사용하여 설정된 관리형 클러스터에서 클러스터를 제거하려면 다음 단계를 완료합니다.

1. *관리형 클러스터 세트를 방금 만든 경우 리소스 할당 관리를 선택하여 리소스 할당 관리 페이지로 직접 이동합니다. 이 절차의 5단계를 계속 진행합니다.*

2. *클러스터가 이미 존재하는 경우 기본 탐색에서 인프라 > 클러스터를 선택하고 클러스터 설정 탭이 선택되어 있는지 확인하여 클러스터 페이지에 액세스합니다.*

3. *클러스터 세트 세부 정보를 보려면 관리형 클러스터에서 제거할 클러스터 세트의 이름을 선택합니다.*

4. *작업 > 리소스 할당 관리를 선택합니다.*

5. *리소스 할당 관리 페이지에서 클러스터 세트에서 제거할 리소스의 확인란을 선택합니다.*

   이 단계에서는 이미 클러스터 세트의 멤버인 리소스를 제거하거나 클러스터 세트의 멤버가 아닌 리소스를 추가합니다. 관리 클러스터의 세부 정보를 확인하여 리소스가 이미 클러스터 세트의 멤버인지 확인할 수 있습니다.

참고: 관리 대상 클러스터에서 다른 클러스터로 설정된 관리 클러스터를 이동하는 경우 두 관리 클러스터 세트에 대해 필요한 RBAC 권한이 있어야 합니다.

1.14.4.2. 명령줄을 사용하여 ManagedClusterSet에서 클러스터 제거

명령줄을 사용하여 설정된 관리형 클러스터에서 관리형 클러스터를 제거하려면 다음 단계를 완료하십시오.

1. *다음 명령을 실행하여 관리형 클러스터 세트의 관리형 클러스터 목록을 표시합니다.*

   ```
   oc get managedclusters -l cluster.open-cluster-management.io/clusterset=<clusterset1>
   ```

   clusterset1 을 관리 클러스터 세트의 이름으로 교체합니다.

2. 제거할 클러스터의 항목을 찾습니다.

3. 제거할 클러스터의 **yaml** 항목에서 레이블을 제거합니다. 레이블 예제는 다음 코드를 참조하십시오.

```
labels:
  cluster.open-cluster-management.io/clusterset: clusterset1
```

참고: 관리 대상 클러스터에서 다른 클러스터로 설정된 관리 클러스터를 이동하는 경우 두 관리 클러스터 세트 모두에서 필요한 **RBAC** 권한을 사용할 수 있어야 합니다.

1.14.5. 배치와 함께 ManagedClusterSets 사용

배치 리소스는 배치 네임스페이스에 바인딩된 **ManagedClusterSets** 에서 **ManagedClusters** 세트를 선택하는 규칙을 정의하는 네임스페이스 범위 리소스입니다.

필수 액세스: 클러스터 관리자, 클러스터 세트 관리자

1.14.5.1. 배치 개요

관리형 클러스터의 배치 작동 방식에 대한 다음 정보를 참조하십시오.

- **Kubernetes** 클러스터는 클러스터 범위의 **ManagedClusters** 로 허브 클러스터에 등록됩니다.
- **ManagedClusters** 는 클러스터 범위 **ManagedClusterSets** 로 구성됩니다.
- **ManagedClusterSets** 는 워크로드 네임스페이스에 바인딩됩니다.
- 네임스페이스 범위 배치는 잠재적인 **ManagedClusters** 의 작업 세트를 선택하는 **ManagedClusterSets** 의 일부를 지정합니다.
- 배치는 레이블 및 클레임 선택기를 사용하여 해당 작업 세트에서 선택합니다.

중요: 배치 네임스페이스에 바인딩된 **ManagedCluster Set** 이 없는 경우 **ManagedCluster** 를 선택하지 않습니다.

- **ManagedClusters** 배치는 테인트 및 톨러레이션을 사용하여 제어할 수 있습니다. 자세한 내용은 테인트 및 허용 오차를 사용하여 관리 클러스터 배치를 참조하십시오.

배치 사양에는 다음 필드가 포함됩니다.

- **ClusterSets** 는 **ManagedClusters** 가 선택된 **ManagedClusterSets** 를 나타냅니다.
  - 지정하지 않으면 배치 네임스페이스에 바인딩된 **ManagedClusterSets** 에서 **ManagedClusters** 가 선택됩니다.
  - 지정된 경우 이 세트의 교집합에서 **ManagedClusters** 가 선택되고 **placement** 네임스페이스에 바인딩된 **ManagedClusterSets** 가 선택됩니다.
- **NumberOfClusters** 는 배치 요구 사항을 충족하는 원하는 **ManagedClusters** 수를 나타냅니다.

  지정하지 않으면 배치 요구 사항을 충족하는 모든 **ManagedClusters** 가 선택됩니다.
- 서술자 는 레이블 및 클레임 선택기가 있는 **ManagedClusters** 를 선택하는 서술자 슬라이스를 나타냅니다. 서술자는 **ORed**입니다.
- **prioritizerPolicy** 는 우선순위 정책을 나타냅니다.
  - 모드는 **Exact,Additive, ""** 입니다. 여기서 **""** 는 기본적으로 **Additive** 입니다.
    - **Additive** 모드에서는 구성 값이 특별히 제공되지 않는 모든 우선순위가 기본 구성으로 활성화됩니다. 현재 기본 구성에서 **CloudEvent ady** 및 **Balance** 우선순위에는 가중치가 **1**이고 다른 우선순위는 **0**입니다. 나중에 기본 구성이 변경될 수 있으므로 우선순위가 변경될 수 있습니다. 추가 모드에서는 우선순위를 모두 구성할 필요가 없습니다.
    -

Exact 모드에서 구성 값과 함께 특별히 제공되지 않는 모든 우선순위에는 가중치가 0입니다. exact 모드를 사용하려면 원하는 우선순위의 전체 세트를 입력해야 하지만 릴리스 간 동작 변경은 방지할 수 있습니다.

- *구성은 우선순위 지정자의 구성을 나타냅니다.*
  - **scoreCoordinate**는 우선순위 및 점수 소스의 구성을 나타냅니다.
    - *type 은 우선순위가 높은 점수의 유형을 정의합니다. 유형은 BuiltIn,AddOn, " ", 여기서 " " 는 기본적으로 BuiltIn 입니다. 유형이 BuiltIn 인 경우 내장 우선 순위 이름 을 지정해야 합니다. 유형이 AddOn 인 경우 AddOn 에서 점수 소스를 구성해야 합니다.*
    - *builtin은 BuiltIn 우선순위r의 이름을 정의합니다. 다음 목록에는 유효한 BuiltIn 우선순위 지정자 이름이 포함되어 있습니다.*
      - *균형: 클러스터 간 결정에 균형을 유지합니다.*
      - *steady: 기존 결정이 안정되었는지 확인합니다.*
      - *ResourceAllocatableCPU 및 ResourceAllocatableMemory: 할당 가능한 리소스를 기반으로 클러스터를 정렬합니다.*
    - *Addon 은 리소스 이름과 점수 이름을 정의합니다. AddOnPlacementScore 는 애드온 점수를 설명하기 위해 도입되었습니다. 자세한 내용은 확장 가능 예약을 참조하십시오.*
      - *resourceName 은 AddOnPlacementScore 의 리소스 이름을 정의합니다. 배치 우선순위는 이 이름으로 AddOnPlacementScore 사용자 정의 리소스를 선택합니다.*
      - *scoreName 은 AddOnPlacementScore 내부의 점수 이름을 정의합니다. AddOnPlacementScore 에는 점수 이름 목록과 점수 값이 포함되어 있습니다. scoreName 은 우선순위에서 사용할 점수를 지정합니다.*

weight 는 우선순위의 가중치를 정의합니다. 값은 **[-10,10]** 범위에 있어야 합니다. 각 우선순위는 **[-100, 100]** 범위에서 클러스터의 정수 점수를 계산합니다. 클러스터의 최종 점수는 다음 공식 합계(**weight * priorityr_score**) 에 따라 결정됩니다. 가중치가 높아지면 우선 순위가 클러스터 선택에서 더 높은 가중치를 수신하지만 가중치 **0**은 우선순위가 비활성화되었음을 나타냅니다. 음수 가중치는 마지막 선택된 항목 중 하나임을 나타냅니다.

참고: **configurations.name** 파일은 **v1beta1**에서 제거되고 **scoreCoordinate.builtIn** 파일로 대체됩니다. **name** 및 **scoreCoordinate.builtIn** 이 모두 정의된 경우 **scoreCoordinate.builtIn** 의 값을 사용하여 선택을 결정합니다.

1.14.5.2. 배치 예

해당 네임스페이스에서 **ManagedClusterSet Binding** 을 생성하여 하나 이상의 **ManagedClusterSet**을 네임스페이스에 바인딩해야 합니다. 참고: **managedclustersets/bind** 의 가상 하위 리소스에서 **CREATE** 에 대한 역할 기반 액세스가 필요합니다. 다음 예제를 참조하십시오.

- **labelSelector** 를 사용하여 **ManagedClusters** 를 선택할 수 있습니다. **labelSelector** 만 라벨 공급 업체의 클러스터와 일치하는 다음 샘플을 참조하십시오. **OpenShift :**

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement1
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
        labelSelector:
          matchLabels:
            vendor: OpenShift
```

- **claimSelector** 를 사용하여 **ManagedClusters** 를 선택할 수 있습니다. **claimSelector** 만 **region.open-cluster-management.io** 와 **us-west-1** 과 일치하는 다음 샘플을 참조하십시오.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement2
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
        claimSelector:
          matchExpressions:
            - key: region.open-cluster-management.io
```

```
          operator: In
          values:
            - us-west-1
```

- 특정 clusterSets 에서 ManagedClusters 를 선택할 수 있습니다. claimSelector 만 clusterSets: clusterset1 clusterset2 와 일치하는 다음 샘플을 참조하십시오.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement3
  namespace: ns1
spec:
  clusterSets:
    - clusterset1
    - clusterset2
  predicates:
    - requiredClusterSelector:
        claimSelector:
          matchExpressions:
            - key: region.open-cluster-management.io
              operator: In
              values:
                - us-west-1
```

- 원하는 수의 ManagedClusters 를 선택합니다. numberOfClusters 가 3 인 다음 샘플을 참조하십시오.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement4
  namespace: ns1
spec:
  numberOfClusters: 3
  predicates:
    - requiredClusterSelector:
        labelSelector:
          matchLabels:
            vendor: OpenShift
        claimSelector:
          matchExpressions:
            - key: region.open-cluster-management.io
              operator: In
              values:
                - us-west-1
```

- 가장 큰 할당 가능 메모리가 있는 클러스터를 선택합니다.

참고: Kubernetes Node Allocatable 과 달리 '모든 할당 가능'은 각 클러스터의 Pod에 사용할 수 있는 컴퓨팅 리소스의 양으로 정의됩니다.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement6
  namespace: ns1
spec:
  numberOfClusters: 1
  prioritizerPolicy:
    configurations:
      - scoreCoordinate:
          builtIn: ResourceAllocatableMemory
```

- *할당 가능한 가장 큰 CPU 및 메모리가 있는 클러스터를 선택하고 리소스 변경에 민감하게 배치합니다.*

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement7
  namespace: ns1
spec:
  numberOfClusters: 1
  prioritizerPolicy:
    configurations:
      - scoreCoordinate:
          builtIn: ResourceAllocatableCPU
        weight: 2
      - scoreCoordinate:
          builtIn: ResourceAllocatableMemory
        weight: 2
```

- *할당 가능한 가장 큰 메모리와 가장 큰 애드온 점수 cpu 비율이 있는 두 클러스터를 선택하고 배치 결정을 고정합니다.*

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement8
  namespace: ns1
spec:
  numberOfClusters: 2
  prioritizerPolicy:
    mode: Exact
    configurations:
      - scoreCoordinate:
          builtIn: ResourceAllocatableMemory
      - scoreCoordinate:
```

```
      builtIn: Steady
    weight: 3
  - scoreCoordinate:
      type: AddOn
      addOn:
        resourceName: default
        scoreName: cpuratio
```

1.14.5.3. 배치 결정

cluster.open-cluster-management.io/placement name} 레이블이 있는 하나 이상의 Placement Decision s 가 생성되어 배치에서 선택한 ManagedCluster 를 나타냅니다.

ManagedCluster 를 선택하고 Placement Decision 에 추가하면 이 배치를 사용하는 구성 요소가 이 ManagedCluster 에 워크로드를 적용할 수 있습니다. ManagedCluster 가 더 이상 선택되지 않고 PlacementDecisions 에서 제거된 후 이 ManagedCluster 에 적용되는 워크로드를 적절하게 제거해야 합니다.

다음 PlacementDecision 샘플을 참조하십시오.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: PlacementDecision
metadata:
  labels:
    cluster.open-cluster-management.io/placement: placement1
  name: placement1-kbc7q
  namespace: ns1
  ownerReferences:
    - apiVersion: cluster.open-cluster-management.io/v1beta1
      blockOwnerDeletion: true
      controller: true
      kind: Placement
      name: placement1
      uid: 05441cf6-2543-4ecc-8389-1079b42fe63e
status:
  decisions:
    - clusterName: cluster1
      reason: ''
    - clusterName: cluster2
      reason: ''
    - clusterName: cluster3
      reason: ''
```

1.14.5.4. 애드온 상태

배포된 애드온의 상태에 따라 배치에 사용할 관리 클러스터를 선택할 수 있습니다. 예를 들어 클러스터에 활성화된 특정 추가 기능이 있는 경우에만 배치용으로 관리 클러스터를 선택하려고 합니다.

*배치를 생성할 때 애드온의 라벨과 필요한 경우 상태를 지정하여 이 작업을 수행할 수 있습니다. 클러스터에서 애드온이 활성화된 경우 **ManagedCluster** 리소스에 레이블이 자동으로 생성됩니다. 애드온이 비활성화된 경우 라벨이 자동으로 제거됩니다.*

*각 애드온은 **feature.open-cluster-management.io/addon-<addon-<addon_name>=<status_of_addon >** 형식의 레이블로 표시됩니다.*

***addon_name** 을 선택할 관리 클러스터에서 활성화해야 하는 애드온의 이름으로 바꿉니다.*

*클러스터를 선택한 경우 **status_of_addon** 을 애드온에 보유해야 하는 상태로 교체합니다. **status_of_addon** 의 가능한 값은 다음 목록에 있습니다.*

- *사용 가능: 애드온이 활성화되어 사용 가능합니다.*
- *비정상: 애드온이 활성화되어 있지만 리스가 지속적으로 업데이트되지 않습니다.*
- ***unreachable:** 애드온이 활성화되어 있지만 리스가 없습니다. 이는 관리 클러스터가 오프라인 상태일 때도 발생할 수 있습니다.*

*예를 들어 사용 가능한 **application-manager** 애드온은 다음과 같은 관리 대상 클러스터의 레이블로 표시됩니다.*

```
feature.open-cluster-management.io/addon-application-manager: available
```

애드온 및 해당 상태를 기반으로 배치를 생성하는 다음 예제를 참조하십시오.

- *다음 **YAML** 콘텐츠를 추가하여 **application-manager** 가 활성화된 모든 관리 클러스터를 포함하는 배치를 생성할 수 있습니다.*

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement1
  namespace: ns1
spec:
  predicates:
```

```
    - requiredClusterSelector:
        labelSelector:
          matchExpressions:
            - key: feature.open-cluster-management.io/addon-application-manager
              operator: Exists
```

- 다음 YAML 콘텐츠를 추가하여 application-manager 가 사용 가능한 상태로 활성화된 모든 관리 클러스터를 포함하는 배치를 생성할 수 있습니다.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement2
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
        labelSelector:
          matchLabels:
            "feature.open-cluster-management.io/addon-application-manager": "available"
```

- 다음 YAML 콘텐츠를 추가하여 application-manager 가 비활성화된 모든 관리 클러스터를 포함하는 배치를 생성할 수 있습니다.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement3
  namespace: ns1
spec:
  predicates:
    - requiredClusterSelector:
        labelSelector:
          matchExpressions:
            - key: feature.open-cluster-management.io/addon-application-manager
              operator: DoesNotExist
```

1.14.5.5. 확장 가능한 예약

배치 리소스 기반 예약에서 우선순위에는 관리 클러스터의 점수를 계산하기 위해 MananagedCluster 리소스에서 제공하는 기본값보다 더 많은 데이터가 필요한 경우가 있습니다. 예를 들어 모니터링 시스템을 통해 가져온 클러스터의 CPU 또는 메모리 사용량 데이터를 기반으로 클러스터를 예약합니다.

API AddOnPlacementScore 는 사용자 정의 점수를 기반으로 보다 확장 가능한 일정을 설정하는 방법을 지원합니다.

- **placement.yaml** 파일의 점수를 지정하여 클러스터를 선택할 수 있습니다.

- 점수 제공자로서 타사 컨트롤러는 허브 클러스터 또는 관리 클러스터에서 실행되어 **AddOnPlacementScore** 의 라이프사이클을 유지하고 이에 대한 업데이트 점수를 유지할 수 있습니다.

자세한 내용은 오픈 클러스터 관리 리포지토리에서 확장 가능한 스케줄링 기능 배치를 참조하십시오.

1.14.6. 테인트 및 허용 오차를 사용하여 관리 클러스터 배치

테인트 및 톨러레이션을 사용하여 관리 클러스터 또는 관리형 클러스터 세트의 배치를 제어할 수 있습니다. 테인트 및 허용 오차는 관리 클러스터가 특정 배치에 대해 선택되지 않도록 하는 방법을 제공합니다. 특정 관리 클러스터가 일부 배치에 포함되지 않도록 하려면 이 제어가 유용할 수 있습니다. 관리형 클러스터에 테인트를 추가하고 배치에 허용 오차를 추가할 수 있습니다. 테인트 및 허용 오차가 일치하지 않으면 해당 배치에 대해 관리 클러스터가 선택되지 않습니다.

1.14.6.1. 관리형 클러스터에 테인트 추가

테인트는 관리형 클러스터의 속성에 지정되며, 배치가 관리 클러스터 또는 관리되는 클러스터 집합을 거절할 수 있습니다. 다음 예와 유사한 명령을 입력하여 관리형 클러스터에 테인트를 추가할 수 있습니다.

```
kubectl taint ManagedCluster <managed_cluster_name> key=value:NoSelect
```

테인트 사양에는 다음 필드가 포함됩니다.

- 필수 키 - 클러스터에 적용되는 테인트 키입니다. 이 값은 해당 배치에 추가되는 기준을 충족하는 관리형 클러스터의 허용 오차 값과 일치해야 합니다. 이 값을 확인할 수 있습니다. 예를 들어 이 값은 **bar** 또는 **foo.example.com/bar** 일 수 있습니다.

- 선택사항 값 - **taint** 키의 **taint** 값입니다. 이 값은 해당 배치에 추가되는 기준을 충족하는 관리형 클러스터의 허용 오차 값과 일치해야 합니다. 예를 들어 이 값은 **value** 일 수 있습니다.

- 필수 **Effect** - 테인트를 허용하지 않는 배치 또는 배치의 허용 오차가 일치하지 않을 때 발생하는 항목에 대한 테인트의 영향입니다. 효과 값은 다음 값 중 하나여야 합니다.

  - **NoSelect** - 배치는 이 테인트를 허용하지 않는 한 클러스터를 선택할 수 없습니다. 테인트를 설정하기 전에 배치에 의해 클러스터를 선택하면 배치 결정에서 클러스터가 제거됩니

다.

- **NoSelectIfNew -** 스케줄러가 새 클러스터인 경우 클러스터를 선택할 수 없습니다. 배치는 테인트를 허용하고 클러스터가 이미 클러스터가 결정되도록 하는 경우에만 클러스터를 선택할 수 있습니다.

- 필수 **TimeAdded -** 테인트가 추가된 시간입니다. 이 값은 자동으로 설정됩니다.

1.14.6.2. 관리형 클러스터의 상태를 반영하기 위해 기본 제공 테인트 식별

관리형 클러스터에 액세스할 수 없는 경우 클러스터를 배치에 추가하지 않도록 합니다. 다음 테인트는 액세스할 수 없는 관리형 클러스터에 자동으로 추가됩니다.

- **cluster.open-cluster-management.io/unavailable -** 이 테인트는 클러스터에 **False** 상태의 **ManagedClusterConditionAvailable** 조건이 있는 경우 관리 클러스터에 추가됩니다. 테인트는 **NoSelect** 의 효과가 있으며, 사용할 수 없는 클러스터가 예약되지 않도록 하는 빈 값이 비어 있습니다. 이 테인트의 예는 다음 콘텐츠에 제공됩니다.

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
 name: cluster1
spec:
 hubAcceptsClient: true
 taints:
   - effect: NoSelect
     key: cluster.open-cluster-management.io/unavailable
     timeAdded: '2022-02-21T08:11:54Z'
```

- **cluster.open-cluster-management.io/unreachable -** 이 테인트는 **ManagedClusterConditionAvailable** 조건의 상태가 **Unknown** 이거나 조건이 없는 경우 관리형 클러스터에 추가됩니다. 테인트는 **NoSelect** 의 효과가 있으며 연결할 수 없는 클러스터가 예약되지 않도록 하는 빈 값이 비어 있습니다. 이 테인트의 예는 다음 콘텐츠에 제공됩니다.

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
spec:
  hubAcceptsClient: true
  taints:
    - effect: NoSelect
      key: cluster.open-cluster-management.io/unreachable
      timeAdded: '2022-02-21T08:11:06Z'
```

1.14.6.3. 배치에 허용 오차 추가

허용 오차는 배치에 적용되며 배치 허용 오차와 일치하는 테인트가 없는 관리형 클러스터를 배치할 수 있습니다. 허용 오차 사양에는 다음 필드가 포함됩니다.

- *선택 사항 키 - 키가 배치를 허용하는 **taint** 키와 일치합니다.*
- *선택 사항 - 허용 오차의 값은 배치를 허용하려면 허용 오차의 테인트 값과 일치해야 합니다.*
- *선택적 **Operator** - 연산자는 키와 값 간의 관계를 나타냅니다. 유효한 연산자는 동일 하고 존재합니다. 기본값은 동일합니다. 키가 동일하고 효과가 동일할 때 허용 오차가 테인트와 일치하며 **Operator**는 다음 값 중 하나입니다.*
  - ***equal** - Operator가 동일 하고 값은 테인트 및 허용 오차에서 동일합니다.*
  - ***exists** - 값의 와일드카드로, 배치가 특정 카테고리의 모든 테인트를 허용할 수 있습니다.*
- *선택사항 **Effect** - 일치시킬 테인트 효과입니다. 비워 두면 모든 테인트 효과와 일치합니다. 지정된 경우 허용되는 값은 **NoSelect** 또는 **NoSelectIfNew** 입니다.*
- *선택 사항 **TolerationSeconds** - 관리 클러스터를 새 배치로 이동하기 전에 허용 오차가 테인트를 허용하는 시간(초)입니다. 효과 값이 **NoSelect** 또는 **PreferNoSelect** 가 아닌 경우 이 필드는 무시됩니다. 기본값은 **nil** 로, 시간 제한이 없음을 나타냅니다. **TolerationSeconds** 의 계산 시작 시간은 클러스터 예약 시간 값 또는 **TolerationSeconds** 추가 시간이 아닌 테인트의 **TimeAdded** 값으로 자동으로 나열됩니다.*

다음 예제에서는 테인트가 있는 클러스터를 허용하는 허용 오차를 구성하는 방법을 보여줍니다.

- *이 예에는 관리형 클러스터의 테인트가 있습니다.*

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
spec:
  hubAcceptsClient: true
```

```
  taints:
    - effect: NoSelect
      key: gpu
      value: "true"
      timeAdded: '2022-02-21T08:11:06Z'
```

- 테인트를 허용할 수 있는 배치에 대한 허용 오차

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Placement
metadata:
  name: placement1
  namespace: default
spec:
  tolerations:
    - key: gpu
      value: "true"
      operator: Equal
```

허용 오차 예제를 정의하면 **key: gpu** 및 **value: "true"** 가 일치하므로 배치에서 **cluster1** 을 선택할 수 있습니다.

참고: 관리형 클러스터는 테인트에 대한 허용 오차가 포함된 배치에 배치할 수 없습니다. 다른 배치에 동일한 허용 오차가 포함된 경우 관리형 클러스터가 해당 배치 중 하나에 배치될 수 있습니다.

1.14.6.4. 임시 허용 오차 지정

TolerationSeconds 값은 허용 오차가 테인트를 허용하는 기간을 지정합니다. 이 임시 허용 오차는 관리 클러스터가 오프라인 상태이고 허용되는 시간 동안 이 클러스터에 배포된 애플리케이션을 다른 관리형 클러스터로 전송할 수 있는 경우 유용할 수 있습니다.

예를 들어 다음 테인트가 있는 관리형 클러스터에 연결할 수 없게 됩니다.

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  name: cluster1
spec:
  hubAcceptsClient: true
  taints:
    - effect: NoSelect
      key: cluster.open-cluster-management.io/unreachable
      timeAdded: '2022-02-21T08:11:06Z'
```

다음 예와 같이 **TolerationSeconds** 값을 사용하여 배치를 정의하는 경우 워크로드는 **5**분 후에 사용 가능한 다른 관리 클러스터로 전송됩니다.

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: Placement
metadata:
  name: demo4
  namespace: demo1
spec:
  tolerations:
    - key: cluster.open-cluster-management.io/unreachable
      operator: Exists
      tolerationSeconds: 300
----
```

관리 클러스터에 **5**분 동안 연결할 수 없는 후 애플리케이션을 다른 관리형 클러스터로 이동합니다.

1.15. 클러스터 풀 관리 (기술 프리뷰)

클러스터 풀은 필요에 따라 구성된 **Red Hat OpenShift Container Platform** 클러스터에 신속하고 비용 효율적인 액세스를 제공합니다. 클러스터 풀은 필요할 때 요청할 수 있는 **Amazon Web Services, Google Cloud Platform** 또는 **Microsoft Azure**에서 구성 가능하고 확장 가능한 수의 **OpenShift Container Platform** 클러스터를 프로비저닝합니다. 개발, 지속적 통합 및 프로덕션 시나리오를 위해 클러스터 환경을 제공하거나 교체할 때 특히 유용합니다. 실행을 유지할 여러 클러스터를 지정할 수 있으며, 클러스터의 나머지는 몇 분 내에 다시 시작하고 요청할 수 있도록 클러스터의 나머지 부분을 하향식 상태로 유지할 수 있습니다.

ClusterClaim 리소스는 클러스터 풀에서 클러스터를 확인하는 데 사용됩니다. 클러스터 클레임이 생성되면 풀에서 실행 중인 클러스터를 할당합니다. 실행 중인 클러스터를 사용할 수 없는 경우 클러스터를 제공하기 위해 하이버네이션 클러스터가 다시 시작되거나 새 클러스터가 프로비저닝됩니다. 클러스터 풀은 새 클러스터를 자동으로 생성하고 계층화 클러스터를 다시 시작하여 풀에서 지정된 크기와 사용 가능한 실행 클러스터 수를 유지합니다.

참고: 클러스터 풀에서 요청한 클러스터가 더 이상 필요하지 않고 삭제되면 리소스가 삭제됩니다. 클러스터가 클러스터 풀로 돌아가지 않습니다.

필수 액세스: 관리자

- 사전 요구 사항

- *클러스터 풀 생성*
- *클러스터 풀에서 클러스터 요청*
- *클러스터 풀 스케일링*
- *클러스터 풀 릴리스 이미지 업데이트*
- *클러스터 풀 삭제*

클러스터 풀을 생성하는 절차는 클러스터를 생성하는 절차와 유사합니다. 클러스터 풀의 클러스터는 즉시 사용할 수 있도록 생성되지 않습니다.

1.15.1. 사전 요구 사항

클러스터 풀을 생성하기 전에 다음 사전 요구 사항을 참조하십시오.

- ***Kubernetes** 허브용 **Red Hat Advanced Cluster Management** 클러스터를 배포해야 합니다.*
- *공급자 환경에서 **Kubernetes** 클러스터를 생성할 수 있도록 **Red Hat Advanced Cluster Management for Kubernetes hub** 클러스터에 대한 인터넷 액세스가 필요합니다.*
- ***AWS, GCP** 또는 **Microsoft Azure** 공급자 인증 정보가 필요합니다. 자세한 내용은 인증 정보 관리 개요 를 참조하십시오.*
- *공급자 환경에 구성된 도메인이 필요합니다. 도메인을 구성하는 방법에 대한 자세한 내용은 공급자 설명서를 참조하십시오.*
- *공급자 로그인 자격 증명이 필요합니다.*
- ***OpenShift Container Platform** 이미지 풀 시크릿이 필요합니다. 이미지 풀 시크릿 사용을*

참조하십시오.

*참고: 이 절차를 사용하여 클러스터 풀을 추가하면 풀에서 클러스터를 요청할 때 **Red Hat Advanced Cluster Management**에서 관리할 클러스터를 자동으로 가져올 수 있습니다. 클러스터 클레임을 사용하여 관리하기 위해 클레임된 클러스터를 자동으로 가져오지 않는 클러스터 풀을 생성하려면 **clusterClaim** 리소스에 다음 주석을 추가합니다.*

```
kind: ClusterClaim
metadata:
  annotations:
    cluster.open-cluster-management.io/createmanagedcluster: "false"
```

"false" 라는 단어는 문자열임을 나타내기 위해 따옴표로 묶어야 합니다.

1.15.2. 클러스터 풀 생성

*클러스터 풀을 생성하려면 탐색 메뉴에서 인프라 > 클러스터를 선택합니다. 클러스터 풀 탭에는 액세스할 수 있는 클러스터 풀이 나열됩니다. **Create cluster pool** 을 선택하고 콘솔의 단계를 완료합니다.*

클러스터 풀에 사용하려는 인프라 인증 정보가 없는 경우 인증 정보 추가 를 선택하여 생성합니다.

목록에서 기존 네임스페이스를 선택하거나 생성할 새 네임스페이스의 이름을 입력할 수 있습니다. 클러스터 풀은 클러스터와 동일한 네임스페이스에 있을 필요가 없습니다.

*클러스터 세트에서 클러스터 풀을 생성하면 네임스페이스 **admin** 권한이 클러스터 풀을 추가하는 네임스페이스에 대한 **clusterset admin** 권한이 있는 모든 사용자에게 적용됩니다. 마찬가지로 네임스페이스 보기 권한은 **clusterset** 보기 권한이 있는 사용자에게 적용됩니다.*

*클러스터 풀의 **RBAC** 역할을 기존 클러스터 세트의 역할 할당을 공유하도록 하려면 클러스터 세트 이름을 선택할 수 있습니다. 클러스터 풀의 클러스터에 대한 클러스터 세트는 클러스터 풀을 생성할 때만 설정할 수 있습니다. 클러스터 풀을 생성한 후에는 클러스터 풀 또는 클러스터 풀의 클러스터에 대한 클러스터 세트 연결을 변경할 수 없습니다. 클러스터 풀에서 요청하는 클러스터는 클러스터 풀과 동일한 클러스터에 자동으로 추가됩니다.*

참고: 클러스터 관리자 권한이 없는 경우 클러스터 세트를 선택해야 합니다. 이 경우 클러스터 세트 이름을 포함하지 않으면 클러스터 세트 생성 요청이 금지된 오류로 인해 거부됩니다. 선택할 수 있는 클러스터 세트가 없는 경우 클러스터 관리자에게 문의하여 클러스터 세트를 생성하고 클러스터 세트 관리자 권한을 부여합니다.

클러스터 풀 크기에서는 클러스터 풀에서 프로비저닝할 클러스터 수를 지정하는 반면, 실행 중인 클러스터 풀은 풀이 계속 실행되고 즉시 사용할 수 있도록 요청할 수 있는 클러스터 수를 지정합니다.

절차는 클러스터 생성 절차와 매우 유사합니다.

공급자에 필요한 정보에 대한 자세한 내용은 다음 정보를 참조하십시오.

- Amazon Web Services에서 클러스터 생성
- Google Cloud Platform에서 클러스터 생성
- Microsoft Azure에서 클러스터 생성

1.15.3. 클러스터 풀에서 클러스터 요청

ClusterClaim 리소스는 클러스터 풀에서 클러스터를 확인하는 데 사용됩니다. 클러스터가 실행 중이고 클러스터 풀에서 준비되면 클레임이 완료됩니다. 클러스터 풀은 클러스터 풀에 지정된 요구 사항을 유지하기 위해 클러스터 풀에 실행 중인 새 클러스터를 자동으로 생성합니다.

참고: 클러스터 풀에서 요청한 클러스터가 더 이상 필요하지 않고 삭제되면 리소스가 삭제됩니다. 클러스터가 클러스터 풀로 돌아가지 않습니다.

필수 액세스: 관리자

1.15.3.1. 사전 요구 사항

클러스터 풀에서 클러스터를 요청하기 전에 다음을 사용할 수 있어야 합니다.

사용 가능한 클러스터가 있거나 없는 클러스터 풀입니다. 클러스터 풀에 사용 가능한 클러스터가 있는 경우 사용 가능한 클러스터가 요청됩니다. 클러스터 풀에 사용 가능한 클러스터가 없는 경우 클레임을 충족하기 위해 클러스터가 생성됩니다. 클러스터 풀을 생성하는 방법에 대한 자세한 내용은 클러스터 풀 생성을 참조하십시오.

1.15.3.2. 클러스터 풀에서 클러스터 클레임

클러스터 클레임을 생성할 때 클러스터 풀에서 새 클러스터를 요청합니다. 클러스터를 사용할 수 있는 경우 클러스터에서 풀에서 확인합니다. 자동 가져오기를 비활성화하지 않는 한 클레임된 클러스터는 관리 클러스터 중 하나로 자동으로 가져옵니다.

클러스터를 요청하려면 다음 단계를 완료합니다.

1. *탐색 메뉴에서 인프라 > 클러스터를 클릭하고 클러스터 풀 탭을 선택합니다.*
2. *클러스터를 클레임할 클러스터 풀의 이름을 찾아 클레임 클러스터를 선택합니다.*

*클러스터를 사용할 수 있는 경우 요청되고 **Managed** 클러스터 탭에 즉시 표시됩니다. 사용 가능한 클러스터가 없는 경우 클러스터를 다시 시작하거나 새 클러스터를 프로비저닝하는 데 몇 분이 걸릴 수 있습니다. 이 기간 동안 클레임 상태는 보류 중입니다. 클러스터 풀을 확장하여 보류 중인 클레임을 보거나 삭제합니다.*

클레임된 클러스터는 클러스터 풀에 있을 때 연결된 클러스터 세트의 멤버로 유지됩니다. 클레임할 때 클레임된 클러스터의 클러스터 세트를 변경할 수 없습니다.

1.15.4. 클러스터 풀 스케일링

클러스터 풀 크기의 클러스터 수를 늘리거나 줄여 클러스터 풀의 클러스터 수를 변경할 수 있습니다.

필수 액세스: 클러스터 관리자

클러스터 풀의 클러스터 수를 변경하려면 다음 단계를 완료합니다.

1. *탐색 메뉴에서 인프라 > 클러스터를 클릭합니다.*
2. *클러스터 풀 탭을 선택합니다.*

3. *변경할 클러스터 풀의 옵션 메뉴에서 스케일링 클러스터 풀 을 선택합니다.*

4. *풀 크기의 값을 변경합니다.*

5. *필요한 경우 실행 중인 클러스터의 수를 업데이트하여 클레임할 때 즉시 사용 가능한 클러스터 수를 늘리거나 줄일 수 있습니다.*

새 값을 반영하도록 클러스터 풀이 확장됩니다.

1.15.5. 클러스터 풀 릴리스 이미지 업데이트

*클러스터 풀의 클러스터가 잠시 동안 정지 상태로 남아 있으면 클러스터의 **Red Hat OpenShift Container Platform** 릴리스 이미지가 역순 상태가 될 수 있습니다. 이 경우 클러스터 풀에 있는 클러스터의 릴리스 이미지 버전을 업그레이드할 수 있습니다.*

필수 액세스: 편집

*클러스터 풀의 클러스터의 **OpenShift Container Platform** 릴리스 이미지를 업데이트하려면 다음 단계를 완료합니다.*

참고: 이 절차에서는 클러스터 풀에 이미 클레임된 클러스터 풀에서 클러스터를 업데이트하지 않습니다. 이 절차를 완료하면 릴리스 이미지에 대한 업데이트는 클러스터 풀과 관련된 다음 클러스터에만 적용됩니다.

- *이 절차로 릴리스 이미지를 업데이트한 후 클러스터 풀에서 생성한 클러스터입니다.*
- *클러스터 풀에서 계층화된 클러스터입니다. 이전 릴리스 이미지가 있는 기존의 계층 구조 클러스터가 제거되고 새 릴리스 이미지가 있는 새 클러스터가 교체됩니다.*

1. *탐색 메뉴에서 인프라 **>** 클러스터를 클릭합니다.*

2. *클러스터 풀 탭을 선택합니다.*

3. 클러스터 풀 표에서 업데이트할 클러스터 풀의 이름을 찾습니다.

4. 표에서 클러스터 풀 의 옵션 메뉴를 클릭하고 릴리스 이미지 업데이트를 선택합니다.

5. 이 클러스터 풀에서 향후 클러스터 생성에 사용할 새 릴리스 이미지를 선택합니다.

클러스터 풀 릴리스 이미지가 업데이트되었습니다.

팁: 클러스터 풀 각각에 대한 상자를 선택하고 작업 메뉴를 사용하여 선택한 클러스터 풀의 릴리스 이미지를 업데이트하여 여러 클러스터 풀의 릴리스 이미지를 업데이트할 수 있습니다.

1.15.6. 클러스터 풀 삭제

클러스터 풀을 생성하고 더 이상 필요하지 않다고 결정하는 경우 클러스터 풀을 제거할 수 있습니다. 클러스터 풀을 삭제하면 승인되지 않은 하버네이션 클러스터가 모두 제거되고 해당 리소스가 해제됩니다.

필수 액세스: 클러스터 관리자

클러스터 풀을 삭제하려면 다음 단계를 완료합니다.

1. 탐색 메뉴에서 인프라 **>** 클러스터를 클릭합니다.

2. 클러스터 풀 탭을 선택합니다.

3. 삭제할 클러스터 풀의 옵션 메뉴에서 **Destroy** 클러스터 풀 삭제를 선택합니다. 클러스터 풀에서 처리되지 않은 클러스터는 모두 삭제됩니다. 모든 리소스를 삭제하는 데 시간이 걸릴 수 있으며 모든 리소스가 삭제될 때까지 클러스터 풀이 콘솔에 계속 표시됩니다.

ClusterPool이 포함된 네임스페이스는 삭제되지 않습니다. 이러한 클러스터의 **ClusterClaim** 리소스가 동일한 네임스페이스에 생성되므로 네임스페이스를 삭제하면 **ClusterPool**에서 요청한 모든 클러스터가 제거됩니다.

팁: 클러스터 풀 각각에 대한 상자를 선택하고 작업 메뉴를 사용하여 선택한 클러스터 풀을 제거함으로써 하나의 작업으로 여러 클러스터 풀을 제거할 수 있습니다.

1.16. CLUSTERCLAIMS

ClusterClaim 은 관리형 클러스터의 클러스터 범위의 CRD(사용자 정의 리소스 정의)입니다. ClusterClaim은 관리형 클러스터가 클레임하는 정보를 나타냅니다. 다음 예제는 YAML 파일에서 식별되는 클레임을 보여줍니다.

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ClusterClaim
metadata:
  name: id.openshift.io
spec:
  value: 95f91f25-d7a2-4fc3-9237-2ef633d8451c
```

다음 표는 Red Hat Advanced Cluster Management for Kubernetes가 관리하는 클러스터에 있을 수 있는 정의된 ClusterClaims를 보여줍니다.

클레임 이름	reserved	변경 가능	설명
id.k8s.io	true	false	업스트림 제안에 정의된 ClusterID
kubeversion.open-cluster-management.io	true	true	Kubernetes 버전
platform.open-cluster-management.io	true	false	관리형 클러스터가 AWS, GCE, Equinix Metal과 같은 플랫폼에서 실행 중입니다.
product.open-cluster-management.io	true	false	OpenShift,anchorhos, EKS 및 GKE와 같은 제품 이름
id.openshift.io	false	false	OpenShift Container Platform 클러스터에서만 사용할 수 있는 OpenShift Container Platform 외부 ID
consoleurl.openshift.io	false	true	OpenShift Container Platform 클러스터에서만 사용할 수 있는 관리 콘솔의 URL

클레임 이름	reserved	변경 가능	설명
version.openshift.io	false	true	OpenShift Container Platform 클러스터에서만 사용할 수 있는 OpenShift Container Platform 버전

이전 클레임이 삭제되거나 관리되는 클러스터에서 업데이트되면 자동으로 이전 버전으로 복원되거나 롤백됩니다.

*관리형 클러스터가 허브에 참여하면 관리 클러스터에서 생성된 **ClusterClaims**가 허브의 **ManagedCluster** 리소스 상태와 동기화됩니다. **ClusterClaims**가 있는 관리형 클러스터는 다음 예와 유사할 수 있습니다.*

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  labels:
    cloud: Amazon
    clusterID: 95f91f25-d7a2-4fc3-9237-2ef633d8451c
    installer.name: multiclusterhub
    installer.namespace: open-cluster-management
    name: cluster1
    vendor: OpenShift
  name: cluster1
spec:
  hubAcceptsClient: true
  leaseDurationSeconds: 60
status:
  allocatable:
    cpu: '15'
    memory: 65257Mi
  capacity:
    cpu: '18'
    memory: 72001Mi
  clusterClaims:
    - name: id.k8s.io
      value: cluster1
    - name: kubeversion.open-cluster-management.io
      value: v1.18.3+6c42de8
    - name: platform.open-cluster-management.io
      value: AWS
    - name: product.open-cluster-management.io
      value: OpenShift
    - name: id.openshift.io
      value: 95f91f25-d7a2-4fc3-9237-2ef633d8451c
    - name: consoleurl.openshift.io
      value: 'https://console-openshift-console.apps.xxxx.dev04.red-chesterfield.com'
    - name: version.openshift.io
```

```
    value: '4.5'
  conditions:
    - lastTransitionTime: '2020-10-26T07:08:49Z'
      message: Accepted by hub cluster admin
      reason: HubClusterAdminAccepted
      status: 'True'
      type: HubAcceptedManagedCluster
    - lastTransitionTime: '2020-10-26T07:09:18Z'
      message: Managed cluster joined
      reason: ManagedClusterJoined
      status: 'True'
      type: ManagedClusterJoined
    - lastTransitionTime: '2020-10-30T07:20:20Z'
      message: Managed cluster is available
      reason: ManagedClusterAvailable
      status: 'True'
      type: ManagedClusterConditionAvailable
  version:
    kubernetes: v1.18.3+6c42de8
```

1.16.1. 기존 ClusterClaims 나열

kubectl 명령을 사용하여 관리 클러스터에 적용되는 ClusterClaim을 나열할 수 있습니다. 이는 ClusterClaim을 오류 메시지와 비교하려는 경우에 유용합니다.

참고: resource clusterclaims.cluster.open-cluster-management.io 에 대한 목록 권한이 있는지 확인합니다.

다음 명령을 실행하여 관리형 클러스터에 있는 기존 ClusterClaim을 모두 나열합니다.

```
kubectl get clusterclaims.cluster.open-cluster-management.io
```

1.16.2. 사용자 정의 ClusterClaims 생성

관리형 클러스터에서 사용자 지정 이름을 사용하여 ClusterClaims를 생성하여 쉽게 식별할 수 있습니다. 사용자 지정 ClusterClaims는 허브 클러스터에서 ManagedCluster 리소스의 상태와 동기화됩니다. 다음 콘텐츠는 사용자 정의된 ClusterClaim 정의의 예를 보여줍니다.

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: ClusterClaim
metadata:
  name: <custom_claim_name>
spec:
  value: <custom_claim_value>
```

spec.value 필드의 최대 길이는 1024입니다. ClusterClaim을 생성하려면 resource clusterclaims.cluster.open-cluster-management.io 에 대한 생성 권한이 필요합니다.

1.17. 호스트된 컨트롤 플레인 클러스터 사용 (기술 프리뷰)

멀티 클러스터 엔진 operator 2.0을 사용하는 Kubernetes 버전 2.5용 Red Hat Advanced Cluster Management는 두 가지 다른 컨트롤 플레인 구성을 사용하여 Red Hat OpenShift Container Platform 클러스터를 배포할 수 있습니다. 독립 실행형 구성은 여러 개의 전용 가상 머신 또는 물리적 머신을 사용하여 OpenShift Container Platform 컨트롤 플레인을 호스팅합니다. 호스트된 컨트롤 플레인을 프로비저닝하여 각 컨트롤 플레인의 전용 물리적 머신 없이도 호스팅 서비스 클러스터에서 Pod로 OpenShift Container Platform 컨트롤 플레인을 프로비저닝할 수 있습니다.

참고: 이 기능은 Kubernetes용 Red Hat Advanced Cluster Management 없이 다중 클러스터 엔진 operator 2.0에서도 작동합니다.

Red Hat Advanced Cluster Management의 경우 Amazon Web Services가 기술 프리뷰로 지원됩니다. Red Hat OpenShift Container Platform 버전 4.10.7 이상의 컨트롤 플레인을 호스팅할 수 있습니다.

컨트롤 플레인은 단일 네임스페이스에 포함되어 호스팅된 컨트롤 플레인 클러스터와 연결된 Pod로 실행됩니다. OpenShift Container Platform에서 이러한 유형의 호스팅 클러스터를 프로비저닝할 때 컨트롤 플레인과 관계없이 작업자 노드를 프로비저닝합니다.

호스팅된 컨트롤 플레인 클러스터의 다음 이점을 참조하십시오.

- *전용 컨트롤 플레인 노드를 호스팅할 필요가 없어 비용 절감*
- *컨트롤 플레인과 워크로드를 분리하여 격리를 개선하고 변경이 필요할 수 있는 구성 오류를 줄입니다.*
- *컨트롤 플레인 노드 부트스트랩에 대한 요구 사항을 제거하여 클러스터 프로비저닝 시간을 크게 줄입니다.*
- *turn-key 배포 지원 또는 완전히 사용자 지정된 OpenShift Container Platform 프로비저닝*

다음 제품 문서에서 호스팅 컨트롤 플레인을 사용하는 방법에 대한 자세한 내용을 참조하십시오.

- 호스트된 컨트롤 플레인 구성
- 호스트된 컨트롤 플레인 리소스 비활성화

1.17.1. 호스트된 컨트롤 플레인 구성

호스트된 컨트롤 플레인을 구성하려면 호스팅 서비스 클러스터 및 호스트된 클러스터가 필요합니다. 기존 클러스터에 **HyperShift Operator**를 배포하면 해당 클러스터를 호스팅 서비스 클러스터로 만들고 호스트된 클러스터 생성을 시작할 수 있습니다.

호스트된 컨트롤 플레인은 기술 프리뷰 기능이므로 관련 구성 요소는 기본적으로 비활성화되어 있습니다. **multiclusterengine** 사용자 정의 리소스를 편집하여 **spec.overrides.components[?(@.name=='hypershift-preview')].enabled** 를 **true** 로 설정하여 기능을 활성화합니다.

다음 명령을 입력하여 호스팅된 컨트롤 플레인 기능이 활성화되어 있는지 확인합니다.

```
oc patch mce multiclusterengine-sample--type=merge -p '{"spec":{"overrides":{"components":
[{"name":"hypershift-preview","enabled": true}]}}}'
```

1.17.1.1. 호스팅 서비스 클러스터 구성

호스팅 서비스 클러스터로 작동하도록 기존 클러스터를 구성하여 호스팅 컨트롤 플레인을 배포할 수 있습니다. 호스팅 서비스 클러스터는 컨트롤 플레인이 호스팅되는 **OpenShift Container Platform** 클러스터이며 허브 클러스터 또는 **OpenShift Container Platform** 관리형 클러스터 중 하나일 수 있습니다.

1.17.1.1.1. 사전 요구 사항

호스팅 서비스 클러스터를 구성하려면 다음 사전 요구 사항이 있어야 합니다.

- 멀티 클러스터 엔진 **Operator**는 **Red Hat OpenShift Container Platform**에서 관리하는 하나 이상의 클러스터에 설치되었습니다. 멀티 클러스터 엔진 **Operator**는 **Red Hat Advanced Cluster Management** 버전 **2.5** 이상을 설치할 때 자동으로 설치되며 **OpenShift Container Platform OperatorHub**의 **Operator**로 **Red Hat Advanced Cluster Management** 없이 설치할 수도 있습니다.
- **Red Hat Advanced Cluster Management hub** 클러스터를 호스팅 서비스 클러스터로 설정하려면 다음 단계를 완료하여 로컬 클러스터를 호스팅 서비스 클러스터로 구성해야 합니다.

1. 다음 예와 유사한 **import-hub.yaml** 이라는 **YAML** 파일을 생성합니다.

```
apiVersion: cluster.open-cluster-management.io/v1
kind: ManagedCluster
metadata:
  labels:
    local-cluster: "true"
  name: local-cluster
spec:
  hubAcceptsClient: true
  leaseDurationSeconds: 60
```

2. 다음을 입력하여 파일을 적용합니다.

```
oc apply -f import-hub.yaml
```

자체적으로 관리하는 허브 클러스터는 클러스터 목록에서 로컬 클러스터로 지정됩니다.

1.17.1.1.2. 호스팅 서비스 클러스터 구성

멀티 클러스터 엔진 **Operator**가 설치된 클러스터에서 **OpenShift Container Platform** 관리 클러스터를 호스팅 서비스 클러스터로 활성화하려면 다음 단계를 완료합니다.

1. **AWS**에서 호스팅 클러스터를 생성하고 관리하려면 **HyperShift Operator**에 대한 **hypershift-operator-oidc-provider-s3-credentials** 라는 **OIDC S3** 인증 정보 시크릿을 생성합니다. 호스팅 서비스 클러스터로 사용되는 관리형 클러스터의 네임스페이스인 관리형 클러스터 네임스페이스에 보안을 저장합니다. **local-cluster** 를 사용한 경우 **local-cluster** 네임스페이스에 보안을 생성합니다.

   시크릿에는 세 개의 필드가 포함되어야 합니다. 버킷 필드에는 호스트 **OIDC** 검색 문서에 대한 공용 액세스 권한이 있는 **S3** 버킷이 포함되어 있습니다. **credentials** 필드는 버킷에 액세스할 수 있는 기본 프로필의 인증 정보가 포함된 파일에 대한 참조입니다. 기본적으로 **HyperShift**는 기본 프로필만 사용하여 버킷 을 작동합니다. **region** 필드는 **S3** 버킷의 리전을 지정합니다.

   보안에 대한 자세한 내용은 **HyperShift** 설명서에서 시작하기 를 참조하십시오. 다음 예제에서는 샘플 **AWS** 시크릿 템플릿을 보여줍니다.

```
oc create secret generic hypershift-operator-oidc-provider-s3-credentials --from-file=credentials=$HOME/.aws/credentials --from-literal=bucket=<s3-bucket-for-hypershift> --from-literal=region=<region> -n <hypershift-hosting-service-cluster>
```

참고: 시크릿에 대한 재해 복구 백업은 자동으로 활성화되지 않습니다. 다음 명령을 실행하여 재해 복구를 위해 **hypershift-operator-oidc-provider-s3-credentials** 보안을 백업할 수 있는 레이블을 추가합니다.

```
oc label secret hypershift-operator-oidc-provider-s3-credentials -n <hypershift-hosting-service-cluster> cluster.open-cluster-management.io/backup=""
```

2. **HyperShift** 애드온을 설치합니다.

HyperShift Operator를 호스팅하는 클러스터는 호스팅 서비스 클러스터입니다. 이 단계에서는 **hypershift-addon** 을 사용하여 관리형 클러스터에 **HyperShift Operator**를 설치합니다.

a. 다음 예와 유사한 파일을 생성하여 **ManagedClusterAddon HyperShift** 애드온을 생성합니다.

```
apiVersion: addon.open-cluster-management.io/v1alpha1
kind: ManagedClusterAddOn
metadata:
  name: hypershift-addon
  namespace: <managed-cluster-name>
spec:
  installNamespace: open-cluster-management-agent-addon
```

managed-cluster-name 을 **HyperShift Operator**를 설치하려는 관리형 클러스터의 이름으로 교체합니다. **Red Hat Advanced Cluster Management hub** 클러스터에 설치하는 경우 이 값에 **local-cluster** 를 사용합니다.

b. 다음 명령을 실행하여 파일을 적용합니다.

```
oc apply -f <filename>
```

파일 이름을 생성한 파일의 이름으로 바꿉니다.

3. 다음 명령을 실행하여 **hypershift-addon** 이 설치되었는지 확인합니다.

```
oc get managedclusteraddons -n <hypershift-hosting-service-cluster> hypershift-addon
```

애드온이 설치된 경우 출력은 다음 예와 유사합니다.

```
NAME              AVAILABLE   DEGRADED   PROGRESSING
hypershift-addon   True
```

HyperShift 애드온이 설치되고 호스팅 서비스 클러스터를 사용하여 **HyperShift** 클러스터를 관리할 수 있습니다.

1.17.1.2. 호스팅된 클러스터 배포

HyperShift Operator를 설치하고 호스팅 서비스 클러스터로 기존 클러스터를 활성화한 후 **HypershiftDeployment** 사용자 지정 리소스를 생성하여 **HyperShift** 호스팅 클러스터를 프로비저닝할 수 있습니다.

1.

콘솔 또는 파일 추가를 사용하여 클라우드 공급자 시크릿을 인증 정보로 생성합니다. **VPC**, 서브넷 및 **NAT** 게이트웨이와 같이 클러스터에 대한 인프라 리소스를 생성할 수 있는 권한이 있어야 합니다. 이 계정은 작업자가 살고 있는 게스트 클러스터의 계정에도 대응해야 합니다. 필요한 권한에 대한 자세한 내용은 **HyperShift** 설명서에서 **AWS** 인프라 및 **IAM** 리소스 만들기 를 참조하십시오.

다음 예제에서는 **AWS**의 형식을 보여줍니다.

```
apiVersion: v1
metadata:
  name: my-aws-cred
  namespace: default     # Where you create HypershiftDeployment resources
type: Opaque
kind: Secret
stringData:
  ssh-publickey:        # Value
  ssh-privatekey:       # Value
  pullSecret:           # Value, required
  baseDomain:           # Value, required
  aws_secret_access_key: # Value, required
  aws_access_key_id:     # Value, required
```

- 콘솔을 사용하여 이 시크릿을 생성하려면 탐색 메뉴의 자격 증명에 액세스하여 인증 정보 생성 단계를 따르십시오.

- 명령줄을 사용하여 보안을 생성하려면 다음 명령을 실행합니다.

```
oc create secret generic <my-secret> -n <hypershift-deployment-namespace> --from-literal=baseDomain='your.domain.com' --from-literal=aws_access_key_id='your-aws-access-key' --from-literal=aws_secret_access_key='your-aws-secret-key' --from-literal=pullSecret='your-quay-pull-secret' --from-literal=ssh-publickey='your-ssh-publickey' --from-literal=ssh-privatekey='your-ssh-privatekey'
```

-

참고: 시크릿에 대한 재해 복구 백업은 자동으로 활성화되지 않습니다. 다음 명령을 실행하여 재해 복구를 위해 보안을 백업할 수 있는 레이블을 추가합니다.

```
oc label secret <my-secret> -n <hypershift-deployment-namespace> cluster.open-cluster-management.io/backup=""
```

2.

*클라우드 공급자 시크릿 네임스페이스에서 **HypershiftDeployment** 사용자 지정 리소스 파일을 생성합니다. **HypershiftDeployment** 사용자 지정 리소스는 공급자 계정에서 인프라를 생성하고, 생성된 인프라에서 인프라 컴퓨팅 용량을 구성하고, 호스팅 컨트롤 플레인을 사용하는 **nodePool** 을 프로비저닝하며, 호스팅 서비스 클러스터에서 호스팅된 컨트롤 플레인을 생성합니다.*

a.

다음 예와 유사한 정보가 포함된 파일을 생성합니다.

```
apiVersion: cluster.open-cluster-management.io/v1alpha1
kind: HypershiftDeployment
metadata:
  name: <cluster>
  namespace: default
spec:
  hostingCluster: <hosting-service-cluster>
  hostingNamespace: clusters
  hostedClusterSpec:
    networking:
      machineCIDR: 10.0.0.0/16    # Default
      networkType: OpenShiftSDN
      podCIDR: 10.132.0.0/14      # Default
      serviceCIDR: 172.31.0.0/16  # Default
    platform:
      type: AWS
    pullSecret:
      name: <cluster>-pull-secret    # This secret is created by the controller
    release:
      image: quay.io/openshift-release-dev/ocp-release:4.10.15-x86_64  # Default
    services:
    - service: APIServer
      servicePublishingStrategy:
        type: LoadBalancer
    - service: OAuthServer
      servicePublishingStrategy:
        type: Route
    - service: Konnectivity
      servicePublishingStrategy:
        type: Route
    - service: Ignition
      servicePublishingStrategy:
        type: Route
    sshKey: {}
  nodePools:
```

```
- name: <cluster>
  spec:
    clusterName: <cluster>
    management:
      autoRepair: false
      replace:
        rollingUpdate:
          maxSurge: 1
          maxUnavailable: 0
        strategy: RollingUpdate
      upgradeType: Replace
    platform:
      aws:
        instanceType: m5.large
      type: AWS
    release:
      image: quay.io/openshift-release-dev/ocp-release:4.10.15-x86_64 # Default
    replicas: 2
infrastructure:
  cloudProvider:
    name: <my-secret>
  configure: True
  platform:
    aws:
      region: <region>
```

cluster 를 클러스터 이름으로 교체합니다.

hosting-service-cluster 를 HyperShift Operator를 호스팅하는 클러스터 이름으로 교체합니다.

my-secret 을 시크릿으로 교체하여 클라우드 공급자에 액세스합니다.

리전 을 클라우드 공급자의 리전으로 바꿉니다.

b. *다음 명령을 입력하여 파일을 적용합니다.*

```
oc apply -f <filename>
```

API의 필드 정의를 참조하여 올바르게 수행할 수 있습니다.

3. *다음 명령을 실행하여 HypershiftDeployment 상태를 확인합니다.*

```
oc get hypershiftdeployment -n default hypershift-demo -w
```

4. 호스팅 클러스터가 생성되면 자동으로 허브로 가져옵니다. **Red Hat Advanced Cluster Management** 콘솔에서 클러스터 목록을 보거나 다음 명령을 실행하여 이를 확인할 수 있습니다.

```
oc get managedcluster <hypershiftDeployment.Spec.infraID>
```

1.17.1.3. 호스팅 서비스 클러스터에 액세스

이제 클러스터에 액세스할 수 있습니다. 액세스 보안은 **hypershift-hosting-service-cluster** 네임스페이스에 저장됩니다. 이 네임스페이스는 호스팅 서비스 클러스터의 이름과 동일합니다. 다음 형식의 시크릿 이름 형식을 확인합니다.

- **kubeconfig secret: < hypershiftDeployment.Spec.Spec.hostingNamespace>-<hypershiftDeployment.Name>-admin-kubeconfig (clusters-hypershift-demo-admin-kubeconfig)**
- **kubeadmin 암호 보안: < hypershiftDeployment.Spec.hostingNamespace>-<hypershiftDeployment.Name>-kubeadmin-password (clusters-hypershift-demo-kubeadmin-password)**

1.17.2. 호스팅된 컨트롤 플레인 리소스 비활성화

호스트된 컨트롤 플레인 클러스터 기능을 비활성화하는 경우 **HyperShift** 호스팅 클러스터를 제거하고 **HyperShift Operator**를 제거해야 합니다.

1.17.2.1. HyperShift 호스트 클러스터 삭제

HyperShift 호스팅 클러스터를 삭제하려면 다음 명령 중 하나를 실행하여 **HypershiftDeployment** 리소스를 삭제합니다.

```
oc delete -f <HypershiftDeployment_yaml_file_name>
```

또는

```
oc delete hd -n <HypershiftDeployment_namespace> <HypershiftDeployment_resource_name>
```

1.17.2.2. HyperShift Operator 설치 제거

관리 또는 호스팅 서비스 클러스터에서 **HyperShift Operator**를 제거하려면 다음 명령을 실행하여 관리 클러스터에서 **hypershift-addon ManagedClusterAddon** 을 삭제합니다.

```
oc delete managedclusteraddon -n <hypershift-management-cluster> hypershift-addon
```

1.18. DISCOVERY 서비스 소개

OpenShift Cluster Manager 에서 사용할 수 있는 **OpenShift 4** 클러스터를 검색할 수 있습니다. 검색 후 관리할 클러스터를 가져올 수 있습니다. **Discovery** 서비스는 백엔드 및 콘솔 사용에 **Discover Operator**를 사용합니다.

OpenShift Cluster Manager 인증 정보가 있어야 합니다. 인증 정보를 생성해야 하는 경우 **Red Hat OpenShift Cluster Manager**에 대한 인증 정보 생성을 참조하십시오.

필수 액세스: 관리자

- 콘솔을 사용하여 **Discovery** 구성
- **CLI**를 사용하여 **Discovery** 구성

1.18.1. 콘솔을 사용하여 Discovery 구성

제품 콘솔을 사용하여 **Discovery**를 활성화합니다.

필수 액세스: 인증 정보가 생성된 네임스페이스에 액세스합니다.

1.18.1.1. 사전 요구 사항

- 인증 정보가 필요합니다. **OpenShift Cluster Manager** 에 연결할 **Red Hat OpenShift Cluster Manager**에 대한 인증 정보 생성을 참조하십시오.

1.18.1.2. Discovery 구성

클러스터를 찾도록 콘솔에서 **Discovery**를 구성합니다. 별도의 인증 정보를 사용하여 여러 **DiscoveryConfig** 리소스를 생성할 수 있습니다. 콘솔의 지침을 따르십시오.

1.18.1.3. 검색된 클러스터 보기

인증 정보를 설정하고 가져올 클러스터를 검색한 후 콘솔에서 해당 인증서를 볼 수 있습니다.

1. 클러스터 > 검색된 클러스터를클릭합니다.

2. 다음 정보를 사용하여 채워진 테이블을 확인합니다.

   - **name** 은 **OpenShift Cluster Manager**에서 지정된 표시 이름입니다. 클러스터에 표시 이름이 없으면 클러스터 콘솔 **URL**을 기반으로 생성된 이름이 표시됩니다. 콘솔 **URL**이 없거나 **OpenShift Cluster Manager**에서 수동으로 수정된 경우 클러스터 외부 **ID**가 표시됩니다.
   - **namespace** 는 인증 정보 및 검색된 클러스터를 생성한 네임스페이스입니다.
   - **type** 은 검색된 클러스터 **Red Hat OpenShift** 유형입니다.
   - 배포 버전 은 검색된 클러스터 **Red Hat OpenShift** 버전입니다.
   - 인프라 공급자는 검색된 클러스터의 클라우드 공급자입니다.
   - 마지막 활성 은 검색된 클러스터가 활성화된 마지막 시간입니다.
   - 검색된 클러스터가 생성될 때 생성됩니다.
   - 검색된 클러스터가 검색될 때 검색됩니다.

3. 테이블에서도 모든 정보를 검색할 수 있습니다. 예를 들어 특정 네임스페이스에 검색된 클러스터 만 표시하려면 해당 네임스페이스를 검색합니다.

4. 이제 클러스터 가져오기를 클릭하여 관리 클러스터를 생성할 수 있습니다. 검색된 클러스터

가져오기 를 참조하십시오.

1.18.1.4. 검색된 클러스터 가져오기

*클러스터를 검색한 후 콘솔의 **Discovered** 클러스터 탭에 표시되는 클러스터를 가져올 수 있습니다.*

1.18.1.5. 사전 요구 사항

***Discovery**를 구성하는 데 사용된 네임스페이스에 액세스해야 합니다.*

1.18.1.6. 검색된 클러스터 가져오기

1. *기존 클러스터 페이지로 이동하여 **Discovered** 클러스터 탭을 클릭합니다.*
2. ***Discovered** 클러스터 표에서 가져올 클러스터를 찾습니다.*
3. *옵션 메뉴에서 클러스터 가져오기 를 선택합니다.*
4. *검색된 클러스터의 경우 문서를 사용하여 수동으로 가져오거나 클러스터 가져오기를 자동으로 선택할 수 있습니다.*
5. *인증 정보 또는 **Kubeconfig** 파일을 사용하여 자동으로 가져오려면 콘텐츠를 복사하여 붙여넣습니다.*
6. ***Import** 를 클릭합니다.*

1.18.2. CLI를 사용하여 Discovery 활성화

***CLI**를 사용하여 검색을 활성화하여 **Red Hat OpenShift Cluster Manager**에서 사용할 수 있는 클러스터를 찾습니다.*

필수 액세스: 관리자

1.18.2.1. 사전 요구 사항

- Red Hat OpenShift Cluster Manager에 연결할 자격 증명을 만듭니다.

1.18.2.2. Discovery 설정 및 프로세스

참고: DiscoveryConfig 의 이름은 discovery 여야 하며 선택한 인증 정보와 동일한 네임스페이스에 생성해야 합니다. 다음 DiscoveryConfig 샘플을 참조하십시오.

```
apiVersion: discovery.open-cluster-management.io/v1
kind: DiscoveryConfig
metadata:
  name: discovery
  namespace: <NAMESPACE_NAME>
spec:
  credential: <SECRET_NAME>
  filters:
    lastActive: 7
    openshiftVersions:
    - "4.10"
    - "4.9"
    - "4.8"
```

1. SECRET_NAME 을 이전에 설정한 자격 증명으로 바꿉니다.

2. NAMESPACE_NAME 을 SECRET_NAME 의 네임스페이스로 바꿉니다.

3. 검색할 클러스터의 마지막 활동(일) 이후의 최대 시간을 입력합니다. 예를 들어 lastActive: 7 을 사용하면 지난 7일 동안 활성화된 클러스터가 검색됩니다.

4. 검색할 Red Hat OpenShift 클러스터 버전을 문자열 목록으로 입력합니다. 참고: openshiftVersions 목록의 모든 항목은 OpenShift 주 및 부 버전을 지정합니다. 예를 들어 "4.9" 를 지정하면 OpenShift 버전 4.9 의 모든 패치 릴리스 (예: 4.9.1,4.9.2)가 포함됩니다.

1.18.2.3. 검색된 클러스터 보기

oc get discoveredclusters -n <namespace >를 실행하여 검색된 클러스터를 확인합니다. 여기서 namespace 는 검색 인증 정보가 존재하는 네임스페이스입니다.

1.18.2.3.1. DiscoveredClusters

오브젝트는 Discovery 컨트롤러에서 생성합니다. 이러한 DiscoveredClusters 는 DiscoveryConfig discoveredclusters.discovery.open-cluster-management.io API에 지정된 필터 및 인증 정보를 사용하여 OpenShift Cluster Manager에 있는 클러스터를 나타냅니다. name 값은 클러스터 외부 ID입니다.

```
apiVersion: discovery.open-cluster-management.io/v1
kind: DiscoveredCluster
metadata:
  name: fd51aafa-95a8-41f7-a992-6fb95eed3c8e
  namespace: <NAMESPACE_NAME>
spec:
  activity_timestamp: "2021-04-19T21:06:14Z"
  cloudProvider: vsphere
  console: https://console-openshift-console.apps.qe1-vmware-pkt.dev02.red-chesterfield.com
  creation_timestamp: "2021-04-19T16:29:53Z"
  credential:
    apiVersion: v1
    kind: Secret
    name: <SECRET_NAME>
    namespace: <NAMESPACE_NAME>
  display_name: qe1-vmware-pkt.dev02.red-chesterfield.com
  name: fd51aafa-95a8-41f7-a992-6fb95eed3c8e
  openshiftVersion: 4.10
  status: Stale
```

1.19. 클러스터 업그레이드

Kubernetes용 Red Hat Advanced Cluster Management를 사용하여 관리할 Red Hat OpenShift Container Platform 클러스터를 생성한 후 Kubernetes 콘솔의 Red Hat Advanced Cluster Management를 사용하여 해당 클러스터를 관리 클러스터가 사용하는 버전 채널에서 사용 가능한 최신 마이너 버전으로 업그레이드할 수 있습니다.

연결된 환경에서 업데이트는 Red Hat Advanced Cluster Management 콘솔에서 업그레이드해야 하는 각 클러스터에 대해 제공되는 알림으로 자동 식별됩니다.

중요: 연결이 끊긴 환경에서 클러스터를 업그레이드하는 프로세스에는 필요한 릴리스 이미지를 구성하고 미러링하는 몇 가지 추가 단계가 필요합니다. Red Hat OpenShift Update Service용 Operator를 사용하여 업그레이드를 식별합니다. 연결이 끊긴 환경에 있는 경우 필요한 단계에 대한 연결이 끊긴 클러스터 업그레이드를 참조하십시오.

참고:

주요 버전으로 업그레이드하려면 해당 버전으로 업그레이드하기 위한 모든 사전 요구 사항을 충족하는지 확인해야 합니다. 콘솔을 사용하여 클러스터를 업그레이드하기 전에 관리형 클러스터에서 버전 채널을

업데이트해야 합니다.

관리형 클러스터에서 버전 채널을 업데이트한 후 **Red Hat Advanced Cluster Management for Kubernetes** 콘솔에 업그레이드에 사용할 수 있는 최신 버전이 표시됩니다.

이 업그레이드 방법은 **Ready** 상태인 **OpenShift Container Platform** 관리 클러스터에서만 작동합니다.

중요: **Kubernetes** 콘솔용 **Red Hat Advanced Cluster Management**를 사용하여 **Red Hat OpenShift Kubernetes Service** 관리 클러스터 또는 **OpenShift Container Platform** 관리 클러스터를 **Red Hat OpenShift Dedicated**에서 업그레이드할 수 없습니다.

연결된 환경에서 클러스터를 업그레이드하려면 다음 단계를 완료합니다.

1. 탐색 메뉴에서 인프라 **&gt ;** 클러스터로 이동합니다. 업그레이드를 사용할 수 있는 경우 배포 버전 열에 표시됩니다.
2. 업그레이드하려는 **Ready** 상태에서 클러스터를 선택합니다. 콘솔을 사용하여 업그레이드하려면 클러스터가 **OpenShift Container Platform** 클러스터여야 합니다.
3. 업그레이드를 선택합니다.
4. 각 클러스터의 새 버전을 선택합니다.
5. 업그레이드를 선택합니다.

클러스터 업그레이드가 실패하면 **Operator**는 일반적으로 업그레이드를 몇 번 재시도하고 중지한 후 실패한 구성 요소의 상태를 보고합니다. 경우에 따라 업그레이드 프로세스가 프로세스를 완료하기 위한 시도로 계속 순환됩니다. 실패한 업그레이드 후 클러스터를 이전 버전으로 롤백하는 것은 지원되지 않습니다. 클러스터 업그레이드가 실패하는 경우 **Red Hat** 지원에 문의하십시오.

1.19.1. 채널 선택

Red Hat Advanced Cluster Management 콘솔을 사용하여 **OpenShift Container Platform** 버전 **4.6**

*이상에서 클러스터 업그레이드 채널을 선택할 수 있습니다. 채널을 선택하면 에라타 버전 (**4.8.1 > 4.8.2 > 4.8.3** 등)과 릴리스 버전 (**4.8 > 4.9** 등) 모두에서 사용할 수 있는 클러스터 업그레이드를 자동으로 상기시킵니다.*

클러스터 채널을 선택하려면 다음 단계를 완료합니다.

1. ***Red Hat Advanced Cluster Management** 탐색에서 인프라 **>** 클러스터를 선택합니다.*
2. *클러스터 세부 정보 페이지를 보려면 변경할 클러스터 이름을 선택합니다. 클러스터에 다른 채널을 사용할 수 있는 경우 채널 필드에 편집 아이콘이 표시됩니다.*
3. *편집 아이콘을 클릭하여 필드의 설정을 수정합니다.*
4. *새 채널 필드에서 채널을 선택합니다.*

클러스터의 클러스터 세부 정보 페이지에서 사용 가능한 채널 업데이트에 대한 알림을 확인할 수 있습니다.

1.19.2. 연결이 끊긴 클러스터 업그레이드

***Red Hat OpenShift Update Service**를 **Kubernetes**용 **Red Hat Advanced Cluster Management for Kubernetes**와 함께 사용하여 연결이 끊긴 환경에서 클러스터를 업그레이드할 수 있습니다.*

*보안 문제로 인해 클러스터가 인터넷에 직접 연결되지 않는 경우도 있습니다. 이로 인해 업그레이드가 사용 가능한 시기와 해당 업그레이드를 처리하는 방법을 알기가 어렵습니다. **OpenShift Update Service**를 구성하면 도움이 될 수 있습니다.*

***OpenShift Update Service**는 연결이 끊긴 환경에서 사용 가능한 관리 클러스터 버전을 모니터링하고 연결이 끊긴 환경에서 클러스터를 업그레이드할 수 있도록 하는 별도의 **Operator** 및 피연산자입니다. **OpenShift Update Service**가 구성된 후 다음 작업을 수행할 수 있습니다.*

1. *연결이 끊긴 클러스터에 업그레이드를 사용할 수 있는 경우를 모니터링합니다.*
2. *그래프 데이터 파일을 사용하여 업그레이드를 위해 로컬 사이트에 미러링된 업데이트를 식별*

합니다.

3. ***Red Hat Advanced Cluster Management** 콘솔을 사용하여 클러스터에 업그레이드를 사용할 수 있음을 알립니다.*

- *사전 요구 사항*
- *연결이 끊긴 미러 레지스트리 준비*
- ***OpenShift Update Service**용 **Operator** 배포*
- *그래프 데이터 **init** 컨테이너 빌드*
- *미러링된 레지스트리에 대한 인증서 구성*
- ***OpenShift Update Service** 인스턴스 배포*
- *기본 레지스트리를 덮어쓰는 정책을 배포합니다(선택 사항)*
- *연결이 끊긴 카탈로그 소스를 배포하는 정책을 배포합니다.*
- *관리 클러스터 매개변수 변경에 대한 정책 배포*
- *사용 가능한 업그레이드 보기*
- *채널 선택*
- *클러스터 업그레이드*

1.19.2.1. 사전 요구 사항

OpenShift Update Service를 사용하여 연결이 끊긴 클러스터를 업그레이드하기 전에 다음 사전 요구 사항이 있어야 합니다.

- 제한된 **OLM**이 구성된 **Red Hat OpenShift Container Platform** 버전 **4.6** 이상에서 실행 중인 배포된 **Red Hat Advanced Cluster Management hub** 클러스터입니다. 제한된 **OLM** 을 구성하는 방법에 대한 자세한 내용은 제한된 네트워크에서 **Operator Lifecycle Manager** 사용을 참조하십시오.

  팁: 제한된 **OLM**을 구성할 때 카탈로그 소스 이미지를 기록해 둡니다.

- **Red Hat Advanced Cluster Management hub** 클러스터에서 관리하는 **OpenShift Container Platform** 클러스터

- 클러스터 이미지를 미러링할 수 있는 로컬 저장소에 대한 인증 정보에 액세스합니다. 이 리포지토리를 생성하는 방법에 대한 자세한 내용은 연결 해제 설치 미러링 을 참조하십시오.

  참고: 업그레이드한 현재 클러스터 버전의 이미지를 미러링된 이미지 중 하나로 항상 사용할 수 있어야 합니다. 업그레이드가 실패하면 클러스터는 업그레이드를 시도할 때 클러스터 버전으로 다시 돌아갑니다.

1.19.2.2. 연결이 끊긴 미러 레지스트리 준비

업그레이드하려는 이미지와 현재 이미지를 로컬 미러 레지스트리로 미러링해야 합니다. 이미지를 미러링하려면 다음 단계를 완료합니다.

1. 다음 예와 유사한 콘텐츠가 포함된 스크립트 파일을 생성합니다.

```
UPSTREAM_REGISTRY=quay.io
PRODUCT_REPO=openshift-release-dev
RELEASE_NAME=ocp-release
OCP_RELEASE=4.5.2-x86_64
LOCAL_REGISTRY=$(hostname):5000
LOCAL_SECRET_JSON=/path/to/pull/secret

oc adm -a ${LOCAL_SECRET_JSON} release mirror \
--
from=${UPSTREAM_REGISTRY}/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE} \
--to=${LOCAL_REGISTRY}/ocp4 \
--to-release-image=${LOCAL_REGISTRY}/ocp4/release:${OCP_RELEASE}
```

/path/to/pull/secret 을 OpenShift Container Platform 풀 시크릿의 경로로 교체합니다.

2. 스크립트를 실행하여 이미지를 미러링하고, 설정을 구성하고, 릴리스 콘텐츠와 릴리스 이미지를 분리합니다.

팁: ImageContentSourcePolicy 를 생성할 때 이 스크립트의 마지막 줄의 출력을 사용할 수 있습니다.

1.19.2.3. OpenShift Update Service용 Operator 배포

OpenShift Container Platform 환경에서 OpenShift Update Service용 Operator를 배포하려면 다음 단계를 완료합니다.

1. hub 클러스터에서 OpenShift Container Platform Operator 허브에 액세스합니다.

2. Red Hat OpenShift Update Service Operator 를 선택하여 Operator를 배포합니다. 필요한 경우 기본값을 업데이트합니다. Operator를 배포하면 openshift-cincinnati 라는 새 프로젝트가 생성됩니다.

3. Operator 설치가 완료될 때까지 기다립니다.

팁: OpenShift Container Platform 명령줄에 oc get pods 명령을 입력하여 설치 상태를 확인할 수 있습니다. Operator가 running 상태인지 확인합니다.

1.19.2.4. 그래프 데이터 init 컨테이너 빌드

OpenShift Update Service는 그래프 데이터 정보를 사용하여 사용 가능한 업그레이드를 결정합니다. 연결된 환경에서 OpenShift Update Service는 Cincinnati 그래프 데이터 GitHub 리포지토리에서 직접 사용 가능한 업그레이드를 위한 그래프 데이터 정보를 가져옵니다. 연결이 끊긴 환경을 구성하므로 init 컨테이너 를 사용하여 로컬 리포지토리에서 그래프 데이터를 사용할 수 있도록 해야 합니다. 그래프 데이터 init 컨테이너 를 생성하려면 다음 단계를 완료합니다.

1. 다음 명령을 입력하여 그래프 데이터 Git 리포지토리를 복제합니다.

```
git clone https://github.com/openshift/cincinnati-graph-data
```

2.

그래프 데이터 **init** 에 대한 정보가 포함된 파일을 만듭니다. 이 샘플 **Dockerfile** 은 **cincinnati-operator GitHub** 리포지토리에서 찾을 수 있습니다. 파일의 내용은 다음 샘플에 표시됩니다.

```
FROM registry.access.redhat.com/ubi8/ubi:8.1

RUN curl -L -o cincinnati-graph-data.tar.gz https://github.com/openshift/cincinnati-graph-data/archive/master.tar.gz

RUN mkdir -p /var/lib/cincinnati/graph-data/

CMD exec /bin/bash -c "tar xvzf cincinnati-graph-data.tar.gz -C /var/lib/cincinnati/graph-data/ --strip-components=1"
```

이 예제에서는 다음을 수행합니다.

- **FROM** 값은 **OpenShift Update Service**가 이미지를 찾는 외부 레지스트리입니다.
- **RUN** 명령은 디렉터리를 생성하고 업그레이드 파일을 패키징합니다.
- **CMD** 명령은 패키지 파일을 로컬 리포지토리에 복사하고 업그레이드할 파일을 추출합니다.

3. 다음 명령을 실행하여 그래프 데이터 **init** 컨테이너 를 빌드합니다.

```
podman build -f <path_to_Dockerfile> -t ${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-container:latest
podman push ${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-graph-data-container:latest --authfile=/path/to/pull_secret.json
```

path_to_Dockerfile 을 이전 단계에서 만든 파일의 경로로 바꿉니다.

${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-data-container 를 로컬 그래프 데이터 **init** 컨테이너의 경로로 교체합니다.

/path/to/pull_secret 을 가져오기 보안 파일의 경로로 바꿉니다.

참고**: podman** 이 설치되지 않은 경우 명령에서 **podman** 을 **docker** 로 교체할 수도 있습니

다.

1.19.2.5. 미러링된 레지스트리에 대한 인증서 구성

미러링된 OpenShift Container Platform 릴리스 이미지를 저장하기 위해 보안 외부 컨테이너 레지스트리를 사용하는 경우 OpenShift Update Service는 업그레이드 그래프를 빌드하기 위해 이 레지스트리에 액세스해야 합니다. OpenShift Update Service Pod에서 작동하도록 CA 인증서를 구성하려면 다음 단계를 완료합니다.

1. image.config.openshift.io 에 있는 OpenShift Container Platform 외부 레지스트리 API를 찾습니다. 외부 레지스트리 CA 인증서가 저장되는 위치입니다.

   자세한 내용은 OpenShift Container Platform 설명서에서 이미지 레지스트리 액세스를 위한 추가 신뢰 저장소 구성을 참조하십시오.

2. openshift-config 네임스페이스에 ConfigMap을 생성합니다.

3. 키 updateservice-registry 아래에 CA 인증서를 추가합니다. OpenShift Update Service는 이 설정을 사용하여 인증서를 찾습니다.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: trusted-ca
data:
  updateservice-registry: |
    -----BEGIN CERTIFICATE-----
    ...
    -----END CERTIFICATE-----
```

4. image.config.openshift.io API에서 클러스터 리소스를 편집하여 additionalTrustedCA 필드를 생성한 ConfigMap의 이름으로 설정합니다.

```
oc patch image.config.openshift.io cluster -p '{"spec":{"additionalTrustedCA":
{"name":"trusted-ca"}}}' --type merge
```

   trusted-ca 를 새 ConfigMap의 경로로 교체합니다.

OpenShift Update Service Operator는 openshift-config 네임스페이스에서 생성한 image.config.openshift.io API 및 변경 사항을 확인한 다음 CA 인증서가 변경된 경우 배포를 다시 시작

합니다.

1.19.2.6. OpenShift Update Service 인스턴스 배포

hub 클러스터에 OpenShift Update Service 인스턴스 배포를 완료하면 이 인스턴스는 클러스터 업그레이드용 이미지가 미러링되어 연결이 끊긴 관리 클러스터에서 사용할 수 있게 됩니다. 인스턴스를 배포하려면 다음 단계를 완료합니다.

1. *openshift-cincinnati 인 Operator의 기본 네임스페이스를 사용하지 않으려면 OpenShift Update Service 인스턴스의 네임스페이스를 생성합니다.*
   a. *OpenShift Container Platform 허브 클러스터 콘솔 탐색 메뉴에서 관리 > 네임스페이스를 선택합니다.*
   b. *네임스페이스 생성을 선택합니다.*
   c. *네임스페이스의 이름과 네임스페이스의 기타 정보를 추가합니다.*
   d. *생성을 선택하여 네임스페이스를 생성합니다.*
2. *OpenShift Container Platform 콘솔의 설치된 Operator 섹션에서 Red Hat OpenShift Update Service Operator 를 선택합니다.*
3. *메뉴에서 Create Instance 를 선택합니다.*
4. *OpenShift Update Service 인스턴스에서 콘텐츠를 붙여넣습니다. YAML 인스턴스는 다음 매니페스트와 유사합니다.*

```
apiVersion: cincinnati.openshift.io/v1beta2
kind: Cincinnati
metadata:
  name: openshift-update-service-instance
  namespace: openshift-cincinnati
spec:
  registry: <registry_host_name>:<port>
```

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

spec.registry 값을 이미지의 연결이 끊긴 로컬 레지스트리 경로로 바꿉니다.

spec.graphDataImage 값을 그래프 데이터 init 컨테이너의 경로로 교체합니다. 팁: 그래프 데이터 init 컨테이너를 푸시하기 위해 podman push 명령을 실행할 때 사용한 값과 동일합니다.

5. 만들기 를 선택하여 인스턴스를 만듭니다.

6. hub 클러스터 CLI에서 oc get pods 명령을 입력하여 인스턴스 생성 상태를 확인합니다. 이 작업은 다소 시간이 걸릴 수 있지만 명령 결과에 인스턴스 및 Operator가 실행 중임을 표시하면 프로세스가 완료됩니다.

1.19.2.7. 기본 레지스트리를 덮어쓰는 정책을 배포합니다(선택 사항)

참고: 이 섹션의 단계는 릴리스를 미러링된 레지스트리로 미러링한 경우에만 적용됩니다.

OpenShift Container Platform에는 업그레이드 패키지를 찾을 위치를 지정하는 기본 이미지 레지스트리 값이 있습니다. 연결이 끊긴 환경에서는 해당 값을 릴리스 이미지를 미러링한 로컬 이미지 레지스트리의 경로로 교체하는 정책을 생성할 수 있습니다.

이 단계에서 정책의 이름은 ImageContentSourcePolicy 입니다. 정책을 생성하려면 다음 단계를 완료합니다.

1. hub 클러스터의 OpenShift Container Platform 환경에 로그인합니다.

2. OpenShift Container Platform 탐색에서 Administration > Custom Resource Definitions 를 선택합니다.

3. Instances 탭을 선택합니다.

4. 연결 해제된 OLM을 설정하여 콘텐츠를 볼 때 생성한 ImageContentSourcePolicy 의 이름을 선택합니다.

5. *YAML 탭을 선택하여 YAML 형식으로 된 콘텐츠를 확인합니다.*

6. *ImageContentSourcePolicy의 전체 콘텐츠를 복사합니다.*

7. *Red Hat Advanced Cluster Management 콘솔에서 관리 &gt ; 정책 만들기를 선택합니다.*

8. *YAML 스위치를 On 으로 설정하여 정책의 YAML 버전을 확인합니다.*

9. *YAML 코드의 모든 콘텐츠를 삭제합니다.*

10. *다음 YAML 콘텐츠를 창에 붙여넣어 사용자 지정 정책을 생성합니다.*

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards: ""
    policy.open-cluster-management.io/categories: ""
    policy.open-cluster-management.io/controls: ""
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
          namespace: default
        spec:
          remediationAction: inform
          severity: low
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: operator.openshift.io/v1alpha1
                kind: ImageContentSourcePolicy
                metadata:
                  name: <your-local-mirror-name>
                spec:
```

```
            repositoryDigestMirrors:
             - mirrors:
                - <your-registry>
               source: registry.redhat.io
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      []  # selects all clusters if not specified
```

11. *템플릿의* **objectDefinition** *섹션 내의 콘텐츠를 콘텐츠로 교체하고* **ImageContentSourcePolicy** *의 설정을 추가합니다.* **path-to-local-mirror** *를 로컬 미러 저장소의 경로로 바꿉니다.*

    팁: **oc adm release mirror** *명령을 입력하여 로컬 미러의 경로를 찾을 수 있습니다.*

12. *지원되는 경우* **Enforce** *상자를 선택합니다.*

13. *생성을 선택하여 정책을 생성합니다.*

1.19.2.8. 연결이 끊긴 카탈로그 소스를 배포하는 정책을 배포합니다.

Catalogsource *정책을 관리형 클러스터로 푸시하여 연결된 위치에서 연결이 끊긴 로컬 레지스트리로 기본 위치를 변경합니다.*

1. *Red Hat Advanced Cluster Management 콘솔에서 Infrastructure > Clusters 를 선택합니다.*

2. *클러스터 목록에서 정책을 수신하도록 관리형 클러스터를 찾습니다.*

3. *관리 클러스터의 name 레이블 값을 기록해 둡니다. 레이블 형식은 name=managed-cluster-name 입니다. 이 값은 정책을 푸시할 때 사용됩니다.*

4. *Red Hat Advanced Cluster Management 콘솔 메뉴에서 Governance > Create policy 를 선택합니다.*

5. *YAML 스위치를 On 으로 설정하여 정책의 YAML 버전을 확인합니다.*

6. *YAML 코드의 모든 콘텐츠를 삭제합니다.*

7. *다음 YAML 콘텐츠를 창에 붙여넣어 사용자 지정 정책을 생성합니다.*

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
        spec:
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: v1
                kind: Pod
                metadata:
                  name: sample-nginx-pod
```

```
            namespace: default
          status:
            phase: Running
      remediationAction: inform
      severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-pod
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      [] # selects all clusters if not specified
```

8. *정책에 다음 내용을 추가합니다.*

```
apiVersion: config.openshift.io/vi
kind: OperatorHub
metadata:
 name: cluster
spec:
 disableAllDefaultSources: true
```

9. *다음 콘텐츠를 추가합니다.*

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: my-operator-catalog
  namespace: openshift-marketplace
spec:
```

```
sourceType: grpc
image: <registry_host_name>:<port>/olm/redhat-operators:v1
displayName: My Operator Catalog
publisher: grpc
```

spec.image 값을 로컬 제한된 카탈로그 소스 이미지의 경로로 교체합니다.

10. *Red Hat Advanced Cluster Management 콘솔 탐색에서 인프라 > 클러스터를 선택하여 관리형 클러스터의 상태를 확인합니다. 정책이 적용되면 클러스터 상태가 준비됩니다.*

1.19.2.9. 관리 클러스터 매개변수 변경에 대한 정책 배포

ClusterVersion 정책을 관리형 클러스터로 푸시하여 업그레이드를 검색하는 기본 위치를 변경합니다.

1. *관리형 클러스터에서 다음 명령을 입력하여 ClusterVersion 업스트림 매개변수가 현재 기본 공용 OpenShift Update Service 피연산자인지 확인합니다.*

```
oc get clusterversion -o yaml
```

반환된 내용은 다음 내용과 유사합니다.

```
apiVersion: v1
items:
- apiVersion: config.openshift.io/v1
  kind: ClusterVersion
[..]
  spec:
    channel: stable-4.4
    upstream: https://api.openshift.com/api/upgrades_info/v1/graph
```

2. *hub 클러스터에서 oc get routes 명령을 입력하여 OpenShift Update Service 피연산자에 대한 경로 URL을 확인합니다.*

팁: 이후 단계에서 이 값을 기록해 두십시오.

3. *hub 클러스터에서 Red Hat Advanced Cluster Management 콘솔 메뉴에서 거버넌스 > 정책 만들기를 선택합니다.*

4. *YAML 스위치를 On 으로 설정하여 정책의 YAML 버전을 확인합니다.*

5. *YAML 코드의 모든 콘텐츠를 삭제합니다.*

6. *다음 YAML 콘텐츠를 창에 붙여넣어 사용자 지정 정책을 생성합니다.*

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-pod
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards:
    policy.open-cluster-management.io/categories:
    policy.open-cluster-management.io/controls:
spec:
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-pod-sample-nginx-pod
        spec:
          object-templates:
            - complianceType: musthave
              objectDefinition:
                apiVersion: v1
                kind: Pod
                metadata:
                  name: sample-nginx-pod
                  namespace: default
                status:
                  phase: Running
          remediationAction: inform
          severity: low
  remediationAction: enforce
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-pod
  namespace: default
placementRef:
  name: placement-policy-pod
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-pod
  kind: Policy
```

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

7. *policy 섹션의 policy.spec 에 다음 내용을 추가합니다.*

```
apiVersion: config.openshift.io/v1
  kind: ClusterVersion
  metadata:
    name: version
  spec:
    channel: stable-4.4
    upstream: https://example-cincinnati-policy-engine-uri/api/upgrades_info/v1/graph
```

spec.upstream 값을 hub 클러스터 OpenShift Update Service 피연산자의 경로로 바꿉니다.

팁: 다음 단계를 완료하여 피연산자의 경로를 결정할 수 있습니다.

a. *hub 클러스터에서 oc get routes -A 명령을 실행합니다.*

b. *cincinnati. + 피연산자의 경로는 HOST/PORT 필드의 값입니다.*

8. *관리형 클러스터 CLI에서 ClusterVersion 의 업스트림 매개변수가 다음과 같이 입력하여 로컬 허브 클러스터 OpenShift Update Service URL로 업데이트되었는지 확인합니다.*

```
oc get clusterversion -o yaml
```

결과가 다음 내용과 유사한지 확인합니다.

```
apiVersion: v1
items:
- apiVersion: config.openshift.io/v1
  kind: ClusterVersion
[..]
  spec:
    channel: stable-4.4
    upstream: https://<hub-cincinnati-uri>/api/upgrades_info/v1/graph
```

1.19.2.10. 사용 가능한 업그레이드 보기

다음 단계를 완료하여 관리 클러스터에 사용 가능한 업그레이드 목록을 볼 수 있습니다.

1. **Red Hat Advanced Cluster Management** 콘솔에 로그인합니다.
2. 탐색 메뉴에서 인프라 **>** 클러스터를 선택합니다.
3. **Ready** 상태에 있는 클러스터를 선택합니다.
4. 작업 메뉴에서 클러스터 업그레이드를 선택합니다.
5. 선택적 업그레이드 경로를 사용할 수 있는지 확인합니다.

참고: 현재 버전이 로컬 이미지 저장소에 미러링되지 않은 경우 사용 가능한 업그레이드 버전이 표시되지 않습니다.

1.19.2.11. 채널 선택

Red Hat Advanced Cluster Management 콘솔을 사용하여 **OpenShift Container Platform** 버전 **4.6** 이상에서 클러스터 업그레이드 채널을 선택할 수 있습니다. 이러한 버전은 미러 레지스트리에서 사용할 수 있어야 합니다. 업그레이드 채널을 지정하려면 채널 선택 단계를 완료합니다.

1.19.2.12. 클러스터 업그레이드

연결이 끊긴 레지스트리를 구성한 후 **Red Hat Advanced Cluster Management** 및 **OpenShift Update Service**는 연결이 끊긴 레지스트리를 사용하여 업그레이드를 사용할 수 있는지 확인합니다. 사용 가능한 업그레이드가 표시되지 않는 경우 현재 클러스터 수준의 릴리스 이미지와 하나 이상의 이후 수준

이 로컬 저장소에 미러링되어 있는지 확인합니다. 현재 버전의 클러스터의 릴리스 이미지를 사용할 수 없는 경우 업그레이드를 사용할 수 없습니다.

업그레이드하려면 다음 단계를 완료합니다.

1. ***Red Hat Advanced Cluster Management** 콘솔에서 **Infrastructure > Clusters** 를 선택합니다.*
2. *사용 가능한 업그레이드가 있는지 확인할 클러스터를 찾습니다.*
3. *사용 가능한 업그레이드가 있는 경우 클러스터의 배포 버전 열에 사용 가능한 업그레이드가 있음을 나타냅니다.*
4. *클러스터의 옵션 메뉴를 선택하고 클러스터 업그레이드를 선택합니다.*
5. *업그레이드 대상 버전을 선택하고 업그레이드를 선택합니다.*

관리 클러스터는 선택한 버전으로 업데이트됩니다.

*클러스터 업그레이드가 실패하면 **Operator**는 일반적으로 업그레이드를 몇 번 재시도하고 중지한 후 실패한 구성 요소의 상태를 보고합니다. 경우에 따라 업그레이드 프로세스가 프로세스를 완료하기 위한 시도로 계속 순환됩니다. 실패한 업그레이드 후 클러스터를 이전 버전으로 롤백하는 것은 지원되지 않습니다. 클러스터 업그레이드가 실패하는 경우 **Red Hat** 지원에 문의하십시오.*

1.20. 관리에서 클러스터 제거

***Kubernetes**용 **Red Hat Advanced Cluster Management**를 사용하여 생성된 관리에서 **OpenShift Container Platform** 클러스터를 제거하면 이를 분리 하거나 삭제 할 수 있습니다. 클러스터를 분리하면 관리에서 제거되지만 완전히 삭제되지는 않습니다. 관리하려는 경우 다시 가져올 수 있습니다. 이는 클러스터가 **Ready** 상태인 경우에만 옵션입니다.*

다음 절차에서는 다음 상황 중 하나에서 클러스터를 제거합니다.

- *이미 클러스터를 삭제하고 **Red Hat Advanced Cluster Management**에서 삭제된 클러스터*

를 삭제하려고 합니다.

- *관리에서 클러스터를 제거하려고 하지만 클러스터를 삭제하지 않았습니다.*

중요:

- *클러스터를 삭제하면 관리에서 클러스터가 제거되고 클러스터의 구성 요소가 삭제됩니다.*
- *관리형 클러스터를 분리하면 관련 네임스페이스가 자동으로 삭제됩니다. 이 네임스페이스에 사용자 정의 리소스를 배치하지 마십시오.*
  - *콘솔을 사용하여 클러스터 제거*
  - *명령줄을 사용하여 클러스터 제거*
  - *클러스터를 제거한 후 나머지 리소스 제거*
  - *클러스터를 제거한 후 **etcd** 데이터베이스 조각 모음*

1.20.1. 콘솔을 사용하여 클러스터 제거

*탐색 메뉴에서 **Infrastructure > Clusters** 로 이동하여 관리에서 삭제하려는 클러스터 옆에 있는 옵션 메뉴 옆에 있는 클러스터의 **Destroy cluster** 또는 **Detach** 클러스터를 선택합니다.*

*+ 팁: 분리 또는 제거하려는 클러스터의 확인란을 선택하고 **Detach** 또는 **Destroy** 를 선택하여 여러 클러스터를 분리하거나 제거할 수 있습니다.*

*참고: 로컬 클러스터라고 하는 동안 **hub** 클러스터를 분리하려고 하면 **disableHubSelfManagement** 의 기본 설정이 **false** 인지 확인하십시오. 이 설정을 사용하면 **hub** 클러스터가 분리될 때 자체적으로 다시 가져오고 자체적으로 관리되며 **MultiClusterHub** 컨트롤러를 조정합니다. 허브 클러스터가 분리 프로세스를 완료하고 다시 가져오는 데 시간이 걸릴 수 있습니다.*

*프로세스가 완료될 때까지 기다리지 않고 **hub** 클러스터를 다시 가져오려면 다음 명령을 입력하여*

multiclusterhub-operator Pod를 다시 시작하고 더 빨리 다시 가져올 수 있습니다.

```
oc delete po -n open-cluster-management `oc get pod -n open-cluster-management | grep multiclusterhub-operator| cut -d' ' -f1`
```

온라인 상태에서 설치 시 설명된 대로 **disableHubSelfManagement** 값을 **true** 로 변경하여 **hub** 클러스터의 값을 자동으로 가져오지 않도록 변경할 수 있습니다.

1.20.2. 명령줄을 사용하여 클러스터 제거

hub 클러스터의 명령줄을 사용하여 관리형 클러스터를 분리하려면 다음 명령을 실행합니다.

```
oc delete managedcluster $CLUSTER_NAME
```

분리 후 관리형 클러스터를 삭제하려면 다음 명령을 실행합니다.

```
oc delete clusterdeployment <CLUSTER_NAME> -n $CLUSTER_NAME
```

참고: **local-cluster** 라는 **hub** 클러스터를 분리하려고 하면 **disableHubSelfManagement** 의 기본 설정은 **false** 입니다. 이 설정을 사용하면 허브 클러스터가 분리될 때 자체적으로 다시 가져오고 자체적으로 관리되고 **MultiClusterHub** 컨트롤러를 조정합니다. 허브 클러스터가 분리 프로세스를 완료하고 다시 가져오는 데 시간이 걸릴 수 있습니다. 프로세스가 완료될 때까지 기다리지 않고 **hub** 클러스터를 다시 가져오려면 다음 명령을 입력하여 **multiclusterhub-operator Pod**를 재시작하고 더 빨리 다시 가져올 수 있습니다.

```
oc delete po -n open-cluster-management `oc get pod -n open-cluster-management | grep multiclusterhub-operator| cut -d' ' -f1`
```

온라인 상태에서 설치 시 설명된 대로 **disableHubSelfManagement** 값을 **true** 로 변경하여 **hub** 클러스터의 값을 자동으로 가져오지 않도록 변경할 수 있습니다.

1.20.3. 클러스터를 제거한 후 나머지 리소스 제거

제거한 관리형 클러스터에 나머지 리소스가 있는 경우 나머지 구성 요소를 모두 제거하는 데 필요한 추가 단계가 있습니다. 이러한 추가 단계가 필요한 경우 다음 예제를 포함합니다.

- 관리 클러스터는 완전히 생성되기 전에 분리되었으며 **klusterlet** 과 같은 구성 요소는 관리형 클러스터에 남아 있습니다.

- 관리형 클러스터를 분리하기 전에 클러스터를 관리하는 허브가 손실되거나 삭제되었으며 관리 대상 클러스터를 허브에서 분리할 수 없습니다.

- 관리 클러스터는 분리될 때 온라인 상태가 아닙니다.

이러한 상황 중 하나가 관리형 클러스터의 시도된 분리에 적용되는 경우 관리 클러스터에서 제거할 수 없는 일부 리소스가 있습니다. 관리형 클러스터를 분리하려면 다음 단계를 완료합니다.

1. **oc** 명령행 인터페이스가 구성되어 있는지 확인합니다.

2. 관리 클러스터에 **KUBECONFIG** 가 구성되어 있는지 확인합니다.

   oc get ns | grep open-cluster-management-agent 를 실행하는 경우 두 개의 네임스페이스가 표시됩니다.

   ```
   open-cluster-management-agent        Active   10m
   open-cluster-management-agent-addon   Active   10m
   ```

3. 나머지 리소스를 제거하려면 다음 명령을 실행합니다.

   ```
   oc delete namespaces open-cluster-management-agent open-cluster-management-agent-addon --wait=false
   oc get crds | grep open-cluster-management.io | awk '{print $1}' | xargs oc delete crds --wait=false
   oc get crds | grep open-cluster-management.io | awk '{print $1}' | xargs oc patch crds --type=merge -p '{"metadata":{"finalizers": []}}'
   ```

4. 다음 명령을 실행하여 네임스페이스와 모든 열린 클러스터 관리 **CR**이 모두 제거되었는지 확인합니다.

   ```
   oc get crds | grep open-cluster-management.io | awk '{print $1}'
   oc get ns | grep open-cluster-management-agent
   ```

1.20.4. 클러스터를 제거한 후 etcd 데이터베이스 조각 모음

많은 관리형 클러스터가 있으면 **hub** 클러스터의 **etcd** 데이터베이스 크기에 영향을 미칠 수 있습니다. **OpenShift Container Platform 4.8**에서는 관리형 클러스터를 삭제하면 **hub** 클러스터의 **etcd** 데이터베이스 크기가 자동으로 줄어들지 않습니다. 일부 시나리오에서는 **etcd** 데이터베이스가 공간이 부족해질

수 있습니다. **etcdserver: mvcc:** 데이터베이스 공간이 초과된 오류가 표시됩니다. 이 오류를 수정하려면 데이터베이스 기록을 압축하고 **etcd** 데이터베이스를 조각 모음하여 **etcd** 데이터베이스의 크기를 줄입니다.

참고: **OpenShift Container Platform** 버전 **4.9** 이상에서는 **etcd Operator**가 디스크를 자동으로 조각 모음하고 **etcd** 기록을 압축합니다. 수동 조작이 필요하지 않습니다. 다음 절차는 **OpenShift Container Platform** 버전 **4.8** 및 이전 버전에 적용됩니다.

다음 절차를 완료하여 **etcd** 기록을 압축하고 **hub** 클러스터에서 **etcd** 데이터베이스 조각 모음을 풉니다.

1.20.4.1. 사전 요구 사항

- **OpenShift CLI(oc)**를 설치합니다.
- **cluster-admin** 권한이 있는 사용자로 로그인합니다.

1.20.4.2. 절차

1. **etcd** 기록을 압축합니다.
   a. **etcd** 멤버에 대한 원격 쉘 세션을 엽니다. 예를 들면 다음과 같습니다.

```
$ oc rsh -n openshift-etcd etcd-control-plane-0.example.com etcdctl endpoint status --cluster -w table
```

   b. 다음 명령을 실행하여 **etcd** 기록을 압축하십시오.

```
sh-4.4#etcdctl compact $(etcdctl endpoint status --write-out="json" |  egrep -o '"revision":[0-9]*' | egrep -o '[0-9]*' -m1)
```

   출력 예

```
$ compacted revision 158774421
```

2. *etcd 데이터베이스 조각 모음 및 etcd 데이터 조각 모음 에 설명된 대로 NOSPACE 경고를 지웁니다.*

1.21. 클러스터 백업 및 복원 OPERATOR

클러스터 백업 및 복원 Operator는 Kubernetes 허브 클러스터용 Red Hat Advanced Cluster Management 클러스터가 중단되어 다시 생성해야 하는 경우를 위한 재해 복구 솔루션을 제공합니다. 허브 클러스터에서 실행되며 OADP Operator 에 따라 Velero를 설치하고 허브 클러스터에서 데이터가 저장된 백업 스토리지 위치로의 연결을 생성합니다. Velero는 백업 및 복원 작업을 실행하는 구성 요소입니다. 클러스터 백업 및 복원 Operator 솔루션은 관리 클러스터, 애플리케이션, 정책 및 베어 메탈 자산과 같은 모든 Red Hat Advanced Cluster Management 허브 클러스터 리소스에 대한 백업 및 복원 지원을 제공합니다.

허브 클러스터 설치를 확장하는 타사 리소스의 백업을 지원합니다. 이 백업 솔루션을 사용하면 지정된 시간 간격으로 실행되는 cron 기반 백업 일정을 정의할 수 있습니다. 허브 클러스터가 다운되면 새 허브 클러스터를 배포하고 백업된 데이터가 새 허브 클러스터로 이동합니다.

클러스터 백업 및 복원 Operator가 자동으로 설치되지 않습니다. MultiClusterHub 리소스에서 cluster-backup 매개변수를 true 로 설정하여 백업 구성 요소를 활성화합니다. 클러스터 백업 Operator는 Red Hat Advanced Cluster Management가 설치된 open-cluster-management-backup 네임스페이스에 설치됩니다. 클러스터 백업 Operator를 설치하면 OADP Operator도 자동으로 설치됩니다.

참고:

- *OADP Operator 1.0에는 다중 아키텍처 빌드가 비활성화되어 있으며 공식 릴리스에 대한 x86_64 빌드만 생성합니다. 즉 x86_64 이외의 아키텍처를 사용하는 경우 백업 구성 요소에서 설치한 OADP Operator를 올바른 버전으로 교체해야 합니다. 이 경우 OADP Operator를 제거하고 아키텍처와 일치하는 Operator를 찾은 다음 설치합니다.*

- *이전에 hub 클러스터에 OADP Operator를 설치하고 사용한 경우, 구성 요소 네임스페이스에 설치된 OADP에서 백업 구성 요소가 작동하므로 이 버전을 제거합니다. 백업 구성 요소와 함께 설치된 OADP Operator가 소유한 DataProtectionApplication 리소스에 대해 동일한 스토리지 위치를 사용합니다. 이는 이전 Operator와 동일한 백업 데이터에 액세스합니다. 이제 Velero 백업 리소스가 이 허브 클러스터의 새 OADP Operator 네임스페이스 내에 로드됩니다.*

Velero 는 Red Hat Advanced Cluster Management hub 클러스터에 OADP Operator와 함께 설치됩니다. Velero는 Red Hat Advanced Cluster Management hub 클러스터 리소스를 백업하고 복원하는

데 사용됩니다.

Velero에 지원되는 스토리지 공급자 목록은 S3 호환 오브젝트 저장소 공급자를 참조하십시오.

- 사전 요구 사항
- 백업 및 복원 Operator 아키텍처
  - 백업되는 리소스
    - 백업 데이터 확장
    - 관리형 클러스터 활성화 시 복원된 리소스
  - 리소스 요청 및 제한 사용자 정의
  - 서버 측 암호화를 사용하여 데이터 보호
  - 클러스터 백업 예약
- 백업 복원
  - 새 hub 클러스터 준비
  - 복원하기 전에 hub 클러스터 정리
  - 관리 활성화 시 복원된 리소스
  - 수동 리소스 복원

- 백업을 확인하는 동안 수동 리소스 복원
- 가져온 관리 클러스터 복원
- 활성화 리소스 복원

- 활성 수동 구성
  - 관리형 클러스터 활성화 데이터

- 재해 복구

- 정책을 사용하여 백업 검증

1.21.1. 사전 요구 사항

- 백업이 저장된 클라우드 스토리지에 대한 인증 정보 시크릿 생성 단계를 완료해야 합니다. 보안 리소스는 **open-cluster-management-backup** 네임스페이스인 **OADP Operator** 네임스페이스에 생성해야 합니다.

- **DataProtectionApplication** 리소스를 생성할 때 생성된 시크릿을 사용합니다.

  DataProtectionApplication 리소스의 인스턴스를 생성하려면 다음 단계를 완료합니다.

  1. **Red Hat OpenShift Container Platform** 콘솔에서 **Operator > 설치된 Operator** 를 선택합니다.
  2. **DataProtectionApplication**에서 **Create instance** 를 클릭합니다.
  3. **{ocp-short)** 콘솔을 사용하거나 **DataProtectionApplication** 예제에서 언급한 **YAML** 파일을 사용하여 구성을 선택하여 **Velero** 인스턴스를 생성합니다.

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

5. *DataProtectionApplication 리소스에 대해 사양(spec:) 값을 적절하게 설정합니다. 그런 다음 만들기 를 클릭합니다.*

쉽게 사용할 수 있도록 리소스 값이 언급됩니다. 기본 백업 스토리지 위치를 사용하려는 경우 backupStorageLocations 섹션에서 다음 값, default: true 를 설정합니다. DataProtectionApplication 리소스는 다음 YAML 파일과 유사합니다.

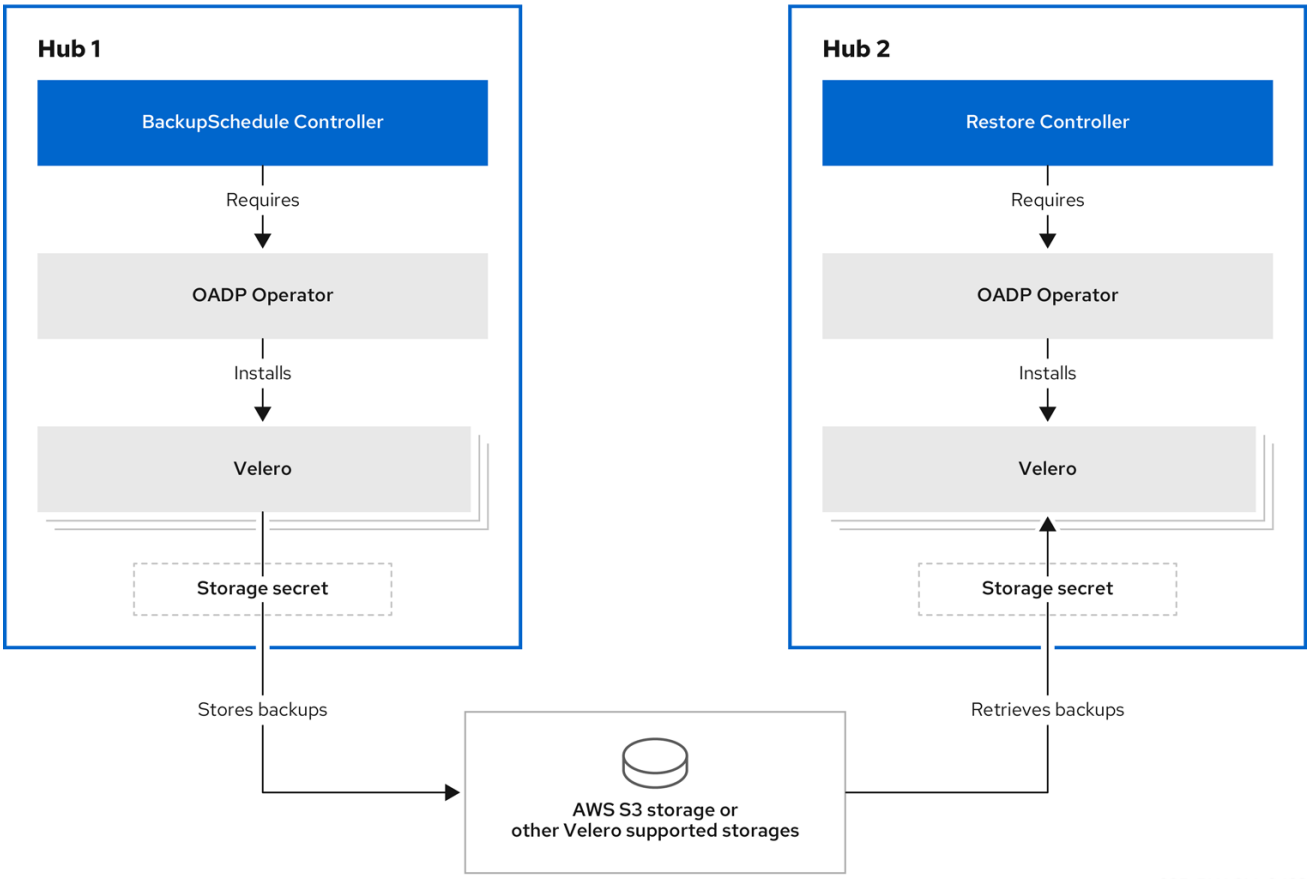
DataProtectionApplication 리소스는 다음 YAML 파일과 유사합니다.

```
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"
```

DataProtectionApplication 리소스를 생성하는 예제를 참조하십시오.

1.21.2. 백업 및 복원 Operator 아키텍처

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



1.21.2.1. 백업되는 리소스

클러스터 백업 및 복원 **Operator** 솔루션은 관리 클러스터, 애플리케이션, 정책 및 베어 메탈 자산과 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 간격으로 실행되고 허브 클러스터 콘텐츠의 최신 버전을 지속적으로 백업하는 **cron** 기반 백업 일정을 정의할 수 있습니다.

허브 클러스터가 중단된 경우 **hub** 클러스터를 교체해야 하거나 재해 시나리오에 있는 경우 새 허브 클러스터를 배포하고 백업할 수 있으며 데이터를 새 허브 클러스터로 이동할 수 있습니다.

백업 데이터를 식별하기 위해 다음과 같이 정렬된 클러스터 백업 및 복원 프로세스 목록을 확인합니다.

- **MultiClusterHub** 네임스페이스의 모든 리소스를 제외합니다. 이는 현재 허브 클러스터 **ID**에 연결된 설치 리소스를 백업하지 않기 위한 것이며 백업하면 안 됩니다.
- **.open-cluster-management.io** 가 접미사로 지정된 **API** 버전으로 모든 **CRD**를 백업합니다. 이 접미사는 모든 **Red Hat Advanced Cluster Management** 리소스가 백업되었음을 나타냅니다.
- 다음 **API** 그룹에서 모든 **CRD**를 백업합니다. **argoproj.io,app.k8s.io,core.observatorium.io,hive.openshift.io.**
- 다음 **API** 그룹에서 모든 **CRD**를 제외합니다. **admission.cluster.open-cluster-management.io,admission.work.open-cluster-management.io,internal.open-cluster-management.io,operator.open-cluster-management.io,work.open-cluster-management.io,search.open-cluster-management.io,admission.hive.openshift.io,velero.io.io .**
- 포함된 **API** 그룹의 일부이지만 필요하지 않거나 백업되는 **owner-resources**에 의해 다시 생성되는 다음 **CRD**를 제외합니다. **clustermanagementaddon,observabilityaddon,applicationmanager,certpolicycontroller,iampolicycontroller,searchcollector,workmanager,backupschedule,restore,clusterclaim.cluster.open-cluster-management.io.**
- **cluster.open-cluster-management.io/type,hive.openshift.io/secret-type,cluster.open-cluster-management.io/backup** 라벨이 있는 보안 및 **ConfigMap**을 백업합니다.
- 백업하려는 기타 리소스에 대해 **cluster.open-cluster-management.io/backup** 레이블을 사용하며 이전에 언급한 기준에 포함되지 않습니다. 다음 예제를 참조하십시오.

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

참고: **hive.openshift.io.ClusterDeployment** 리소스에서 사용하는 보안을 백업해야 하며 콘솔을 사용하여 클러스터를 생성할 때만 **cluster.open-cluster-management.io/backup** 레이블

로 자동으로 주석이 추가됩니다. 대신 **GitOps**를 사용하여 **Hive** 클러스터를 배포하는 경우 **cluster.open-cluster-management.io/backup** 레이블을 **ClusterDeployment** 에서 사용하는 보안에 수동으로 추가해야 합니다.

- 백업하지 않으려는 특정 리소스를 제외합니다. 예를 들어 백업 프로세스에서 **Velero** 리소스를 제외하려면 다음 예제를 참조하십시오.

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

==== extend backup data

리소스에 **cluster.open-cluster-management.io/backup** 라벨을 추가하여 클러스터 백업 및 복원으로 타사 리소스를 백업할 수 있습니다. 레이블 값은 빈 문자열을 포함하여 모든 문자열이 될 수 있습니다. 백업 중인 구성 요소를 식별하는 데 도움이 될 수 있는 값을 사용합니다. 예를 들어, 구성 요소를 **IDP** 솔루션으로 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** 레이블에 **cluster-activation** 값을 사용합니다. 관리형 클러스터 활성화 리소스를 복원하면 관리 클러스터는 복원이 시작된 허브 클러스터에서 적극적으로 관리합니다.

1.21.2.1.1. 관리형 클러스터 활성화 시 복원된 리소스

리소스에 **cluster.open-cluster-management.io/backup** 레이블을 추가하면 **acm-resources-generic-schedule** 백업에서 리소스가 자동으로 백업됩니다. 관리 클러스터를 새 허브 클러스터로 이동한 후 복원된 리소스에서 **veleroManagedClustersBackupName:latest** 가 사용되는 경우에만 리소스를 복원해야 하는 경우 레이블 값을 **cluster-activation** 으로 설정해야 합니다. 이렇게 하면 관리 대상 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
```

cluster.open-cluster-management.io/backup: cluster-activation 레이블로 식별되고 **acm-resources-generic-schedule** 백업으로 저장된 활성화 데이터 리소스 외에도 클러스터 백업 및 복원 **Operator**에는 기본적으로 활성화 세트의 몇 가지 리소스가 포함됩니다. **acm-managed-clusters-schedule** 백업으로 지원되는 리소스는 다음과 같습니다.

- *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*

1.21.2.2. 리소스 요청 및 제한 사용자 정의

Velero가 처음 설치되면 다음 샘플에 정의된 대로 Velero pod가 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 백업하는 경우 업데이트해야 할 수 있습니다. 예를 들어, 백업이 2000 클러스터를 관리하는 hub 클러스터에서 실행되면 메모리 부족 오류(OOM)로 인해 Velero pod가 충돌합니다. 이 시나리오에 대해 다음 구성을 통해 백업이 완료될 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
```

DataProtectionApplication 매개변수에 대한 자세한 내용은 Velero 리소스 요청 및 제한 사용자 지정을 참조하십시오.

1.21.2.3. 서버 측 암호화를 사용하여 데이터 보호

서버 측 암호화는 스토리지 위치에서 데이터를 수신하는 애플리케이션 또는 서비스의 데이터 암호화입니다. 백업 메커니즘 자체는 전송 중(백업 스토리지 위치로 이동) 또는 유휴 상태에서 데이터를 암호화하지 않으며(백업 스토리지 위치의 디스크에 저장됨)는 데이터를 암호화하지 않습니다. 대신 오브젝트 및 스냅샷 시스템의 기본 메커니즘에 의존합니다.

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

어 있습니다.

serverSideEncryption 및 **kmsKeyId** 매개변수를 사용하여 **Amazon S3**에 저장된 백업의 암호화를 활성화할 수 있습니다. 자세한 내용은 **Backup Storage Location YAML** 을 참조하십시오. 다음 샘플은 **DataProtectionApplication** 리소스를 설정할 때 **AWS KMS** 키 **ID**를 지정합니다.

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

Velero 지원 스토리지 공급자를 참조하여 다른 스토리지 공급자의 구성 가능한 모든 매개변수에 대해 알아보십시오.

1.21.2.4. 클러스터 백업 예약

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

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name: schedule-acm
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 bsch -n <oadp-operator-ns>
```

이전 명령의 **< oadp-operator-ns >** 매개변수는 **BackupSchedule** 이 생성되는 네임스페이스입니다. 이는 **OADP Operator**가 설치된 네임스페이스와 동일합니다. **backupschedule.cluster.open-cluster-management.io** 리소스는 백업을 생성하는 데 사용되는 **6**개의 **schedule.velero.io** 리소스를 생성합니다. 다음 명령을 실행하여 예약된 백업 목록을 확인합니다.

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

리소스는 다음 그룹에서 별도로 백업됩니다.

- Hive, Red Hat Advanced Cluster Management 및 사용자 생성 자격 증명을 위한 3개의 백업 파일이 포함된 인증 정보 백업

- 리소스 백업: Red Hat Advanced Cluster Management 리소스용 백업 1개와 일반 리소스용 백업이 포함됩니다. 이러한 리소스는 cluster.open-cluster-management.io/backup 레이블을 사용합니다.

- 관리형 클러스터 백업: 백업이 복원되는 허브 클러스터에 대한 관리형 클러스터 연결을 활성화하는 리소스만 포함합니다.

참고: 리소스 백업 파일에는 관리되는 클러스터별 리소스가 포함되어 있지만 관리 클러스터를 허브 클러스터에 연결하는 리소스의 서브 세트는 포함되지 않습니다. 관리 클러스터를 연결하는 리소스를 활성화 리소스라고 하며 관리형 클러스터 백업에 포함됩니다. 새 허브 클러스터에서 인증 정보 및 리소스 백업에 대해서만 백업을 복원하면 새 허브 클러스터에서 Hive API를 사용하여 생성된 모든 관리 클러스터를 분리된 상태로 표시합니다. 그러나 가져오기 작업을 사용하여 기본 허브 클러스터에서 가져온 관리형 클러스터는 수동 허브 클러스터에서 활성화 데이터가 복원될 때만 나타납니다. 현재 관리 클러스터는 백업 파일을 생성한 원래 허브 클러스터에 계속 연결됩니다.

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

1.21.3. 백업 복원

일반적인 복원 시나리오에서는 백업이 실행되는 hub 클러스터를 사용할 수 없으며 백업된 데이터를 새 허브 클러스터로 이동해야 합니다. 이 작업은 새 허브 클러스터에서 클러스터 복원 작업을 실행하여 수행됩니다. 이 경우 복원 작업은 백업이 생성된 것과 다른 허브 클러스터에서 실행됩니다.

백업이 수집된 동일한 허브 클러스터에서 데이터를 복원하려는 경우도 있으므로 이전 스냅샷의 데이터를 복구할 수 있습니다. 이 경우 복원 및 백업 작업 모두 동일한 허브 클러스터에서 실행됩니다.

hub 클러스터에서 restore.cluster.open-cluster-management.io 리소스를 생성한 후 다음 명령을 실행하여 복원 작업 상태를 가져올 수 있습니다. oc get restore -n <oadp-operator-ns>. 또한 백업 파일에 포함된 백업 리소스가 생성되었는지 확인할 수 있어야 합니다.

*참고: **restore.cluster.open-cluster-management.io** 리소스가 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 생성한 세 가지 백업 유형을 모두 복원하는 데 사용됩니다. 그러나 특정 유형의 백업만 설치하도록 선택할 수 있습니다(관리된 클러스터만, 사용자 인증 정보만 또는 허브 클러스터 리소스만).

복원은 다음과 같은 세 가지 필수 사양 속성을 정의합니다. 여기서 복원 논리는 백업 파일 유형에 대해 정의됩니다.

- ***veleroManagedClustersBackupName** 은 관리 클러스터 활성화 리소스에 대한 복원 옵션을 정의하는 데 사용됩니다.*
- ***veleroCredentialsBackupName** 은 사용자 자격 증명의 복원 옵션을 정의하는 데 사용됩니다.*
- ***veleroResourcesBackupName** 은 허브 클러스터 리소스(애플리케이션,정책 및 관리형 클러스터 수동 데이터와 같은 기타 허브 클러스터 리소스)에 대한 복원 옵션을 정의하는 데 사용됩니다.*

  이전에 언급한 속성의 유효한 옵션은 다음과 같습니다.

  - ***latest -** 이 속성은 이 유형의 백업에 사용 가능한 마지막 백업 파일을 복원합니다.*
  - *건너뛰기 **-** 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 다음 템플릿 규칙인 **< restore.cluster.open-cluster-management.io name>-<velero-backup-resource-name > .** 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-20210102205438 은 관리형 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 관리 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: 백업 유형에 skip 이 사용되는 경우 restore.velero.io 가 생성되지 않습니다.

클러스터 복원 리소스의 다음 YAML 샘플을 확인합니다. 이 샘플에서는 사용 가능한 최신 백업 파일을 사용하여 세 가지 유형의 백업 파일이 복원됩니다.

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

참고: 관리형 클러스터 백업의 acm-managed-clusters 백업이 다른 hub 클러스터에 복원되면 Hive API에서 생성한 관리형 클러스터만 새 hub 클러스터에 자동으로 연결됩니다. 다른 모든 관리형 클러스터는 Pending Import 상태로 유지되며 새 hub 클러스터로 다시 가져와야 합니다. 자세한 내용은 가져온 관리 클러스터 복구(기술 프리뷰)를 참조하십시오.

1.21.3.1. 새 hub 클러스터 준비

새 허브 클러스터에서 복원 작업을 실행하기 전에 hub 클러스터를 수동으로 구성하고 초기 허브 클러스터에 동일한 운영자를 설치해야 합니다. Red Hat Advanced Cluster Management Operator를 초기

허브 클러스터와 동일한 네임스페이스에 설치하고 DataProtectionApplication 리소스를 만든 다음 이전에 데이터를 백업한 초기 허브 클러스터와 동일한 스토리지 위치에 연결해야 합니다.

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

1.21.3.2. 복원하기 전에 hub 클러스터 정리

***Velero**는 현재 **hub** 클러스터에서 기존 백업 리소스를 건너뜁니다. 이렇게 하면 새 허브 클러스터에서 허브 클러스터 데이터를 복원할 때 사용할 수 있는 시나리오가 제한됩니다. 새 **hub** 클러스터를 사용하고 복원이 두 번 이상 적용되는 경우 복원을 실행하기 전에 데이터를 정리하지 않는 한 허브 클러스터는 수동 구성으로 사용하지 않는 것이 좋습니다. 새 허브 클러스터의 데이터는 복원된 리소스와 함께 사용할 수 있는 데이터를 반영하지 않습니다.*

***restore.cluster.open-cluster-management.io** 리소스가 생성되면 클러스터 백업 및 복원 **Operator**는 **Velero** 복원이 시작되기 전에 **hub** 클러스터를 정리하여 복원을 준비하는 일련의 단계를 실행합니다.*

***cleanup** 옵션은 **cleanupBeforeRestore** 속성을 사용하여 정리할 오브젝트의 하위 집합을 식별합니다. 이 정리에 대해 세 가지 옵션을 설정할 수 있습니다.*

- ***none:** 필요하지 않음, **Velero** 복원만 시작합니다. 이는 새로운 허브 클러스터에서 사용됩니다.*
- ***CleanupRestored:** 이전 **Red Hat Advanced Cluster Management** 복원에서 생성된 모든 리소스를 정리합니다. 이 속성은 **cleanup All** 속성보다 개입이 적기 때문에 사용하는 것이 좋습니다.*
- ***CleanupAll:** 복원 작업으로 인해 리소스가 생성되지 않더라도 **Red Hat Advanced Cluster Management** 백업에 포함될 수 있는 **hub** 클러스터 클러스터의 모든 리소스를 정리합니다. 이는 정리가 필요한 **hub** 클러스터에 추가 콘텐츠를 만들 때 사용됩니다. 이 옵션은 이전 백업이 아닌 사용자가 생성한 **hub** 클러스터의 리소스를 정리하므로 이 옵션을 신중하게 사용합니다. **Initialup Restored** 옵션을 사용하고 허브 클러스터가 재해 시나리오의 수동 클러스터로 지정되면 **hub** 클러스터 콘텐츠를 수동으로 업데이트하지 않는 것이 좋습니다. **clear upAll** 옵션을 마지막 대안으로 사용합니다.*

참고:

-

복원된 백업에 리소스가 없는 경우 **Velero**는 **velero** 복원 리소스에 대한 상태 **PartiallyFailed** 를 설정합니다. 즉, 해당 백업이 비어 있기 때문에 생성된 **restore.velero.io** 리소스에서 리소스를 복원하지 않는 경우 **restore.cluster.open-cluster-management.io** 리소스가 **PartiallyFailed** 상태에 있을 수 있습니다.

- 새 백업을 사용할 수 있는 경우 **syncRestoreWithNewBackups:true** 를 사용하여 수동 데이터를 계속 복원하지 않는 한 **restore.cluster.open-cluster-management.io** 리소스가 한 번 실행됩니다. 이 경우 동기화 샘플이 있는 복원 페시브를 따릅니다. *백업을 확인하는 동안 페시브 리소스 복원* 을 참조하십시오. 복원 작업이 완료되고 동일한 허브 클러스터에서 다른 복원 작업을 실행하려면 새 **restore.cluster.open-cluster-management.io** 리소스를 생성해야 합니다.

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

1.21.3.3. 활성화 리소스 복원

허브 클러스터에서 클러스터를 관리하려면 **restore-passive-activate** 샘플을 사용합니다. 이 경우 다른 데이터가 수동 리소스를 사용하는 허브 클러스터에서 이미 복원되었다고 가정합니다.

1.21.3.4. 수동 리소스 복원

수동 데이터는 시크릿, **ConfigMaps**, 애플리케이션, 정책 및 모든 관리 클러스터 사용자 정의 리소스와 같은 백업 데이터로, 관리 클러스터와 허브 클러스터 간의 연결을 활성화하지 않습니다. 백업 리소스는 인증 정보 백업 및 복원 리소스에 의해 **hub** 클러스터에서 복원됩니다.

1.21.3.5. 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.

VeleroResourcesBackupName 및 **VeleroCredentialsBackupName** 매개변수를 **latest** 로 설정하고 **VeleroManagedClustersBackupName** 매개변수를 건너뛰도록 설정합니다. **VeleroManagedClustersBackupName** 이 **latest** 로 설정된 직후 관리 클러스터는 새 허브 클러스터에서 활성화되고 이제 기본 허브 클러스터입니다.

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

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

예를 들어 다음 리소스는 **10**분마다 백업을 확인합니다.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm-passive-sync
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
```

1.21.3.6. 가져온 관리 클러스터 복원

Hive API를 사용하여 기본 허브 클러스터와 연결된 관리 클러스터만 활성화 데이터가 복원되는 새 **hub** 클러스터와 자동으로 연결됩니다. 이러한 클러스터는 클러스터 탭의 **Create cluster** 버튼을 사용하여 기본 허브 클러스터에 생성되었습니다. **Import cluster** 버튼을 사용하여 초기 **hub** 클러스터와 연결된 관리 클러스터는 활성화 데이터가 복원될 때 가져오기 보류 중으로 표시되고 새 허브 클러스터에서 다시 가져와야 합니다.

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

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

Pending Import 상태의 각 클러스터의 관리 클러스터 네임스페이스에 **auto-import-secret** 리소스를 생성합니다. 가져오기 구성 요소에서 새 허브 클러스터에서 자동 가져오기를 시작할 수 있는 충분한 권한이 있는 **kubeconfig** 또는 토큰을 사용합니다. 관리형 클러스터와 연결하는 토큰을 사용하여 각 관리 클러스터에 대한 액세스 권한이 있어야 합니다. 토큰에는 **klusterlet** 역할 바인딩 또는 동일한 권한이 있는 역할이 있어야 합니다.

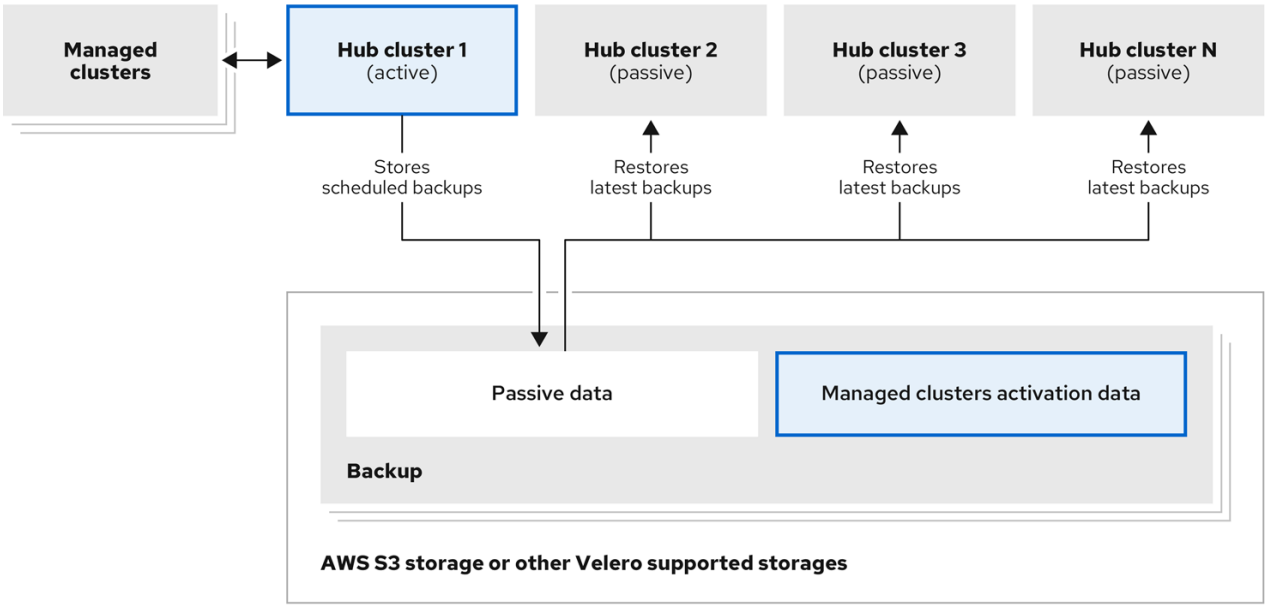
1.21.4. 활성 수동 구성

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

*수동 허브 클러스터는 지속적으로 최신 백업을 검색하고 수동 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 중단될 때 기본 허브 클러스터가 기본 허브가 되도록 보류 중입니다.*

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

*다음 다이어그램에서 활성 허브 클러스터는 로컬 클러스터를 관리하고 주기적으로 **hub** 클러스터 데이터를 백업합니다.*



패시브 허브 클러스터는 관리형 클러스터 활성화 데이터를 제외하고 이 데이터를 복원하여 관리 클러스터를 수동 허브 클러스터로 이동합니다. 패시브 허브 클러스터는 수동 데이터를 지속적으로 복원할 수 있으며 백업을 확인하는 동안 수동 리소스 복원 섹션을 참조하십시오. 패시브 허브 클러스터는 수동 데이터를 일회성 작업으로 복원할 수 있으며 자세한 내용은 수동 리소스 복원 섹션을 참조하십시오.

1.21.4.1. 관리형 클러스터 활성화 데이터

관리형 클러스터 활성화 데이터 또는 기타 활성화 데이터는 백업 리소스입니다. 새 **hub** 클러스터에서 활성화 데이터를 복원하면 관리 클러스터는 복원이 실행되는 허브 클러스터에서 적극적으로 관리됩니다. 활성화 데이터 리소스는 **cluster.open-cluster-management.io/backup: cluster-activation** 레이블을 사용할 때 관리 클러스터 백업 및 리소스 일반 백업에 의해 저장됩니다.

1.21.4.2. 관리 활성화 시 복원된 리소스

cluster.open-cluster-management.io/backup: cluster-activation 레이블을 리소스에 추가하면 리소스가 **acm-resources-generic-schedule** 백업 리소스에서 자동으로 백업됩니다. 복원 리소스에서 **veleroManagedClustersBackupName:latest** 라벨 값을 설정하면 일반적으로 리소스를 복원해야 합니다. 관리 클러스터를 새 허브 클러스터로 이동할 때 이러한 리소스를 복원해야 하는 경우 **veleroManagedClustersBackupName:latest** 라벨 값을 **cluster-activation** 으로 설정합니다. 이렇게 하면 관리 대상 클러스터 활성화가 시작되지 않는 한 리소스가 복원되지 않습니다.

리소스는 다음 예와 유사합니다.

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

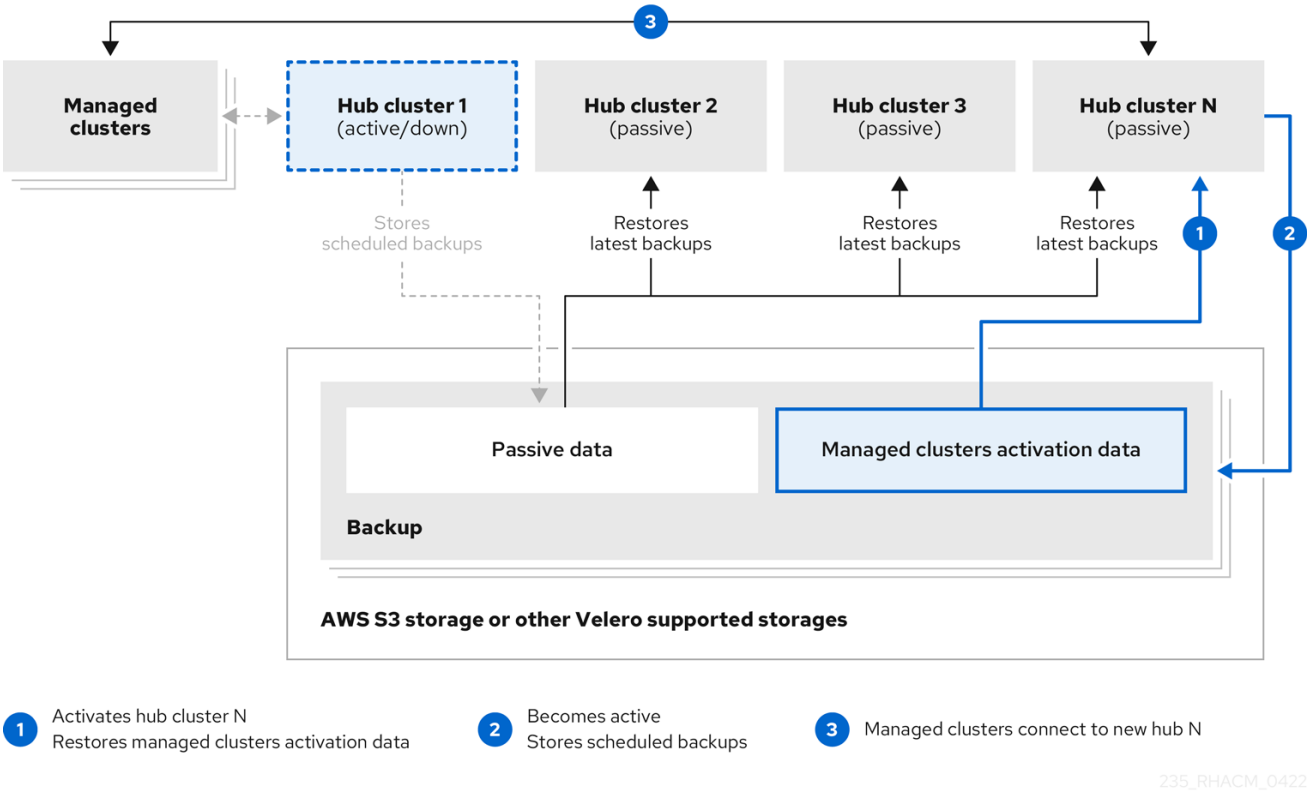
acm-managed-clusters-schedule 리소스에서 지원하는 활성화 세트에도 기본 리소스가 있습니다. **acm-managed-clusters-schedule** 리소스에서 복원한 다음 기본 리소스를 확인합니다.

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

- *clusterclaim.hive.openshift.io*
- *clustercurator.cluster.open-cluster-management.io*
- *clustersync.hiveinternal.openshift.io*
- *baremetalhost.metal3.io*
- *bmceventsubscription.metal3.io*
- *hostfirmwaresettings.metal3.io*

1.21.5. 재해 복구

*기본 허브 클러스터가 다운되면 관리자가 관리형 클러스터를 인수하도록 수동 허브 클러스터 중 하나를 선택합니다. 다음 이미지에서 관리자는 **Hub** 클러스터 **N** 을 새 기본 허브 클러스터로 사용하기로 결정합니다.*



***hub cluster N** 은 관리형 클러스터 활성화 데이터를 복원합니다. 이 시점에서 관리 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 클러스터, **Hub** 클러스터 **N** 에서 백업을 활성화합니다.*

*다른 모든 수동 허브 클러스터에서는 이제 새 기본 허브 클러스터에서 생성한 백업 데이터를 사용하여 수동 데이터를 복원합니다. **Hub N** 은 이제 기본 허브 클러스터로서 클러스터를 관리하고 데이터를 백업합니다.*

1.21.6. 정책을 사용하여 백업 검증

*클러스터 백업 및 복원 **Operator Helm** 차트(**cluster-backup-chart**)는 **hub** 클러스터에 **backup-restore-enabled** 정책을 설치합니다. 이 정책은 백업 및 복원 구성 요소의 문제에 대해 알려주는 데 사용됩니다. **backup-restore-enabled** 정책에는 다음 제약 조건을 확인하는 템플릿 세트가 포함됩니다.*

- ***Pod** 검증*

  *다음 템플릿은 백업 구성 요소 및 종속 항목에 대한 **Pod** 상태를 확인합니다.*

  -

ACM-backup-pod-running 템플릿은 백업 및 복원 Operator Pod가 실행 중인지 확인합니다.

- *OADP-pod-running 템플릿은 OADP Operator Pod가 실행 중인지 확인합니다.*
- *Velero-pod-running 템플릿은 Velero pod가 실행 중인지 확인합니다.*

- *Data Protection 애플리케이션 유효성 검사*
  - *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 상태가 BackupCollision 이 아닌지 확인합니다. BackupSchedule.cluster.open-cluster-management.io 가 현재 허브 클러스터에 있는 경우. 이렇게 하면 스토리지 위치에 백업 데이터를 작성할 때 현재 hub 클러스터가 다른 허브 클러스터와 충돌하지 않는지 확인합니다.*

    BackupCollision 상태에 대한 정의의 경우 Backup Collisions 섹션을 참조하십시오.
- *BackupSchedule 및 복원 상태 검증*
  - *ACM -backup-phase-validation 템플릿은 현재 클러스터에 BackupSchedule.cluster.open-cluster-management.io 가 있는 경우 상태가 Failed, Empty 상태가 아닌지 확인합니다. 이렇게 하면 이 클러스터가 기본 hub 클러스터이고 백업을 생성하는 경우 BackupSchedule.cluster.open-cluster-management.io 상태가 정상입니다.*

- *동일한 템플릿에서 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 복원 중인 경우 **Restore.cluster.open-cluster-management.io** 상태가 정상입니다.*

- *백업 보유 상태 검증*

  - ***ACM -managed-clusters-schedule-backups-available** 템플릿은 **BackupStorageLocation.velero.io** 에서 지정한 위치에서 **Backup.velero.io** 리소스를 사용할 수 있는지, 백업이 **BackupSchedule.cluster.open-cluster-management.io** 리소스에서 생성하는 경우 백업을 생성합니다. 이는 **backup** 및 **restore** 연산자를 사용하여 백업을 한 번 이상 실행했는지 확인합니다.*

- *완료를 위한 백업*

  - ***acm-backup-in-progress-report** 템플릿은 **Backup.velero.io** 리소스가 **InProgress** 상태에 고착되어 있는지 확인합니다. 이 검증은 리소스가 많은 경우 백업이 실행될 때 **velero Pod**가 재시작되고 백업이 완료되지 않고 계속 진행 중이므로 백업이 추가됩니다. 일반적인 백업 중에 백업 리소스는 실행 중 어느 시점에서 진행 중이지만 중단되지 않고 완료로 실행됩니다. **acm-backup-in-progress-report** 템플릿은 일정이 실행되는 동안 경고가 표시되고 백업이 진행 중인 것을 확인하는 것이 정상입니다.*

- ***cron** 작업으로 적극적으로 실행되는 백업*

  - ***BackupSchedule.cluster.open-cluster-management.io** 는 스토리지 위치에 적극적으로 실행되고 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.*

    ***acm-validation-policy-schedule** 백업은 백업 **cron** 일정에 설정된 시간이 설정된 후 만료되도록 설정됩니다. 백업을 생성하기 위해 **cron** 작업이 실행되지 않으면 만료되고 새 작업이 생성되지 않기 때문에 이전 **acm-validation-policy-schedule** 백업이 삭제됩니다. 결과적으로 언제든지 **acm-validation-policy-schedule** 백업이 없는 경우 백업을 생성하는 활성 **cron** 작업이 없음을 의미합니다.*

    *이 정책은 **hub** 클러스터가 활성 상태이고 백업을 생성 또는 복원할 때 모든 백업 문제를 **hub** 클러스터 관리자에게 알리는 데 도움이 됩니다.*

클러스터 백업 및 복원 연산자를 활성화하고 관리하는 방법에 대해 알아보고, 백업 및 복원 연산자 관

리를 참조하십시오.

1.21.7. 백업 및 복원 Operator 관리

클러스터 백업 및 복원 Operator를 활성화하여 클러스터 리소스의 백업 및 복원을 예약합니다.

필수 액세스: 클러스터 관리자

- 사전 요구 사항
- 백업 및 복원 연산자 활성화
- backup 및 restore 연산자 사용
- 복원 이벤트 보기

1.21.7.1. 사전 요구 사항

활성 및 수동 허브 클러스터 모두의 경우:

- Red Hat OpenShift Container Platform 클러스터에서 Kubernetes Operator 버전 2.5.x용 Red Hat Advanced Cluster Management를 설치합니다. Red Hat Advanced Cluster Management를 설치할 때 MultiClusterHub 리소스가 자동으로 생성되고 실행 중 상태가 표시됩니다.
- 클러스터 백업 및 복원 Operator를 수동으로 설치해야 합니다. 클러스터 백업 및 복원 Operator (cluster-backup)를 활성화합니다. cluster-backup 매개변수를 true 로 설정하여 MultiClusterHub 리소스를 편집합니다. 이렇게 하면 cluster-backup 리소스와 동일한 네임스페이스에 OADP Operator가 설치됩니다.

패시브 허브 클러스터의 경우:

- 수동 허브 클러스터에서 복원 작업을 실행하기 전에 hub 클러스터를 수동으로 구성하고 활성 허브 클러스터에 모든 운영자를 설치하고 활성 허브 클러스터와 동일한 네임스페이스에 설치

해야 합니다.

- *Red Hat Advanced Cluster Management Operator가 초기 허브 클러스터와 동일한 네임스페이스에 설치되어 있는지 확인합니다. 그런 다음 DataProtectionApplication 리소스를 만들고 초기 허브 클러스터가 데이터를 백업한 것과 동일한 스토리지 위치에 연결합니다. 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"
```

- *복원 작업을 실행하기 전에 Ansible Automation Platform, Red Hat OpenShift Container Platform GitOps 또는 인증서 관리자와 같은 다른 Operator가 설치되어 있는지 확인합니다. 이렇게 하면 새 허브 클러스터가 초기 허브 클러스터와 동일한 방식으로 구성됩니다.*

- *패시브 허브 클러스터는 백업 및 복원 Operator를 설치할 때 초기 허브 클러스터와 동일한 네임스페이스 이름을 사용하고 이전 허브 클러스터에 구성된 다른 Operator를 사용해야 합니다.*

1.21.7.2. 백업 및 복원 연산자 활성화

처음으로 MultiClusterHub 리소스를 생성할 때 클러스터 백업 및 복원 Operator를 활성화할 수 있습니다. cluster-backup 매개변수가 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 제거하려면 cluster-backup 을 false 로 설정합니다.

백업 및 복원 Operator가 활성화되면 MultiClusterHub 리소스가 다음 YAML 파일과 유사합니다.

```
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: false
      name: cluster-proxy-addon
```

```
    - enabled: true <<<<<<<<<
      name: cluster-backup
  separateCertificateManagement: false
```

1.21.7.3. backup 및 restore 연산자 사용

백업을 예약 및 복원하려면 다음 단계를 완료합니다.

1. backup and restore operator, backupschedule.cluster.open-cluster-management.io 및 restore.cluster.open-cluster-management.io 리소스를 사용하여 cluster_v1beta1_backupschedule.yaml 샘플 파일을 사용하여 backupschedule.cluster.open-cluster-management.io 리소스를 생성합니다. cluster-backup-operator 샘플을 참조하십시오. 다음 명령을 실행하여 cluster_v1beta1_backupschedule.yaml 샘플 파일을 사용하여 backupschedule.cluster.open-cluster-management.io 리소스를 생성합니다.

```
kubectl create -n <oadp-operator-ns> -f
config/samples/cluster_v1beta1_backupschedule.yaml
```

리소스는 다음 파일과 유사합니다.

```
apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name: schedule-acm
spec:
  veleroSchedule: 0 */6 * * * # Create a backup every 6 hours
  veleroTtl: 72h # deletes scheduled backups after 72h; optional, if not specified, the
maximum default value set by velero is used - 720h
```

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

- veleroSchedule 은 필수 속성이며 백업을 예약하는 cron 작업을 정의합니다.
- veleroTtl 은 선택적 속성이며 예약된 백업 리소스에 대한 만료 시간을 정의합니다. 지정하지 않으면 Velero에 의해 설정된 최대 기본값이 사용되며, 이는 10.0.0.1 h 입니다.

2. 3 schedule.velero.io 리소스의 정의를 표시하는 backupschedule.cluster.open-cluster-management.io 리소스의 상태를 확인합니다. 다음 명령을 실행합니다.

```
oc get bsch -n <oadp-operator-ns>
```

3. 복원 시나리오에서는 복원 시나리오를 위해 다른 허브 클러스터에서 복원 작업이 실행됩니다. 복원 작업을 시작하려면 백업을 복원하려는 **hub** 클러스터에서 **restore.cluster.open-cluster-management.io** 리소스를 생성합니다.

   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.

4. 다음 명령을 실행하여 **cluster_v1beta1_restore.yaml** 샘플 파일을 사용하여 **restore.cluster.open-cluster-management.io** 리소스를 생성합니다. **oadp-operator-ns** 를 **OADP Operator**를 설치하는 데 사용되는 네임스페이스 이름으로 교체해야 합니다. **OADP Operator** 설치 네임스페이스의 기본값은 **oadp-operator** 입니다.

```
kubectl create -n <oadp-operator-ns> -f config/samples/cluster_v1beta1_restore.yaml
```

   리소스는 다음 파일과 유사합니다.

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

   restore.cluster.open-cluster-management.io 의 세 가지 필수 사양 속성에 대한 다음 설명을 확인합니다.

- **veleroManagedClustersBackupName** 은 관리 클러스터 작동 데이터의 복원 옵션을 정의하는 데 사용됩니다.
- **veleroCredentialsBackupName** 은 사용자 자격 증명의 복원 옵션을 정의하는 데 사용됩니다.
- **veleroResourcesBackupName** 은 허브 클러스터 리소스(애플리케이션,정책 및 조작된 클러스터 수동 데이터와 같은 기타 허브 리소스)에 대한 복원 옵션을 정의하는 데 사용됩니다.

이전에 언급한 속성의 유효한 옵션은 다음과 같습니다.

- ***latest -** 이 속성은 이 유형의 백업에 사용 가능한 마지막 백업 파일을 복원합니다.*
- *건너뛰기 **-** 이 속성은 현재 복원 작업을 사용하여 이 유형의 백업을 복원하지 않습니다.*
- ***backup_name -** 이 속성은 이름을 참조하여 지정된 백업을 복원합니다.*

5. *다음 명령을 실행하여 **Velero** 복원 리소스를 확인합니다.*

```
oc get restore.velero.io -n <oadp-operator-ns>
```

*다음 **YAML** 예제를 보고 다른 유형의 백업 파일을 복원합니다.*

- *세 가지 유형의 백업 리소스를 모두 복원합니다.*

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

- *관리형 클러스터 리소스만 복원하십시오.*

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

- ***acm-managed-clusters-schedule-202902205438** 백업을 사용하여 관리 클러스터의 리소스만 복원하십시오.*

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

참고:

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

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

1.21.7.4. 복원 이벤트 보기

다음 명령을 사용하여 복원 이벤트에 대한 정보를 가져옵니다.

```
oc describe -n <oadp-n> <restore-name>
```

이벤트 목록은 다음 샘플과 유사합니다.

```
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
```

필요한 사양 속성 및 유효한 옵션에 대한 설명은 백업 복원 을 참조하십시오.