



Red Hat Advanced Cluster Management for Kubernetes 2.10

설치

설치

설치

법적 공지

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

초록

연결된 네트워크 및 연결이 끊긴 네트워크, 설치 요구 사항 및 권장 사항, 다중 클러스터 고급 구성, 업그레이드 및 제거 방법에 대한 자세한 내용을 확인하십시오.

차례

1장. 설치 및 업그레이드	3
1.1. 성능 및 확장	3
1.2. 온라인으로 연결하는 동안 설치	14
1.3. 연결이 끊긴 네트워크 환경에 설치	19
1.4. MULTICLUSTERHUB 고급 구성	24
1.5. 업그레이드	30
1.6. 연결이 끊긴 네트워크 환경에서 업그레이드	31
1.7. 정책을 사용하여 연결이 끊긴 클러스터 업그레이드	33
1.8. 설치 제거	43

1장. 설치 및 업그레이드

Kubernetes용 Red Hat Advanced Cluster Management를 설치하면 다중 노드 클러스터 프로덕션 환경이 설정됩니다. Red Hat Advanced Cluster Management for Kubernetes를 표준 또는 고가용성 구성으로 설치할 수 있습니다.

설치하기 전에 각 제품에 필요한 하드웨어 및 시스템 구성을 검토하십시오. 지원되는 Red Hat OpenShift Container Platform 버전을 사용하여 Linux에 온라인을 설치할 수 있습니다.

지원되는 OpenShift Container Platform 버전이 있어야 합니다. 예를 들어 AWS에서 Red Hat OpenShift Service 또는 Red Hat OpenShift Dedicated를 사용할 수 있습니다. Kubernetes Operator용 다중 클러스터 엔진을 설치해야 합니다.

더 이상 사용되지 않음: Red Hat Advanced Cluster Management 2.7 및 이전 버전은 더 이상 지원되지 않습니다. 문서는 사용할 수 있지만 에라타 또는 기타 업데이트는 사용할 수 없습니다.

모범 사례: 최신 버전으로 업그레이드합니다.

FIPS 알림: **spec.ingress.sslCiphers** 에 자체 암호를 지정하지 않으면 **multiclusterhub-operator** 에서 기본 암호 목록을 제공합니다. FIPS 컴플라이언스를 업그레이드하고 사용하려면 멀티**clusterhub** 리소스에서 다음 두 가지 암호를 제거하십시오. **ECDHE-ECDSA-CHACHA20-POLY1305** 및 **ECDHE-RSA-CHACHA20-POLY1305**.

이 문서는 특정 구성 요소 또는 기능이 도입되어 최신 버전의 OpenShift Container Platform에서만 테스트되지 않는 한 가장 빨리 지원되는 OpenShift Container Platform 버전을 참조합니다.

전체 지원 정보는 [Red Hat Advanced Cluster Management for Kubernetes의 지원 매트릭스 및 라이프 사이클 및 업데이트 정책](#)을 참조하십시오.

설치 절차에 대한 자세한 내용은 다음 문서를 참조하십시오.

- [클러스터 크기 조정](#)
- [성능 및 확장](#)
- [온라인으로 연결하는 동안 설치](#)
- [연결이 끊긴 네트워크에 설치](#)
- [MultiClusterHub 고급 구성](#)
- [업그레이드](#)
- [연결이 끊긴 네트워크 환경에서 업그레이드](#)
- [정책을 사용하여 연결이 끊긴 클러스터 업그레이드](#)
- [설치 제거](#)

1.1. 성능 및 확장

Red Hat Advanced Cluster Management for Kubernetes는 특정 확장성 및 성능 데이터를 확인하기 위해 테스트되었습니다. 테스트된 주요 영역은 클러스터 확장성 및 검색 성능입니다. 환경을 계획할 때 이 정보를 사용할 수 있습니다.

참고: 데이터는 테스트 시 랩 환경의 결과를 기반으로 합니다.

Red Hat Advanced Cluster Management는 베어 메탈 머신에서 3개의 노드 허브 클러스터를 사용하여 테스트합니다. 테스트 시 소프트웨어 구성 요소 제한을 찾을 수 있는 충분한 리소스 용량(CPU, 메모리 및 디스크)이 있습니다. 환경, 네트워크 속도 및 제품 변경 사항에 따라 결과가 다를 수 있습니다.

- 최대 관리 클러스터 수
- 검색 확장성
- 관찰 기능 확장성
- 백업 및 복원 확장성

1.1.1. 최대 관리 클러스터 수

Red Hat Advanced Cluster Management에서 관리할 수 있는 최대 클러스터 수는 다음과 같은 여러 요인에 따라 다릅니다.

- 배포된 정책 및 애플리케이션 수와 같은 요인에 따라 달라지는 클러스터의 리소스 수입니다.
- 확장에 사용되는 Pod 수와 같은 허브 클러스터 구성.

관리형 클러스터는 Red Hat Enterprise Linux 하이퍼바이저에서 호스팅되는 단일 노드 OpenShift 가상 머신입니다. 가상 머신은 테스트된 상태에서 단일 베어 메탈 머신당 클러스터의 고밀도 수를 달성하는 데 사용됩니다. sushy-emulator는 Redfish API를 사용하여 가상 머신에 액세스할 수 있는 베어 메탈 클러스터를 보유하는 데 libvirt와 함께 사용됩니다. 다음 운영자는 테스트 설치, 토폴로지 Aware Lifecycle Manager, Local Storage Operator 및 Red Hat OpenShift GitOps의 일부입니다. 다음 표에서는 랩 환경 확장 정보를 보여줍니다.

표 1.1. 환경 스케일링 표

노드	수량	운영 체제	하드웨어	CPU 코어 수	메모리	디스크
hub 클러스터 컨트롤 플레인	3	OpenShift Container Platform	베어 메탈	112	512GiB	446GB SSD, 2.9 TB NVMe, 2 x 1.8TB SSD
관리형 클러스터	3500	단일 노드 OpenShift	가상 머신	8	18GiB	120GB

1.1.2. 검색 확장성

검색 구성 요소의 확장성은 데이터 저장소의 성능에 따라 다릅니다. 쿼리 런타임은 검색 성능을 분석할 때 중요한 변수입니다.

1.1.2.1. 쿼리 런타임 고려 사항

쿼리에서 결과를 실행하고 반환하는 데 걸리는 시간에 영향을 줄 수 있는 몇 가지 사항이 있습니다. 환경을 계획하고 구성할 때 다음 항목을 고려하십시오.

- 키워드 검색은 효율적이지 않습니다.
RedHat 을 검색하고 많은 수의 클러스터를 관리하는 경우 검색 결과를 수신하는 데 시간이 더 걸릴 수 있습니다.

- 첫 번째 검색은 사용자 역할 기반 액세스 제어 규칙을 수집하는 데 추가 시간이 걸리기 때문에 이후 검색보다 오래 걸립니다.
- 요청을 완료하는 데 걸리는 시간은 사용자가 액세스할 수 있는 네임스페이스 및 리소스 수에 비례합니다.
참고: 다른 사용자와 검색 쿼리를 저장하고 공유하는 경우 반환된 결과는 해당 사용자의 액세스 수준에 따라 달라집니다. 역할 액세스에 대한 자세한 내용은 [RBAC를 사용하여 OpenShift Container Platform 설명서에서 권한 정의 및 적용](#)을 참조하십시오.
- 가장 심각한 성능은 관리자가 아닌 사용자가 모든 네임스페이스 또는 모든 관리형 클러스터에 액세스할 수 있는 요청에 대해 관찰됩니다.

1.1.3. 관찰 기능 확장성

관찰 기능 서비스를 활성화하고 사용하려면 환경을 계획해야 합니다. 나중에 리소스 사용은 관찰 구성 요소가 설치된 OpenShift Container Platform 프로젝트에 사용됩니다. 사용하려는 값은 모든 관찰 기능 구성 요소의 합계입니다.

참고: 데이터는 테스트 시 랩 환경의 결과를 기반으로 합니다. 환경, 네트워크 속도 및 제품 변경 사항에 따라 결과가 다를 수 있습니다.

1.1.3.1. 샘플 관찰 기능 환경

샘플 환경에서 허브 클러스터 및 관리 클러스터는 Amazon Web Services 클라우드 플랫폼에 있으며 다음과 같은 토폴로지 및 구성이 있습니다.

노드	플레이버	vCPU	RAM(GiB)	디스크 유형	디스크 크기(GiB)	수량	리전
마스터 노드	m5.4xlarge	16	64	gp2	100	3	sa-east-1
작업자 노드	m5.4xlarge	16	64	gp2	100	3	sa-east-1

관찰 기능 배포는 고가용성 환경에 맞게 구성됩니다. 고가용성 환경에서 각 Kubernetes 배포에는 두 개의 인스턴스가 있으며 각 StatefulSet에는 세 개의 인스턴스가 있습니다.

샘플 테스트 중에 다양한 관리 클러스터 수가 메트릭을 푸시하도록 시뮬레이션되며 각 테스트는 24시간 동안 지속됩니다. 다음 처리량을 참조하십시오.

1.1.3.2. 쓰기 처리량

Pods	간격(분)	분당 시계열
400	1	83000

1.1.3.3. CPU 사용량 (millicores)

테스트 중에 CPU 사용량이 안정적입니다.

크기	CPU 사용량
10개의 클러스터	400
20개의 클러스터	800

1.1.3.4. RSS 및 작업 세트 메모리

RSS 및 작업 세트 메모리에 대한 다음 설명을 확인합니다.

- **메모리 사용량 RSS:** 지표 `container_memory_rss` 에서 테스트 중에 안정적으로 유지됩니다.
- **메모리 사용량 작업 세트:** `container_memory_working_set_bytes` 메트릭에서 테스트와 함께 증가합니다.

다음 결과는 24 시간 테스트에서 수행됩니다.

크기	메모리 사용량 RSS	메모리 사용량 작업 세트
10개의 클러스터	9.84	4.93
20개의 클러스터	13.10	8.76

1.1.3.5. thanos-receive 구성 요소의 영구 볼륨

중요: 지표는 보존 시간(24일)에 도달할 때까지 `thanos-receive` 에 저장됩니다. 다른 구성 요소에는 `thanos-receive` 구성 요소 만큼의 볼륨이 필요하지 않습니다.

테스트와 함께 디스크 사용량이 증가합니다. 데이터는 1일 후 디스크 사용량을 나타내므로 최종 디스크 사용량에 4를 곱합니다.

다음 디스크 사용량을 참조하십시오.

크기	디스크 사용량(GiB)
10개의 클러스터	2
20개의 클러스터	3

1.1.3.6. 네트워크 전송

테스트 중에 네트워크 전송은 안정성을 제공합니다. 크기 및 네트워크 전송 값을 참조하십시오.

크기	인바운드 네트워크 전송	아웃 바운드 네트워크 전송
10개의 클러스터	초당 6.55MBS	초당 5.80MBS

크기	인바운드 네트워크 전송	아웃 바운드 네트워크 전송
20개의 클러스터	13.08 MBS/초	초당 10.9MBS

1.1.3.7. Amazon S3(Simple Storage Service)

Amazon Simple Storage Service(S3)의 총 사용량이 증가합니다. 메트릭 데이터는 기본 보존 시간(five days)에 도달할 때까지 S3에 저장됩니다. 다음 디스크 사용량을 참조하십시오.

크기	디스크 사용량(GiB)
10개의 클러스터	16.2
20개의 클러스터	23.8

1.1.4. 백업 및 복원 확장성

대규모 확장 환경에서 수행되는 테스트에는 백업 및 복원을 위한 다음 데이터가 표시됩니다.

표 1.2. 관리되는 클러스터 백업의 실행 시간 표

백업	duration	리소스 수	백업 메모리
credentials	2m5s	18272 리소스	55MiB 백업 크기
관리형 클러스터	3m22s	58655 리소스	38MiB 백업 크기
resources	1m34s	1190 리소스	1.7MiB 백업 크기
일반/사용자	2m56s	0 리소스	16.5KiB 백업 크기

총 백업 시간은 **10m**입니다.

표 1.3. 패시브 허브 클러스터 복원을 위한 런타임 표

백업	duration	리소스 수
redentials	47m8s	18272 리소스
resources	3m10s	1190 리소스
일반/사용자 백업	0m	0 리소스

총 복원 시간은 **50m18s**입니다.

백업 파일 수는 **BackupSchedule** 이 생성될 때 **veleroTtl** 매개변수 옵션을 사용하여 정리됩니다. 지정된 TTL(Time to Live)보다 오래된 생성 시간이 있는 백업이 만료되고 Velero에 의해 스토리지 위치에서 자동으로 삭제됩니다.

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

1.1.5. 클러스터 크기 조정

각 Red Hat Advanced Cluster Management for Kubernetes 클러스터는 고유하며 다음 지침에 따라 샘플 배포 크기를 제공합니다. 권장 사항은 크기와 목적에 따라 분류됩니다.

Red Hat Advanced Cluster Management는 지원 서비스의 규모 조정 및 배치를 위해 다음과 같은 차원을 적용합니다.

- **가용성 영역**은 클러스터 전체에서 잠재적인 오류 도메인을 분리합니다. 일반적인 클러스터는 3개 이상의 가용성 영역에서 작업자 노드 용량이 거의 동일합니다.
- **vCPU 예약 및 제한**은 컨테이너에 할당할 작업자 노드에 vCPU 용량을 설정합니다. vCPU는 Kubernetes 컴퓨팅 단위와 동일합니다. 자세한 내용은 Kubernetes [Meaning of CPU](#) 를 참조하십시오.
- **메모리 예약 및 제한**은 컨테이너에 할당할 작업자 노드에 메모리 용량을 설정합니다.
- **영구 데이터**는 제품에 의해 관리되며 Kubernetes에서 사용하는 etcd 클러스터에 저장됩니다.

중요: OpenShift Container Platform의 경우 클러스터의 마스터 노드를 세 개의 가용성 영역에 배포합니다.

1.1.5.1. 제품 환경

참고: 다음 요구 사항은 최소 요구사항이 아닙니다.

표 1.4. 제품 환경

노드 유형	가용성 영역	etcd	총 예약된 메모리	총 예약된 CPU
Master	3	3	OpenShift Container Platform 크기 조정 지침	OpenShift Container Platform 크기 조정 지침
작업자 또는 인프라	3	1	12GB	6

Red Hat Advanced Cluster Management 외에도 OpenShift Container Platform 클러스터는 추가 서비스를 실행하여 클러스터 기능을 지원합니다. 자세한 내용은 [인프라 노드에 Red Hat Advanced Cluster Management Hub 클러스터 설치](#)를 참조하십시오.

1.1.5.1.1. 추가 서비스의 OpenShift Container Platform

가용성 영역은 클러스터 전체에서 잠재적인 오류 도메인을 분리합니다.

표 1.5. 추가 서비스

Service	노드 수	가용성 영역	인스턴스 크기	vCPU	메모리	스토리지 크기	Resources
Amazon Web Services의 OpenShift Container Platform	3	3	m5.xlarge	4	16GB	120GB	자세한 내용은 OpenShift Container Platform 제품 설명서 에서 사용자 지정으로 AWS에 클러스터 설치를 참조하십시오. 또한 머신 유형에 대해 자세히 알아보십시오.
Google Cloud Platform의 OpenShift Container Platform	3	3	N1-standard-4 (0.95-6.5GB)	4	15GB	120GB	할당량에 대한 자세한 내용은 Google Cloud Platform 제품 설명서 를 참조하십시오. 또한 머신 유형에 대해 자세히 알아보십시오.
Microsoft Azure의 OpenShift Container Platform	3	3	Standard_D4_v3	4	16GB	120GB	자세한 내용은 OpenShift Container Platform 설명서 에서 Azure 계정 구성을 참조하십시오.

Service	노드 수	가용성 영역	인스턴스 크기	vCPU	메모리	스토리지 크기	Resources
VMware vSphere의 OpenShift Container Platform	3	3		4 (소켓당 2개 코어)	16GB	120GB	자세한 내용은 다음 제품 설명서를 참조하십시오.
IBM Z 시스템의 OpenShift Container Platform	3	3		10	16GB	100GB	<p>자세한 내용은 OpenShift Container Platform 설명서에서 IBM Z 시스템에 클러스터 설치를 참조하십시오.</p> <p>IBM Z 시스템은 각 코어에서 실행할 수 있는 vCPU 수를 확장하는 SMT(동시 멀티스레딩)를 구성할 수 있는 기능을 제공합니다. SMT를 구성한 경우 하나의 물리적 코어 (IFL)는 두 개의 논리 코어(스레드)를 제공합니다. 하이퍼바이저는 두 개 이상의 vCPU를 제공할 수 있습니다.</p> <p>SMT(동시 멀티스레</p>

Service	노드 수	가용성 영역	인스턴스 크기	vCPU	메모리	스토리지 크기	당) 또는 Resource S 레딩이 활성화되지 않은 경우 하나의 vCPU는 하나의 물리적 코어와 동일합니다. 활성화하면 다음과 같은 공식을 사용하여 해당 비율을 계산합니다. (코어당 스레드 수 × 코어 수) × 소켓 수 = vCPU 수 SMT에 대한 자세한 내용은 Simultaneous multithreading 을 참조하십시오.
IBM Power 시스템의 OpenShift Container Platform	3	3		16	16GB	120GB	<p>자세한 내용은 OpenShift Container Platform 설명서에서 Power 시스템에 클러스터 설치를 참조하십시오.</p> <p>IBM Power 시스템은 각 코어에서 실행할 수 있는 vCPU 수를 확장하는</p>

Service	노드 수	가용성 영역	인스턴스 크기	vCPU	메모리	스토리지 크기	SMT(동시 멀티스레딩)를 구성할 수 있는 기능을 제공합니다. SMT를 구성한 경우 SMT 수준이 16 vCPU 요구 사항을 충족하는 방법을 결정합니다. 가장 일반적인 구성은 다음과 같습니다.
							<p>SMT-8(IBM Power VM을 실행하는 시스템의 기본 구성)에서 실행되는 두 코어는 필요한 16개의 vCPU를 제공합니다.</p> <p>SMT-4에서 실행되는 4개의 코어는 필요한 16개의 vCPU를 제공합니다.</p> <p>SMT에 대한 자세한 내용은 Simultaneous multithreading 을 참조하십시오.</p>

Service	노드 수	가용성 영역	인스턴스 크기	vCPU	메모리	스토리지 크기	Resources
OpenShift Container Platform 온-프레미스	3			4	16GB	120GB	<p>자세한 내용은 다음 제품 설명서를 참조하십시오.</p> <p>OpenShift Container Platform 베어 메탈에서 Red Hat Advanced Cluster Management for Kubernetes Hub 클러스터를 설치하고 지원할 수 있습니다. 허브 클러스터는 소형 베어 메탈 토폴로지에서 실행할 수 있으며, 여기에는 스케줄링 가능한 컨트롤 플레인 노드가 3개, 추가 작업자가 0개 있습니다.</p>

1.1.5.1.2. 단일 노드 OpenShift Container Platform 클러스터 생성 및 관리

요구 사항을 알아보려면 [단일 노드에](#) 설치를 확인합니다. 각 클러스터는 고유하므로 다음 지침은 크기와 목적에 따라 분류되는 샘플 배포 요구 사항만 제공합니다.

가용성 영역은 클러스터 전체에서 잠재적인 오류 도메인을 분리합니다. 일반적인 클러스터에는 3개 이상의 가용성 영역에 동일한 작업자 노드 용량이 있습니다. 고가용성은 지원되지 않습니다.

중요: OpenShift Container Platform의 경우 클러스터의 마스터 노드를 세 개의 가용성 영역에 배포합니다.

3500개의 단일 노드 OpenShift Container Platform 클러스터를 생성하고 관리하는 데 필요한 요구 사항의 예를 참조하십시오. Red Hat Advanced Cluster Management를 사용하여 단일 노드 OpenShift 클러스

터 (230 이상)를 생성하고 허브 클러스터를 사용하여 단일 노드 OpenShift 클러스터를 관리하는 데 필요한 최소 요구 사항을 참조하십시오.

표 1.6. 마스터(예약 가능)

노드 수	메모리(클러스터 사용량 사용)	메모리(단일 노드 min-max)	CPU 클러스터	CPU 단일 노드
3	289GB	64GB - 110GB	90	44

1.2. 온라인으로 연결하는 동안 설치

Red Hat Advanced Cluster Management Hub 클러스터를 포함하는 구성 요소의 설치, 업그레이드 및 제거를 관리하는 Operator Lifecycle Manager를 통해 Kubernetes용 Red Hat Advanced Cluster Management를 설치합니다.

필수 액세스: 클러스터 관리자. **OpenShift Container Platform Dedicated 환경에 필요한 액세스 권한:** **cluster-admin** 권한이 있어야 합니다. 기본적으로 **dedicated-admin** 역할에는 OpenShift Container Platform Dedicated 환경에서 네임스페이스를 생성하는 데 필요한 권한이 없습니다.

- 기본적으로 허브 클러스터 구성 요소는 추가 구성없이 OpenShift Container Platform 클러스터의 작업자 노드에 설치됩니다. OpenShift Container Platform OperatorHub 웹 콘솔 인터페이스를 사용하거나 OpenShift Container Platform CLI를 사용하여 작업자 노드에 허브 클러스터를 설치할 수 있습니다.
- 인프라 노드로 OpenShift Container Platform 클러스터를 구성한 경우 추가 리소스 매개변수와 함께 OpenShift Container Platform CLI를 사용하여 해당 인프라 노드에 hub 클러스터를 설치할 수 있습니다. 자세한 내용은 *인프라 노드에 Red Hat Advanced Cluster Management Hub 클러스터 설치* 섹션을 참조하십시오.
- OpenShift Container Platform 또는 Red Hat Advanced Cluster Management에서 생성되지 않은 Kubernetes 클러스터를 가져오려면 이미지 가져오기 보안을 구성해야 합니다.

고급 구성을 구성하는 방법에 대한 자세한 내용은 문서의 [MultiClusterHub 고급 구성](#) 섹션의 옵션을 참조하십시오.

- [사전 요구 사항](#)
- [OpenShift Container Platform 설치 확인](#)
- [OperatorHub 웹 콘솔 인터페이스에서 설치](#)
- [OpenShift Container Platform CLI에서 설치](#)
- [인프라 노드에 Red Hat Advanced Cluster Management Hub 클러스터 설치](#)

1.2.1. 사전 요구 사항

Red Hat Advanced Cluster Management를 설치하기 전에 다음 요구 사항을 참조하십시오.

- Red Hat OpenShift Container Platform 클러스터는 OpenShift Container Platform 콘솔의 OperatorHub 카탈로그에서 Red Hat Advanced Cluster Management Operator에 액세스할 수 있어야 합니다.
- catalog.redhat.com 에 액세스해야 합니다.

- OpenShift Container Platform 버전 4.13 이상을 환경에 배포해야 하며 OpenShift Container Platform CLI로 로그인해야 합니다. OpenShift Container Platform에 대한 다음 설치 설명서를 확인하고 필요한 경우 이전 버전으로 변경합니다. [OpenShift Container Platform 설치](#)
- **oc** 명령을 실행하도록 OpenShift Container Platform CLI(명령줄 인터페이스)를 구성해야 합니다. OpenShift Container Platform [CLI 설치 및 구성에 대한 정보는 CLI로 시작하기](#) 를 참조하십시오.
- OpenShift Container Platform 권한에서 네임스페이스를 생성할 수 있어야 합니다. 네임스페이스가 없으면 설치에 실패합니다.
- Operator의 종속 항목에 액세스하려면 인터넷 연결이 있어야 합니다.
- **중요:** OpenShift Container Platform Dedicated 환경에 설치하려면 다음 요구 사항을 참조하십시오.
 - OpenShift Container Platform Dedicated 환경이 구성되어 실행되고 있어야 합니다.
 - 허브 클러스터를 설치하는 OpenShift Container Platform Dedicated 환경에 대한 **cluster-admin** 권한이 있어야 합니다.
 - 가져오려면 kubernetes operator for 2.10의 **stable-2.0** 채널을 사용해야 합니다.

1.2.2. OpenShift Container Platform 설치 확인

레지스트리 및 스토리지 서비스를 포함하여 지원되는 OpenShift Container Platform 버전이 설치되어 작동해야 합니다. OpenShift Container Platform 설치에 대한 자세한 내용은 OpenShift Container Platform 설명서를 참조하십시오.

1. Red Hat Advanced Cluster Management Hub 클러스터가 OpenShift Container Platform 클러스터에 설치되어 있지 않은지 확인합니다. Red Hat Advanced Cluster Management는 각 OpenShift Container Platform 클러스터에 하나의 Red Hat Advanced Cluster Management 허브 클러스터 설치만 허용합니다. Red Hat Advanced Cluster Management Hub 클러스터가 설치되지 않은 경우 다음 단계를 계속 진행합니다.
2. OpenShift Container Platform 클러스터가 올바르게 설정되었는지 확인하려면 다음 명령을 사용하여 OpenShift Container Platform 웹 콘솔에 액세스합니다.

```
kubectl -n openshift-console get route
```

다음 예제 출력을 참조하십시오.

```
openshift-console console console-openshift-console.apps.new-coral.purple-chesterfield.com
console https reencrypt/Redirect None
```

3. 브라우저에서 URL을 열고 결과를 확인합니다. 콘솔 URL에 **console-openshift-console.router.default.svc.cluster.local** 이 표시되면 OpenShift Container Platform을 설치할 때 **openshift_master_default_subdomain** 값을 설정합니다. URL의 다음 예제를 참조하십시오. <https://console-openshift-console.apps.new-coral.purple-chesterfield.com>.

콘솔 또는 CLI에서 Red Hat Advanced Cluster Management를 설치할 수 있습니다. 두 절차가 문서화되어 있습니다.

1.2.3. OperatorHub 웹 콘솔 인터페이스에서 설치

모범 사례: OpenShift Container Platform 탐색의 *관리자* 보기에서 OpenShift Container Platform과 함께 제공되는 OperatorHub 웹 콘솔 인터페이스를 설치합니다.

1. **Operators > OperatorHub** 를 선택하여 사용 가능한 Operator 목록에 액세스하고 *Kubernetes Operator용 Advanced Cluster Management* 를 선택합니다.
2. *Operator 서브스크립션* 페이지에서 설치 옵션을 선택합니다.
 - 네임스페이스 정보:
 - Red Hat Advanced Cluster Management Hub 클러스터는 자체 네임스페이스 또는 프로젝트에 설치해야 합니다.
 - 기본적으로 OperatorHub 콘솔 설치 프로세스는 **open-cluster-management** 라는 네임스페이스를 생성합니다. **모범 사례:** 사용 가능한 경우 **open-cluster-management** 네임스페이스를 계속 사용합니다.
 - **open-cluster-management** 라는 네임스페이스가 이미 있는 경우 다른 네임스페이스를 선택합니다.
 - 채널: 선택한 채널은 설치 중인 릴리스에 해당합니다. 채널을 선택하면 식별된 릴리스를 설치하고 해당 릴리스 내에서 향후 에라타 업데이트를 수행하도록 설정합니다.
 - 업데이트 승인 전략: 승인 전략은 서브스크립션을 구독한 채널 또는 릴리스에 업데이트를 적용하는 데 필요한 인적 상호 작용을 식별합니다.
 - **Automatic** 을 선택하여 해당 릴리스 내의 업데이트가 자동으로 적용되었는지 확인합니다.
 - 업데이트를 사용할 수 있을 때 알림을 받으려면 **수동** 을 선택합니다. 업데이트가 적용되는 시기에 대해 우려 사항이 있는 경우 이 방법을 사용하는 것이 좋습니다.

중요: 다음 마이너 릴리스로 업그레이드하려면 *OperatorHub* 페이지로 돌아가서 최신 릴리스의 새 채널을 선택해야 합니다.

3. **설치** 를 선택하여 변경 사항을 적용하고 Operator를 생성합니다.
4. *MultiClusterHub* 사용자 정의 리소스를 생성합니다.
 - a. OpenShift Container Platform 콘솔 탐색에서 **Installed Operators > Advanced Cluster Management for Kubernetes** 를 선택합니다.
 - b. **MultiClusterHub** 탭을 선택합니다.
 - c. **Create MultiClusterHub** 를 선택합니다.
 - d. YAML 파일에서 기본값을 업데이트합니다. 문서의 *MultiClusterHub 고급 구성* 섹션의 옵션을 참조하십시오.
 - 다음 예제에서는 기본 템플릿을 보여줍니다. **네임스페이스가 프로젝트 네임스페이스** 인지 확인합니다. 예제를 참조하십시오.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
```

5. **생성** 을 선택하여 사용자 지정 리소스를 초기화합니다. Red Hat Advanced Cluster Management Hub 클러스터를 빌드하고 시작하는 데 최대 10분이 걸릴 수 있습니다.
Red Hat Advanced Cluster Management Hub 클러스터가 생성되면 **MultiClusterHub** 리소스 상태가 Red Hat Advanced Cluster Management Operator의 *MultiClusterHub* 탭에서 *Running* 을 표시합니다. 콘솔에 액세스하려면 콘솔 [액세스 주제](#)를 참조하십시오.

1.2.4. OpenShift Container Platform CLI에서 설치

1. Operator 요구 사항이 포함된 Red Hat Advanced Cluster Management Hub 클러스터 네임스페이스를 생성합니다. 다음 명령을 실행합니다. 여기서 **namespace** 는 Red Hat Advanced Cluster Management hub 클러스터 네임스페이스의 이름입니다. **namespace** 값은 OpenShift Container Platform 환경에서 *Project* 라고 할 수 있습니다.

```
oc create namespace <namespace>
```

2. 프로젝트 네임스페이스를 생성한 네임스페이스로 전환합니다. **namespace** 를 1단계에서 생성한 Red Hat Advanced Cluster Management Hub 클러스터 네임스페이스의 이름으로 교체합니다.

```
oc project <namespace>
```

3. YAML 파일을 생성하여 **OperatorGroup** 리소스를 구성합니다. 각 네임스페이스에는 하나의 operator 그룹만 있을 수 있습니다. **default** 를 operator 그룹의 이름으로 바꿉니다. **namespace** 를 프로젝트 네임스페이스의 이름으로 교체합니다. 다음 샘플을 참조하십시오.

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <default>
  namespace: <namespace>
spec:
  targetNamespaces:
  - <namespace>
```

4. 다음 명령을 실행하여 **OperatorGroup** 리소스를 생성합니다. **operator-group** 을 생성한 operator 그룹 YAML 파일의 이름으로 교체합니다.

```
oc apply -f <path-to-file>/<operator-group>.yaml
```

5. YAML 파일을 생성하여 OpenShift Container Platform 서브스크립션을 구성합니다. 파일은 다음 샘플과 유사하며 **release-2.x** 를 현재 릴리스로 교체합니다.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: acm-operator-subscription
spec:
  sourceNamespace: openshift-marketplace
  source: redhat-operators
  channel: release-2.x
  installPlanApproval: Automatic
  name: advanced-cluster-management
```

참고: 인프라 노드에 Red Hat Advanced Cluster Management Hub 클러스터를 설치하려면 [Operator Lifecycle Manager 서브스크립션 추가 구성](#) 섹션을 참조하십시오.

- 다음 명령을 실행하여 OpenShift Container Platform 서브스크립션을 생성합니다. 서브스크립션을 생성한 서브스크립션 파일의 이름으로 교체합니다.

```
oc apply -f <path-to-file>/<subscription>.yaml
```

- YAML 파일을 생성하여 **MultiClusterHub** 사용자 정의 리소스를 구성합니다. 기본 템플릿은 다음 예와 유사해야 합니다. **namespace** 를 프로젝트 네임스페이스의 이름으로 교체합니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec: {}
```

참고: 인프라 노드에 Red Hat Advanced Cluster Management Hub 클러스터를 설치하려면 [MultiClusterHub 사용자 정의 리소스 추가 구성](#) 섹션을 참조하십시오.

- 다음 명령을 실행하여 **MultiClusterHub** 사용자 정의 리소스를 생성합니다. **custom-resource** 를 사용자 정의 리소스 파일의 이름으로 교체합니다.

```
oc apply -f <path-to-file>/<custom-resource>.yaml
```

다음 오류와 함께 이 단계가 실패하면 리소스가 계속 생성되고 적용됩니다. 리소스가 생성될 때 몇 분 후에 명령을 다시 실행합니다.

```
error: unable to recognize "/mch.yaml": no matches for kind "MultiClusterHub" in version "operator.open-cluster-management.io/v1"
```

- 다음 명령을 실행하여 사용자 정의 리소스를 가져옵니다. **MultiClusterHub** 사용자 정의 리소스 상태가 명령을 실행한 후 **status.phase** 필드에 **Running** 으로 표시되는 데 최대 10분이 걸릴 수 있습니다.

```
oc get mch -o=jsonpath='{.items[0].status.phase}'
```

Red Hat Advanced Cluster Management를 다시 설치하고 Pod가 시작되지 않는 경우 이 문제를 해결하기 위한 단계의 [설치 제거 실패 해결](#)을 참조하십시오.

참고:

- ClusterRoleBinding** 이 있는 **ServiceAccount** 는 Red Hat Advanced Cluster Management 및 Red Hat Advanced Cluster Management를 설치하는 네임스페이스에 대한 액세스 권한이 있는 모든 사용자 인증 정보에 자동으로 클러스터 관리자 권한을 부여합니다.
- 설치 시 Red Hat Advanced Cluster Management Hub 클러스터용으로 예약된 **local-cluster** 라는 네임스페이스도 생성합니다. **local-cluster** 라는 기존 네임스페이스는 있을 수 없습니다. 보안상의 이유로 **cluster-administrator** 액세스 권한이 없는 사용자에게 **local-cluster** 네임스페이스에 대한 액세스를 릴리스하지 마십시오.

1.2.5. 인프라 노드에 Red Hat Advanced Cluster Management Hub 클러스터 설치

OpenShift Container Platform 클러스터는 승인된 관리 구성 요소를 실행하기 위한 인프라 노드를 포함하도록 구성할 수 있습니다. 인프라 노드에서 구성 요소를 실행하면 해당 관리 구성 요소를 실행하는 노드에 OpenShift Container Platform 서브스크립션 할당량이 할당되지 않습니다.

OpenShift Container Platform 클러스터에 인프라 노드를 추가한 후 [OpenShift Container Platform CLI](#)에서 설치 지침을 따르고 Operator Lifecycle Manager 서브스크립션 및 **MultiClusterHub** 사용자 정의 리소스에 구성을 추가합니다.

1.2.5.1. OpenShift Container Platform 클러스터에 인프라 노드 추가

OpenShift Container Platform 설명서에서 [인프라 머신 세트 생성](#)에 설명된 절차를 따르십시오. 인프라 노드는 관리되지 않은 워크로드가 해당 워크로드가 실행되지 않도록 Kubernetes **taint** 및 **레이블** 로 구성됩니다.

Red Hat Advanced Cluster Management에서 제공하는 인프라 노드 활성화와 호환되려면 인프라 노드에 다음과 같은 **테인트** 및 **라벨**이 적용되어야 합니다.

```
metadata:
  labels:
    node-role.kubernetes.io/infra: ""
spec:
  taints:
  - effect: NoSchedule
    key: node-role.kubernetes.io/infra
```

1.2.5.2. Operator Lifecycle Manager 서브스크립션 추가 구성

Operator Lifecycle Manager 서브스크립션을 적용하기 전에 다음 추가 구성을 추가합니다.

```
spec:
  config:
    nodeSelector:
      node-role.kubernetes.io/infra: ""
    tolerations:
    - key: node-role.kubernetes.io/infra
      effect: NoSchedule
      operator: Exists
```

1.2.5.3. MultiClusterHub 사용자 정의 리소스 추가 구성

MultiClusterHub 사용자 정의 리소스를 적용하기 전에 다음 추가 구성을 추가합니다.

```
spec:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
```

1.3. 연결이 끊긴 네트워크 환경에 설치

연결이 끊긴 Red Hat OpenShift Container Platform 클러스터에 Red Hat Advanced Cluster Management for Kubernetes를 설치해야 할 수 있습니다. 연결이 끊긴 허브 클러스터에 설치하려면 연결된 네트워크 환경에 해당하는 일반적인 설치 또는 업그레이드 단계 외에도 다음 단계를 수행합니다.

필수 액세스: 모든 설치 및 업그레이드 작업에 대한 *클러스터 관리* 액세스 권한이 필요합니다.

다음 섹션을 참조하십시오.

- [사전 요구 사항](#)

- [OpenShift Container Platform 설치 확인](#)
- [Operator Lifecycle Manager 구성](#)
- [이미지 콘텐츠 소스 정책 구성](#)
- [Kubernetes Operator 및 허브용 Red Hat Advanced Cluster Management 설치](#)

1.3.1. 사전 요구 사항

Red Hat Advanced Cluster Management for Kubernetes를 설치하기 전에 다음 요구 사항을 충족해야 합니다.

- 연결이 끊긴 네트워크 환경에 설치되므로 미러링된 Operator Lifecycle Manager 카탈로그 및 Operator 이미지를 저장하려면 로컬 이미지 레지스트리에 액세스해야 합니다. 이 환경에 OpenShift Container Platform 클러스터를 설치할 때 로컬 이미지 레지스트리를 이미 설정했기 때문에 동일한 로컬 이미지 레지스트리를 사용할 수 있어야 합니다.
- 인터넷과 로컬 미러 레지스트리에 모두 액세스할 수 있는 워크스테이션이 있어야 합니다.
- 지원되는 Red Hat OpenShift Container Platform 버전은 환경에 배포해야 하며 CLI(명령줄 인터페이스)로 로그인해야 합니다. Red Hat [OpenShift Container Platform 설치에 대한 자세한 내용은 OpenShift Container Platform 설치 설명서](#) 를 참조하십시오. Red Hat OpenShift CLI를 사용하여 **oc** 명령 설치 및 구성에 대한 정보는 [CLI로 시작하기](#) 를 참조하십시오.
- 클러스터 크기를 검토하여 허브 클러스터의 용량 설정에 대해 알아봅니다.

1.3.2. OpenShift Container Platform 설치 확인

- 연결되어 있는 동안 **oc -n openshift-console get route** 명령을 실행하여 OpenShift Container Platform 웹 콘솔에 액세스합니다. 다음 예제 출력을 참조하십시오.

```
openshift-console      console      console-openshift-console.apps.new-coral.purple-
chesterfield.com      console      https reencrypt/Redirect  None
```

브라우저에서 URL을 열고 결과를 확인합니다. 콘솔 URL에 **console-openshift-console.router.default.svc.cluster.local** 이 표시되면 OpenShift Container Platform을 설치할 때 **openshift_master_default_subdomain** 값을 설정합니다.

1.3.3. 로컬 이미지 레지스트리의 가용성 확인

모범 사례: Operator Lifecycle Manager Operator 관련 콘텐츠에 기존 미러 레지스트리를 사용합니다.

연결이 끊긴 환경에서 Red Hat Advanced Cluster Management for Kubernetes를 설치하려면 로컬 미러 이미지 레지스트리를 사용해야 합니다. 연결이 끊긴 환경에서 OpenShift Container Platform 클러스터 설치를 이미 완료했기 때문에 Red Hat OpenShift Container Platform 클러스터 설치 중에 사용할 미러 레지스트리를 이미 설정했습니다.

로컬 이미지 레지스트리가 아직 없는 경우 Red Hat OpenShift Container Platform 설명서의 [연결이 끊긴 설치를 위해 이미지 미러링](#) 에 설명된 절차를 완료하여 생성합니다.

1.3.4. Operator Lifecycle Manager 구성

Red Hat Advanced Cluster Management for Kubernetes는 Operator로 패키지되므로 Operator Lifecycle Manager를 사용하여 설치가 완료됩니다.

연결이 끊긴 환경에서 Operator Lifecycle Manager는 연결이 끊긴 클러스터에서 액세스할 수 없는 이미지 레지스트리에서 호스팅되므로 Red Hat에서 제공하는 표준 Operator 소스에 액세스할 수 없습니다. 대신 클러스터 관리자는 미리링된 이미지 레지스트리 및 Operator 카탈로그를 사용하여 연결이 끊긴 환경에서 Operator를 설치하고 업그레이드할 수 있습니다.

Red Hat Advanced Cluster Management for Kubernetes 설치를 위한 연결이 끊긴 클러스터를 준비하려면 OpenShift Container Platform 설명서의 [제한된 네트워크에서 Operator Lifecycle Manager](#) 사용에 설명된 절차를 따르십시오.

1.3.4.1. 추가 요구 사항

이전 절차를 완료하면 Kubernetes용 Red Hat Advanced Cluster Management에도 해당하는 다음 요구 사항을 확인하십시오.

1.3.4.1.1. 미리 카탈로그에 Operator 패키지 포함

- 미리 카탈로그에 필요한 Operator 패키지를 포함합니다. Red Hat은 Red Hat Operator 카탈로그에 Red Hat Advanced Cluster Management for Kubernetes operator를 제공합니다. **registry.redhat.io/redhat/redhat-operator-index** 인덱스 이미지에서 제공합니다. 이 카탈로그 인덱스 이미지의 미리를 준비할 때 Red Hat에서 제공하는 전체 카탈로그를 미리링하거나 사용하려는 Operator 패키지만 포함된 하위 집합을 미리링할 수 있습니다. 전체 미리 카탈로그를 생성하는 경우 Red Hat Advanced Cluster Management for Kubernetes를 설치하는 데 필요한 모든 패키지가 포함되어 있으므로 특별한 고려 사항이 필요하지 않습니다. 그러나 부분적으로 또는 필터링된 미리링된 카탈로그를 생성하는 경우 특정 패키지를 식별하려면 목록에 다음 패키지 이름을 포함해야 합니다.
 - **advanced-cluster-management**
 - **multicluster-engine**
- 두 미리링 절차 중 하나를 사용합니다.
- OPM 유틸리티를 사용하여 미리링된 카탈로그 또는 레지스트리를 생성하는 경우 **opm index prune**에서는 다음 예에 표시된 대로 다음의 패키지 이름을 **-p** 옵션 값에 포함합니다. 현재 버전은 **4.x**를 대체합니다.

```
opm index prune \
  -f registry.redhat.io/redhat/redhat-operator-index:v4.x \
  -p advanced-cluster-management,multicluster-engine \
  -t myregistry.example.com:5000/mirror/my-operator-index:v4.x
```

- 대신 **oc-mirror** 플러그인을 사용하여 미리링된 카탈로그 또는 레지스트리를 채우는 경우 다음 예에 표시된 대로 **ImageSetConfiguration**의 패키지 목록 섹션에 다음 패키지 이름을 현재 **4.x**를 교체한 상태로 포함합니다.

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: myregistry.example.com:5000/mirror/oc-mirror-metadata
  mirror:
    platform:
      channels:
        - name: stable-4.x
      type: ocp
```

```

operators:
- catalog: registry.redhat.io/redhat/redhat-operator-index:v4.13
packages:
- name: advanced-cluster-management
- name: multicloud-engine
additionalImages: []
helm: {}

```

1.3.4.1.2. 미리 레지스트리를 사용하도록 구성

Red Hat Advanced Cluster Management for Kubernetes를 설치하는 데 필요한 이전 패키지로 로컬 미리 레지스트리를 채운 경우 [제한된 네트워크에서 Operator Lifecycle Manager를 사용하여](#) 연결이 끊긴 클러스터에서 미리 레지스트리 및 카탈로그를 사용할 수 있도록 하는 항목에 설명된 단계를 완료합니다.

1. 기본 OperatorHub 소스 비활성화
2. Operator 카탈로그 미러링
3. 미러링된 카탈로그의 카탈로그 소스 추가

1.3.4.1.3. 카탈로그 소스 이름 찾기

Red Hat OpenShift Container Platform 설명서의 절차에 설명된 대로 연결이 끊긴 클러스터에 **CatalogSource** 리소스를 추가해야 합니다. **중요:** 나중에 필요한 **metadata.name** 필드의 값을 기록해 둡니다.

다음 예와 유사한 YAML 파일을 사용하여 **CatalogSource** 리소스를 **openshift-marketplace** 네임스페이스에 추가하여 **4.x**를 현재 버전으로 교체합니다.

```

apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: my-mirror-catalog-source
  namespace: openshift-marketplace
spec:
  image: myregistry.example.com:5000/mirror/my-operator-index:v4.x
  sourceType: grpc

```

나중에 생성할 **MulticloudHub** 리소스에서 주식의 **metadata.name** 필드 값이 필요합니다.

1.3.5. 필수 패키지를 사용할 수 있는지 확인

Operator Lifecycle Manager는 정기적으로 사용 가능한 패키지의 카탈로그 소스를 폴링합니다. Operator Lifecycle Manager에서 미러링된 카탈로그의 카탈로그 소스를 폴링한 후 사용 가능한 **PackageManifest** 리소스를 쿼리하여 연결이 끊긴 클러스터에서 필요한 패키지를 사용할 수 있는지 확인할 수 있습니다.

연결이 끊긴 클러스터를 대상으로 다음 명령을 실행합니다.

```
oc -n openshift-marketplace get packagemanifests
```

표시되는 목록에는 미리 카탈로그의 카탈로그 소스에서 다음 패키지를 제공함을 보여주는 항목이 포함되어야 합니다.

- **advanced-cluster-management**

- multicluster-engine

1.3.6. 이미지 콘텐츠 소스 정책 구성

클러스터가 인터넷 호스팅 레지스트리가 아닌 미러 레지스트리에서 Kubernetes Operator용 Red Hat Advanced Cluster Management의 컨테이너 이미지를 가져오려면 미러 레지스트리에 이미지 참조를 리디렉션하도록 연결이 끊긴 클러스터에서 **ImageContentSourcePolicy** 를 구성해야 합니다.

oc adm catalog mirror 명령을 사용하여 카탈로그를 미러링한 경우 필요한 이미지 콘텐츠 소스 정책 구성은 해당 명령으로 생성된 **manifests-*** 디렉터리 내부의 **imageContentSourcePolicy.yaml** 파일에 있습니다.

oc-mirror 플러그인을 사용하여 카탈로그를 미러링한 경우 **imageContentSourcePolicy.yaml** 파일은 **oc-mirror-workspace/results-*** 디렉터리 create by the oc-mirror-workspace/results-* 디렉터리에 있습니다.

두 경우 모두 다음과 같은 **oc apply** 또는 **oc replace** 명령을 사용하여 연결이 끊긴 명령에 정책을 적용할 수 있습니다.

```
oc replace -f ./<path>/imageContentSourcePolicy.yaml
```

필요한 이미지 콘텐츠 소스 정책 설명은 미러 레지스트리를 생성한 방법에 따라 다를 수 있지만 다음 예와 유사합니다.

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  labels:
    operators.openshift.org/catalog: "true"
  name: operator-0
spec:
  repositoryDigestMirrors:
  - mirrors:
    - myregistry.example.com:5000/rhacm2
    source: registry.redhat.io/rhacm2
  - mirrors:
    - myregistry.example.com:5000/multicluster-engine
    source: registry.redhat.io/multicluster-engine
  - mirrors:
    - myregistry.example.com:5000/openshift4
    source: registry.redhat.io/openshift4
  - mirrors:
    - myregistry.example.com:5000/redhat
    source: registry.redhat.io/redhat
```

1.3.7. Kubernetes Operator 및 허브 클러스터용 Red Hat Advanced Cluster Management 설치

앞에서 설명한 대로 Operator Lifecycle Manager 및 Red Hat OpenShift Container Platform을 구성한 후 OperatorHub 콘솔 또는 CLI를 사용하여 Red Hat Advanced Cluster Management for Kubernetes를 설치할 수 있습니다. [연결된 온라인 주제에서 설치하는 동안](#) 설명하는 것과 동일한 지침을 따릅니다.

중요: MulticlusterHub 리소스 생성은 허브 클러스터의 설치 프로세스 시작입니다.

클러스터에 Operator를 설치하려면 미리 카탈로그에 기본이 아닌 카탈로그 소스를 사용해야 하므로 Operator에 미리 카탈로그 소스의 이름을 제공하기 위해 **MulticlusterHub** 리소스에 특수 주석이 필요합니다. 다음 예제는 필요한 **mce-subscription-spec** 주석을 표시합니다.

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

다중 클러스터 엔진 Operator가 Red Hat Advanced Cluster Management 설치 중에 자동으로 설치되므로 **mce-subscription-spec** 주석이 필요합니다. CLI를 사용하여 리소스를 생성하는 경우 **oc apply** 명령을 사용하여 적용하는 YAML에 **mce-subscription-spec** 주석을 포함하여 **MulticlusterHub** 리소스를 생성합니다.

OperatorHub 콘솔을 사용하여 리소스를 생성하는 경우 *YAML 보기*로 전환하고 이전에 표시된 대로 주석을 삽입합니다. **중요: MulticlusterHub** 를 생성하기 위해 *필드 보기* 패널의 주석에 OperatorHub 콘솔에는 필드가 없습니다.

1.4. MULTICLUSTERHUB 고급 구성

Red Hat Advanced Cluster Management for Kubernetes는 필요한 모든 구성 요소를 배포하는 Operator를 사용하여 설치됩니다. 나열된 일부 구성 요소는 기본적으로 활성화되어 있습니다. 구성 요소가 *비활성화* 된 경우 해당 리소스가 활성화될 때까지 클러스터에 배포되지 않습니다. Operator는 다음 구성 요소를 배포하기 위해 작동합니다.

표 1.7. 배포된 구성 요소의 표 목록

이름	설명	활성화됨
app-lifecycle	애플리케이션 및 애플리케이션 업데이트를 구성하고 배포하기 위한 옵션을 통합하고 단순화합니다.	True
cluster-backup	관리 클러스터, 애플리케이션 및 정책과 같은 모든 허브 클러스터 리소스에 대한 백업 및 복원 지원을 제공합니다.	False
cluster-lifecycle	OpenShift Container Platform 및 Red Hat Advanced Cluster Management Hub 클러스터를 위한 클러스터 관리 기능을 제공합니다.	True
cluster-permission	RBAC 리소스를 관리 클러스터에 자동으로 배포하고 해당 리소스의 라이프사이클을 관리합니다.	True

콘솔	Red Hat Advanced Cluster Management 웹 콘솔 플러그인을 활성화합니다.	True
grc	클러스터 정책을 정의할 수 있도록 보안 개선 사항을 활성화합니다.	True
Insights	클러스터의 기존 또는 잠재적 문제를 식별합니다.	True
multicluster-observability	모니터링으로 관리되는 클러스터의 상태에 대한 추가 정보를 얻을 수 있습니다.	True
search	모든 클러스터에서 Kubernetes 리소스를 시각화할 수 있습니다.	True
submariner-addon	온-프레미스 또는 클라우드에 있는 두 개 이상의 관리 클러스터 간에 직접 네트워킹 및 서비스 검색을 활성화합니다.	True
volsync	클러스터 내에서 또는 복제와 호환되지 않는 스토리지 유형이 있는 클러스터에서 비동기식 영구 볼륨 복제를 지원합니다.	True

Red Hat Advanced Cluster Management를 클러스터에 설치하는 경우 나열된 모든 구성 요소가 기본적으로 활성화되어 있는 것은 아닙니다.

MultiClusterHub 사용자 정의 리소스에 하나 이상의 속성을 추가하여 설치 중 또는 설치 후 Red Hat Advanced Cluster Management를 추가로 구성할 수 있습니다. 추가할 수 있는 속성에 대한 정보를 계속 읽습니다.

1.4.1. 콘솔 및 구성 요소 구성

다음 예제는 구성 요소를 활성화하거나 비활성화하는 데 사용할 수 있는 **spec.overrides** 기본 템플릿을 표시합니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace> ❶
spec:
  overrides:
    components:
      - name: <name> ❷
        enabled: true
```

1. **namespace** 를 프로젝트 이름으로 교체합니다.
2. **name** 을 구성 요소의 이름으로 바꿉니다.

또는 다음 명령을 실행할 수 있습니다. **namespace** 를 프로젝트 및 **name** 의 이름으로 구성 요소 이름으로 교체합니다.

```
oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p='[{"op": "add", "path": "/spec/overrides/components/-", "value": {"name": "<name>", "enabled": true}}]'
```

참고: 콘솔 구성 요소가 비활성화되면 Red Hat OpenShift Container Platform 콘솔이 비활성화됩니다.

1.4.2. 사용자 정의 이미지 가져오기 시크릿

OpenShift Container Platform 또는 Red Hat Advanced Cluster Management에서 생성하지 않은 Kubernetes 클러스터를 가져오려면 OpenShift Container Platform 풀 시크릿 정보가 있는 시크릿을 생성하여 배포 레지스트리에서 액세스할 수 있는 콘텐츠에 액세스합니다.

OpenShift Container Platform 클러스터의 시크릿 요구 사항은 OpenShift Container Platform 및 Red Hat Advanced Cluster Management에서 자동으로 해결되므로 관리할 다른 유형의 Kubernetes 클러스터를 가져오지 않는 경우 시크릿을 생성할 필요가 없습니다. OpenShift Container Platform 풀 시크릿은 Red Hat Customer Portal ID와 연결되며 모든 Kubernetes 공급자에서 동일합니다.

중요: 이러한 시크릿은 네임스페이스에 고유하므로 hub 클러스터에 사용하는 네임스페이스에 있는지 확인합니다.

1. cloud.redhat.com/openshift/install/pull-secret 으로 이동하여 OpenShift Container Platform 풀 시크릿 파일을 다운로드합니다.
2. 풀 시크릿 다운로드를 클릭합니다.
3. 다음 명령을 실행하여 보안을 생성합니다.

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

- **secret** 을 생성하려는 시크릿 이름으로 교체합니다.
- 시크릿은 **네임스페이스** 에 따라 네임스페이스를 사용하므로 네임스페이스를 프로젝트 네임스페이스로 교체합니다.
- 다운로드한 OpenShift Container Platform 풀 시크릿의 경로로 **path-to-pull-secret** 을 교체합니다.

다음 예제는 사용자 정의 풀 시크릿을 사용하려는 경우 사용할 **spec.imagePullSecret** 템플릿을 표시합니다. **secret** 을 풀 시크릿 이름으로 교체합니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  imagePullSecret: <secret>
```

1.4.3. availabilityConfig

Red Hat Advanced Cluster Management Hub 클러스터에는 두 가지 가용성, 즉 **High** 및 **Basic** 이 있습니다. 기본적으로 허브 클러스터의 가용성은 **High** 이며 hub 클러스터 구성 요소에 **2**의 **replicaCount** 를 제공합니다. 이는 장애 조치의 경우 더 나은 지원을 제공하지만 **기본** 가용성보다 많은 리소스를 사용하므로 구성 요소에 **1**의 **replicaCount** 가 제공됩니다.

중요: 단일 노드 OpenShift 클러스터에서 다중 클러스터 엔진 Operator를 사용하는 경우 **spec.availabilityConfig** 를 **Basic** 으로 설정합니다.

다음 예제에서는 **기본** 가용성이 있는 **spec.availabilityConfig** 템플릿을 보여줍니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  availabilityConfig: "Basic"
```

1.4.4. nodeSelector

Red Hat Advanced Cluster Management Hub 클러스터에서 노드 선택기 세트를 정의하여 클러스터의 특정 노드에 설치할 수 있습니다. 다음 예제에서는 **node-role.kubernetes.io/infra** 라벨이 있는 노드에 Red Hat Advanced Cluster Management Pod를 할당하는 **spec.nodeSelector** 를 보여줍니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
```

1.4.5. 허용 오차

Red Hat Advanced Cluster Management Hub 클러스터가 클러스터에 정의된 특정 테인트를 허용할 수 있도록 허용 오차 목록을 정의할 수 있습니다.

다음 예제는 **node-role.kubernetes.io/infra** 테인트와 일치하는 **spec.tolerations** 를 보여줍니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  tolerations:
    - key: node-role.kubernetes.io/infra
      effect: NoSchedule
      operator: Exists
```

이전 infra-node 허용 오차는 구성에 허용 오차를 지정하지 않고 기본적으로 Pod에 설정됩니다. 구성에서 허용 오차를 사용자 정의하면 이 기본값이 대체됩니다.

1.4.6. disableHubSelfManagement

기본적으로 Red Hat Advanced Cluster Management Hub 클러스터는 자동으로 가져와서 관리합니다. 이 관리 허브 클러스터의 이름은 **local-cluster** 입니다. hub 클러스터가 자체적으로 관리되는지 여부를 지정하는 설정은 **multiclusterengine** 사용자 정의 리소스에 있습니다. Red Hat Advanced Cluster Management에서 이 설정을 변경하면 **multiclusterengine** 사용자 정의 리소스의 설정이 자동으로 변경됩니다.

참고: 다중 클러스터 엔진 Operator 클러스터를 관리하는 Red Hat Advanced Cluster Management Hub 클러스터에서는 이전 수동 구성이 이 작업으로 교체됩니다.

Red Hat Advanced Cluster Management Hub 클러스터가 자체적으로 관리하도록 하려면 **spec.disableHubSelfManagement** 의 설정을 **false** 에서 **true** 로 변경해야 합니다. 설정이 사용자 지정 리소스를 정의하는 YAML 파일에 포함되어 있지 않은 경우 이를 추가해야 합니다. 허브 클러스터는 이 옵션을 사용하여 관리할 수 있습니다.

이 옵션을 **true** 로 설정하고 허브를 수동으로 관리하려고 하면 예기치 않은 동작이 발생합니다.

다음 예제에서는 허브 클러스터 자체 관리 기능을 비활성화하려는 경우 사용할 기본 템플릿을 보여줍니다. **namespace** 를 프로젝트 이름으로 교체합니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  disableHubSelfManagement: true
```

기본 **local-cluster** 를 활성화하려면 설정을 **false** 로 반환하거나 이 설정을 제거합니다.

1.4.7. disableUpdateClusterImageSets

모든 클러스터에 동일한 릴리스 이미지를 사용하도록 하려면 클러스터를 생성할 때 사용 가능한 고유한 사용자 정의 릴리스 이미지 목록을 생성할 수 있습니다.

사용 가능한 릴리스 이미지를 관리하고 **spec.disableUpdateClusterImageSets** 특성을 설정하기 위해 연결할 때 릴리스 이미지의 사용자 정의 목록 유지 관리의 다음 지침을 참조하십시오. 이 경우 사용자 정의 이미지 목록을 덮어쓰지 않습니다.

다음 예제에서는 클러스터 이미지 세트에 대한 업데이트를 비활성화하는 기본 템플릿을 보여줍니다. **namespace** 를 프로젝트 이름으로 교체합니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  disableUpdateClusterImageSets: true
```

1.4.8. customCAConfigmap (더 이상 사용되지 않음)

기본적으로 Red Hat OpenShift Container Platform은 Ingress Operator를 사용하여 내부 CA를 생성합니다.

다음 예제에서는 Red Hat Advanced Cluster Management에 사용자 지정 OpenShift Container Platform 기본 수신 CA 인증서를 제공하는 데 사용되는 기본 템플릿을 보여줍니다. **namespace** 를 프로젝트 이름으로 교체합니다. **spec.customCAConfigmap** 값을 **ConfigMap**의 이름으로 바꿉니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  customCAConfigmap: <configmap>
```

1.4.9. sslCiphers (더 이상 사용되지 않음)

기본적으로 Red Hat Advanced Cluster Management Hub 클러스터에는 지원되는 SSL 암호의 전체 목록이 포함되어 있습니다.

다음 예제는 관리 수신에 대한 **sslCiphers** 를 나열하는 데 사용되는 기본 **spec.ingress.sslCiphers** 템플릿을 보여줍니다. **namespace** 를 프로젝트 이름으로 교체합니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  ingress:
    sslCiphers:
      - "ECDHE-ECDSA-AES128-GCM-SHA256"
      - "ECDHE-RSA-AES128-GCM-SHA256"
```

1.4.10. ClusterBackup

enableClusterBackup 필드는 더 이상 지원되지 않으며 이 구성 요소로 교체됩니다.

다음 예제에서는 **ClusterBackup** 을 활성화하는 데 사용되는 **spec.overrides** 기본 템플릿을 보여줍니다. **namespace** 를 프로젝트 이름으로 교체합니다.

```
apiVersion: operator.open-cluster-management.io/v1
kind: MultiClusterHub
metadata:
  name: multiclusterhub
  namespace: <namespace>
spec:
  overrides:
    components:
      - name: cluster-backup
        enabled: true
```

또는 다음 명령을 실행할 수 있습니다. **namespace** 를 프로젝트 이름으로 교체합니다.

```
oc patch MultiClusterHub multiclusterhub -n <namespace> --type=json -p=[{"op": "add", "path": "/spec/overrides/components/-", "value": {"name": "cluster-backup", "enabled": true}}]
```

1.5. 업그레이드

Red Hat OpenShift Container Platform 콘솔의 operator 서브스크립션 설정을 사용하여 Kubernetes 업그레이드에 대한 Red Hat Advanced Cluster Management를 제어합니다. Operator Lifecycle Manager Operator **condition** 은 버전 업그레이드 방법을 제어하는 데 도움이 됩니다. Operator를 사용하여 Red Hat Advanced Cluster Management를 처음 배포할 때 다음과 같은 선택을 할 수 있습니다.

중요: 업그레이드는 즉시 이전 버전에서만 지원됩니다. 사용 가능한 다음 기능 릴리스로 업그레이드할 수 있지만 업그레이드하는 동안 릴리스를 건너뛸 수 없습니다.

- **채널:** 채널은 설치 중인 제품의 버전에 해당합니다. 초기 채널 설정은 설치 시 사용 가능한 가장 최신 채널인 경우가 많습니다.
- **승인:** 승인은 채널 내 업데이트에 승인이 필요한지 또는 자동으로 수행되는지 여부를 지정합니다.
 - **Automatic** 으로 설정하면 선택한 채널의 마이너 릴리스(Errata) 업데이트가 관리자의 개입 없이 배포됩니다.
 - **Manual** 로 설정된 경우 채널 내의 마이너 릴리스(Errata)에 대한 각 업데이트에는 관리자가 업데이트를 승인해야 합니다.

필수 액세스: OpenShift Container Platform 관리자

Operator를 사용하여 최신 버전의 Red Hat Advanced Cluster Management로 업그레이드할 때도 이러한 설정을 사용합니다. Operator를 업그레이드하려면 다음 단계를 완료합니다.

중요: 채널 선택에서 최신 버전으로 업그레이드한 후에는 이전 버전으로 되돌릴 수 없습니다. 이전 버전을 사용하려면 Operator를 설치 제거하고 이전 버전으로 다시 설치해야 합니다. OpenShift Container Platform Operator 허브에 로그인합니다.

1. OpenShift Container Platform 탐색에서 Operator > 설치된 **Operator** 를 선택합니다.
2. **Red Hat Advanced Cluster Management for Kubernetes** Operator를 선택합니다.
3. 서브스크립션 탭을 선택하여 서브스크립션 설정을 편집합니다.
4. 업그레이드 상태에 최신으로 레이블이 지정되었는지 확인합니다. 이 상태는 Operator가 선택한 채널에서 사용 가능한 최신 수준에 있음을 나타냅니다. Upgrade Status 가 업그레이드가 보류 중임을 나타내는 경우 다음 단계를 완료하여 채널에서 사용 가능한 최신 마이너 릴리스로 업데이트합니다.
 - a. 승인 필드에서 수동 설정을 클릭하여 값을 편집합니다.
 - b. 자동 업데이트를 활성화하려면 **Automatic** 을 선택합니다.
 - c. 저장을 선택하여 변경 사항을 커밋합니다.
 - d. 자동 업데이트가 Operator에 적용될 때까지 기다립니다. 업데이트는 선택한 채널의 최신 버전에 필요한 업데이트를 자동으로 추가합니다. 업데이트된 업데이트가 모두 완료되면 Upgrade Status 필드에 **Up to date** 가 표시됩니다.

참고: **MultiClusterHub** 사용자 정의 리소스가 업그레이드를 완료하는 데 최대 10분이 걸릴 수 있습니다. 다음 명령을 입력하여 업그레이드가 여전히 진행 중인지 확인할 수 있습니다.

```
oc get mch
```

업그레이드하는 동안 **Status** 필드에 updates가 표시됩니다. 업그레이드가 완료되면 **Status** 필드에 **Running** 이 표시됩니다.

5. 업그레이드 상태가 **최신** 이므로 채널 필드의 값을 클릭하여 편집합니다.
6. 사용 가능한 다음 기능 릴리스의 채널을 선택하지만 채널을 건너뛰지 마십시오.
중요: Operator Lifecycle Manager **Operatorcondition** 리소스는 현재 업그레이드 프로세스 중에 이전 업그레이드를 확인하고 버전을 건너뛰지 않습니다. 동일한 리소스 상태를 확인하여 upgradable 상태가 **true** 또는 **false** 인지 확인할 수 있습니다.
7. **저장**을 선택하여 변경 사항을 저장합니다.
8. 자동 업그레이드가 완료될 때까지 기다립니다. 다음 기능 릴리스로의 업그레이드가 완료되면 채널 내의 최신 패치 릴리스로 업데이트가 배포됩니다.
9. 이후 기능 릴리스로 업그레이드해야 하는 경우 Operator가 원하는 채널의 최신 수준에 있을 때까지 7-9단계를 반복합니다. 모든 패치 릴리스가 최종 채널에 배포되었는지 확인합니다.
10. 선택 사항: 채널 내 향후 업데이트가 **수동 승인**이 필요한 경우 승인 설정을 Manual로 설정할 수 있습니다.

Operator 업그레이드에 대한 자세한 내용은 OpenShift Container Platform 설명서의 [Operator](#) 를 참조하십시오.

1.5.1. 업그레이드를 사용하여 클러스터 풀 관리

[클러스터 풀\(기술 프리뷰\)](#)을 관리하는 경우 업그레이드 후 이러한 클러스터 풀의 자동 관리를 중지하려면 추가 구성이 필요합니다.

ClusterClaim metadata.annotations 에서 **cluster.open-cluster-management.io/createmanagercluster: "false"** 를 설정합니다.

이 설정을 변경하지 않는 한 제품을 업그레이드할 때 기존 클러스터 클레임은 자동으로 가져옵니다.

1.6. 연결이 끊긴 네트워크 환경에서 업그레이드

연결이 끊긴 네트워크 환경에서 Red Hat Advanced Cluster Management for Kubernetes를 업그레이드하는 단계 및 정보를 참조하십시오.

참고: 이 정보는 업그레이드의 업그레이드 절차를 따릅니다. ??? 해당 절차를 검토한 다음 다음 정보를 참조하십시오.

설치 또는 업그레이드 중에 Red Hat Advanced Cluster Management 및 멀티 클러스터 엔진 Operator 간의 상호 의존도와 관련된 중요한 정보가 표시될 수 있습니다. [설치 또는 업그레이드 중에 고려해야 할 사항은 연결이 끊긴 네트워크 환경에](#) 설치를 참조하십시오.

연결된 네트워크 환경에서 업그레이드하는 경우와 마찬가지로 Red Hat Advanced Cluster Management for Kubernetes용 Operator Lifecycle Manager 서브스크립션의 업그레이드 채널을 새 릴리스의 업그레이드 채널로 변경하여 업그레이드 프로세스가 시작됩니다.

그러나 연결이 끊긴 환경의 특수 특성으로 인해 업그레이드 프로세스를 시작하도록 업데이트 채널을 변경하기 전에 다음 미러링 요구 사항을 해결해야 합니다.

1. 미러 카탈로그에서 필수 패키지가 업데이트되었는지 확인합니다.

설치 중 또는 이전 업데이트 중에 연결이 끊긴 네트워크 환경에서 Red Hat Advanced Cluster Management for Kubernetes를 설치하는 데 필요한 Operator 패키지 및 이미지가 포함된 미러 카탈로그와 레지스트리를 생성했습니다. 업그레이드하려면 업데이트된 Operator 패키지 버전을 선택하려면 미러 카탈로그 및 레지스트리를 업데이트해야 합니다.

설치 작업과 유사하게 미러 카탈로그 및 레지스트리에 포함하거나 업데이트할 Operator 목록에 다음 Operator 패키지가 포함되어 있는지 확인해야 합니다.

- **advanced-cluster-manager**
- **multicluster-engine**

2. **MulticlusterHub** 리소스 인스턴스를 확인합니다.

설치 또는 이전 업데이트 중에 **MulticlusterHub** 리소스의 인스턴스를 생성했으며 연결이 끊긴 환경으로 인해 해당 리소스에 **mce-subscription-spec** 주석을 추가했습니다.

미러 카탈로그를 업데이트하는 절차로 인해 이전에 사용한 것과 동일한 이름으로 **CatalogSource** 를 통해 OpenShift Container Platform 클러스터에서 업데이트된 카탈로그를 사용할 수 있는 경우 **mce-subscriptino-spec** 주석을 업데이트하기 위해 **MulticlusterHub** 리소스를 업데이트할 필요가 없습니다.

그러나 미러링된 카탈로그를 업데이트하고 레지스트리에 새로 이름이 지정된 **CatalogSource** 가 생성되는 경우 **MulticlusterHub** 리소스에서 **mce-subscription-spec** 주석을 업데이트하여 새 카탈로그 소스 이름을 반영합니다.

1.6.1. 카탈로그 미러링으로 업그레이드

Red Hat Advanced Cluster Management는 관련 다중 클러스터 엔진 운영자 기능을 사용하여 제품의 일부로 제공되는 기본 서비스를 제공합니다. Red Hat Advanced Cluster Management는 허브 클러스터 설치 및 업그레이드의 일부로 필요한 다중 클러스터 엔진 Operator 및 **MulticlusterEngine** 리소스 인스턴스를 자동으로 설치하고 관리합니다.

연결된 네트워크 환경에서 클러스터 관리자는 특수 미러 카탈로그 및 카탈로그 소스 없이 Red Hat Advanced Cluster Management를 설치하거나 업그레이드할 수 있습니다. 그러나 연결이 끊긴 환경에서 Operator Lifecycle Manager Operator를 설치하려면 이전 섹션에 설명된 대로 특수 미러 카탈로그 및 카탈로그 소스를 사용해야 하므로 설치 후 몇 가지 추가 단계가 필요합니다.

1. 미러 카탈로그를 채우는 절차를 업데이트합니다.

Red Hat Advanced Cluster Management 미러링 절차를 설치할 때 Red Hat Operator 카탈로그의 전체 사본이 생성된 경우 특별한 미러링 업데이트가 필요하지 않습니다. 카탈로그를 새로 고침하여 새 Operator 릴리스의 업데이트된 콘텐츠를 가져옵니다.

그러나 프로시저가 *필터링된* 카탈로그인 미러 카탈로그를 채우는 경우 미러링 프로시저를 업데이트하여 **multicluster-engine** Operator 패키지가 미러 카탈로그에 포함되어 있고 **advanced-cluster-management** 패키지도 포함되어 있는지 확인해야 합니다.

미러 카탈로그 항목의 필수 Operator 패키지 포함에서 미러 카탈로그를 채울 때 사용할 옵션의 예를 참조하십시오. 이러한 새로운 요구 사항과 일치하도록 프로시저에 사용되는 operator-package 목록을 업데이트합니다.

2. **MulticlusterHub** 리소스 인스턴스를 업데이트합니다.

연결이 끊긴 네트워크 환경 설치에 설명된 대로 hub 클러스터를 설치하거나 연결이 끊긴 환경에서 업그레이드할 때 **MulticlusterHub** 리소스에 새 주석이 필요합니다.

모범 사례: Operator Lifecycle Manager 서브스크립션의 Operator Lifecycle Manager 업데이트 채널을 **advanced-cluster-management** Operator 패키지로 변경하기 전에 필요한 주석을 포함하도록 **MulticlusterHub** 리소스 인스턴스를 업데이트합니다. 이번 업데이트를 통해 업그레이드를 지연 없이 진행할 수 있습니다.

다음 예에 표시된 대로 **oc edit** 명령을 사용하여 **Multiclusterub** 리소스를 업데이트하여 **mce-subscription-spec** 주석을 추가합니다.

```
metadata:
  annotations:
    installer.open-cluster-management.io/mce-subscription-spec: '{"source": "<my-mirror-catalog-source>"}'
```

예제 의 **<my-mirror-catalog-source >**를 미리 카탈로그의 **openshift-marketplace** 네임스페이스에 있는 **CatalogSource** 리소스의 이름으로 바꿉니다.

중요: 주석을 추가하기 전에 업그레이드를 시작하면 Operator가 백그라운드에서 **다중 클러스터 엔진** 서브스크립션을 설치하려고 할 때 업그레이드가 시작되지만 중단됩니다. **MulticlusterHub** 리소스의 상태는 이 기간 동안 **업그레이드** 를 계속 표시합니다.

이 문제를 해결하려면 **oc edit** 를 실행하여 이전에 표시된 대로 **mce-subscription-spec** 주석을 추가합니다.

1.7. 정책을 사용하여 연결이 끊긴 클러스터 업그레이드

Red Hat OpenShift Update Service를 Kubernetes 정책용 Red Hat Advanced Cluster Management와 함께 사용하여 연결이 끊긴 환경에서 여러 클러스터를 업그레이드할 수 있습니다.

보안 문제로 인해 클러스터가 인터넷에 직접 연결되지 않는 경우도 있습니다. 이로 인해 업그레이드할 수 있는 시기와 이러한 업그레이드를 처리하는 방법을 알기가 어렵습니다. OpenShift Update Service를 구성하는 것이 도움이 될 수 있습니다.

OpenShift Update Service는 연결이 끊긴 환경에서 사용 가능한 관리 클러스터 버전을 모니터링하고 연결이 끊긴 환경에서 클러스터를 업그레이드하는 데 사용할 수 있도록 하는 별도의 Operator 및 피연산자입니다. OpenShift Update Service가 구성되면 다음 작업을 수행할 수 있습니다.

1. 연결이 끊긴 클러스터에 대한 업그레이드를 사용할 수 있는 시기를 모니터링합니다.
2. 그래프 데이터 파일을 사용하여 업그레이드하기 위해 로컬 사이트로 미러링되는 업데이트를 확인합니다.
3. 콘솔을 사용하여 클러스터에 업그레이드를 사용할 수 있음을 알립니다.
 - [사전 요구 사항](#)
 - [연결이 끊긴 미리 레지스트리 준비](#)
 - [OpenShift Update Service용 Operator 배포](#)
 - [그래프 데이터 init 컨테이너 빌드](#)
 - [미러링된 레지스트리에 대한 인증서 구성](#)
 - [OpenShift Update Service 인스턴스 배포](#)

- 기본 레지스트리를 덮어쓰는 정책 배포(선택 사항)
- 연결이 끊긴 카탈로그 소스 배포
- 정책을 배포하여 관리 클러스터 매개변수 변경
- 사용 가능한 업그레이드 보기
- 채널 선택
- 클러스터 업그레이드

1.7.1. 사전 요구 사항

OpenShift Update Service를 사용하여 연결이 끊긴 클러스터를 업그레이드하려면 다음 사전 요구 사항이 있어야 합니다.

- 제한된 OLM이 구성된 Red Hat OpenShift Container Platform 버전 4.13 이상에서 실행 중인 배포된 허브 클러스터입니다. 제한된 OLM 을 구성하는 방법에 대한 자세한 내용은 [제한된 네트워크에서 Operator Lifecycle Manager](#) 사용을 참조하십시오.
팁: 제한된 OLM을 구성할 때 카탈로그 소스 이미지를 기록합니다.
- 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 [Disconnected installation mirroring](#) 에서 참조하십시오.
참고: 업그레이드할 현재 버전의 클러스터의 이미지를 미러링된 이미지 중 하나로 항상 사용할 수 있어야 합니다. 업그레이드에 실패하면 업그레이드 시도 시 클러스터가 클러스터 버전으로 되돌아갑니다.

1.7.2. 연결이 끊긴 미러 레지스트리 준비

업그레이드할 이미지와 로컬 미러 레지스트리로 업그레이드할 현재 이미지를 모두 미러링해야 합니다. 이미지를 미러링하려면 다음 단계를 완료합니다.

1. 다음 예제와 유사한 콘텐츠가 포함된 스크립트 파일을 생성합니다.

```
UPSTREAM_REGISTRY=quay.io
PRODUCT_REPO=openshift-release-dev
RELEASE_NAME=ocp-release
OCP_RELEASE=4.13.2-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 바꿉니다.

2. 스크립트를 실행하여 이미지를 미러링하고, 설정을 구성하고, 릴리스 이미지와 릴리스 콘텐츠를 분리합니다.

팁: `ImageContentSourcePolicy` 를 생성할 때 이 스크립트의 마지막 줄의 출력을 사용할 수 있습니다.

1.7.3. OpenShift Update Service용 Operator 배포

OpenShift Container Platform 환경에서 OpenShift Update Service용 Operator를 배포하려면 다음 단계를 완료하십시오.

1. hub 클러스터에서 OpenShift Container Platform Operator 허브에 액세스합니다.
2. **Red Hat OpenShift Update Service Operator** 를 선택하여 Operator를 배포합니다. 필요한 경우 기본값을 업데이트합니다. Operator 배포는 `openshift-cincinnati` 라는 새 프로젝트를 생성합니다.
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.

1.7.4. 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 리포지토리](#)에서 직접 업그레이드할 수 있는 그래프 데이터 정보를 가져옵니다. 연결이 끊긴 환경을 구성하기 때문에 `init` 컨테이너를 사용하여 로컬 리포지토리에서 그래프 데이터를 사용할 수 있도록 설정해야 합니다. 그래프 데이터 `init` 컨테이너를 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 생성합니다. 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 Service가 이미지를 찾는 외부 레지스트리입니다.
- **RUN** 명령은 디렉토리를 생성하고 업그레이드 파일을 패키징합니다.
- **CMD** 명령은 패키지 파일을 로컬 리포지토리에 복사하고 업그레이드를 위해 파일을 추출합니다.

3. 다음 명령을 실행하여 `그래프 데이터 init` 컨테이너를 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 생성한 파일의 경로로 바꿉니다.

`${DISCONNECTED_REGISTRY}/cincinnati/cincinnati-data-container` 를 로컬 그래프 데이터 init 컨테이너의 경로로 교체합니다.

`/path/to/pull_secret` 을 풀 시크릿 파일의 경로로 바꿉니다.

참고: **podman** 이 설치되어 있지 않은 경우 명령의 **podman** 을 **docker** 로 교체할 수도 있습니다.

1.7.5. 미러링된 레지스트리에 대한 인증서 구성

미러링된 OpenShift Container Platform 릴리스 이미지를 저장하기 위해 보안 외부 컨테이너 레지스트리를 사용하는 경우 OpenShift Update Service는 업데이트 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 액세스에 대한 추가 신뢰 저장소 구성](#) 을 참조하십시오.
2. **openshift-config** 네임스페이스에 ConfigMap을 생성합니다.
3. **updateservice-registry** 키 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 **openshift-config** 네임스페이스에서 생성한 ConfigMap을 모니터링한 다음 CA 인증서가 변경된 경우 배포를 다시 시작합니다.

1.7.6. OpenShift Update Service 인스턴스 배포

허브 클러스터에 OpenShift Update Service 인스턴스 배포를 완료하면 클러스터 업그레이드의 이미지를 미리링하고 연결이 끊긴 관리 클러스터에서 사용할 수 있는 이 인스턴스가 있습니다. 인스턴스를 배포하려면 다음 단계를 완료합니다.

1. **openshift-cincinnati** 인 Operator의 기본 네임스페이스를 사용하지 않으려면 OpenShift Update Service 인스턴스에 대한 네임스페이스를 생성합니다.
 - a. OpenShift Container Platform 허브 클러스터 콘솔 탐색 메뉴에서 **관리 > 네임스페이스** 를 선택합니다.
 - b. **네임스페이스 생성** 을 선택합니다.
 - c. 네임스페이스의 이름과 네임스페이스에 대한 기타 정보를 추가합니다.
 - d. **생성** 을 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 유사할 수 있습니다.

```
apiVersion: cincinnati.openshift.io/v1beta2
kind: Cincinnati
metadata:
  name: openshift-update-service-instance
  namespace: openshift-cincinnati
spec:
  registry: <registry_host_name>:<port> 1
  replicas: 1
  repository: ${LOCAL_REGISTRY}/ocp4/release
  graphDataImage: '<host_name>:<port>/cincinnati-graph-data-container' 2
```

1 **1** **spec.registry** 값을 이미지의 로컬 연결이 끊긴 레지스트리의 경로로 바꿉니다.

2 **2** **spec.graphDataImage** 값을 그래프 데이터 init 컨테이너의 경로로 교체합니다. 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. 다소 시간이 걸릴 수 있지만 명령 결과에 인스턴스 및 Operator가 실행 중임을 표시하는 경우 프로세스가 완료됩니다.

1.7.7. 기본 레지스트리를 덮어쓰는 정책 배포(선택 사항)

참고: 이 섹션의 단계는 릴리스를 미리링된 레지스트리로 미리링한 경우에만 적용됩니다. **더 이상 사용되지 않음: PlacementRule**

OpenShift Container Platform에는 업그레이드 패키지를 찾는 위치를 지정하는 기본 이미지 레지스트리 값이 있습니다. 연결이 끊긴 환경에서는 해당 값을 릴리스 이미지를 미리링한 로컬 이미지 레지스트리의 경로로 교체하는 정책을 생성할 수 있습니다.

이러한 단계의 경우 정책의 이름은 *policy-mirror* 입니다. 정책을 생성하려면 다음 단계를 완료합니다.

1. hub 클러스터의 OpenShift Container Platform 환경에 로그인합니다.
2. 콘솔에서 **Governance > Create policy** 를 선택합니다.
3. **YAML** 스위치를 *On* 으로 설정하여 정책의 YAML 버전을 확인합니다.
4. YAML 코드의 모든 콘텐츠를 삭제합니다.
5. 다음 YAML 콘텐츠를 창에 붙여넣어 사용자 정의 정책을 생성합니다.

```

apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-mirror
  namespace: default
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
    apiVersion: policy.open-cluster-management.io/v1
    kind: ConfigurationPolicy
    metadata:
      name: policy-image-content-source-policy
    spec:
      object-templates:
      - complianceType: musthave
        objectDefinition:
          apiVersion: operator.openshift.io/v1alpha1
          kind: ImageContentSourcePolicy
          metadata:
            name: <your-local-mirror-name>
          spec:
            repositoryDigestMirrors:
            - mirrors:
              - <your-registry> 1
              source: registry.redhat.io
        ---
      apiVersion: policy.open-cluster-management.io/v1
      kind: PlacementBinding
      metadata:
        name: binding-policy-mirror
        namespace: default
      placementRef:
        name: placement-policy-mirror
        kind: PlacementRule
        apiGroup: apps.open-cluster-management.io
      subjects:
      - name: policy-mirror
        kind: Policy
        apiGroup: policy.open-cluster-management.io
        ---
      apiVersion: apps.open-cluster-management.io/v1
      kind: PlacementRule

```

```

metadata:
  name: placement-policy-mirror
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
    [] # selects all clusters if not specified

```

- 1 **your-registry** 를 로컬 미러 저장소의 경로로 바꿉니다. **oc adm release mirror** 명령을 입력하여 로컬 미러의 경로를 찾을 수 있습니다.

6. 지원되는 경우 **Enforce** 를 선택합니다.
7. **만들기** 를 선택하여 정책을 생성합니다.

1.7.8. 연결이 끊긴 카탈로그 소스 배포

Catalogsource 정책을 관리 클러스터로 내보내 연결된 위치에서 연결이 끊긴 로컬 레지스트리로 기본 위치를 변경합니다.

1. 콘솔 메뉴에서 **Governance > Create policy** 를 선택합니다.
2. **YAML** 스위치를 *On* 으로 설정하여 정책의 YAML 버전을 확인합니다.
3. **YAML** 코드의 모든 콘텐츠를 삭제합니다.
4. 다음 **YAML** 콘텐츠를 창에 붙여넣어 사용자 정의 정책을 생성합니다.

```

apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-catalog
  namespace: default
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
      apiVersion: policy.open-cluster-management.io/v1
      kind: ConfigurationPolicy
      metadata:
        name: policy-catalog
      spec:
        object-templates:
        - complianceType: musthave
          objectDefinition:
            apiVersion: config.openshift.io/v1
            kind: OperatorHub
            metadata:
              name: cluster
            spec:
              disableAllDefaultSources: true

```

```

- complianceType: musthave
  objectDefinition:
    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' 1
      displayName: My Operator Catalog
      publisher: grpc
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-catalog
  namespace: default
placementRef:
  name: placement-policy-catalog
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-catalog
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-catalog
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
    [] # selects all clusters if not specified

```

1 **spec.image** 값을 로컬 제한된 카탈로그 소스 이미지의 경로로 바꿉니다.

5. 지원되는 경우 **Enforce** 를 선택합니다.
6. **만들기** 를 선택하여 정책을 생성합니다.

1.7.9. 정책을 배포하여 관리 클러스터 매개변수 변경

ClusterVersion 정책을 관리 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 콘텐츠는 다음 내용과 유사할 수 있습니다.

```
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
```

- 허브 클러스터에서 다음 명령을 입력하여 OpenShift Update Service 피연산자의 경로 URL을 확인합니다. **oc get routes**. 이후 단계에서는 이 값을 참조하십시오.
- 허브 클러스터 콘솔 메뉴에서 **Governance > Create a policy**를 선택합니다.
- YAML** 스위치를 *On* 으로 설정하여 정책의 YAML 버전을 확인합니다.
- YAML** 코드의 모든 콘텐츠를 삭제합니다.
- 다음 **YAML** 콘텐츠를 창에 붙여넣어 사용자 정의 정책을 생성합니다.

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-cluster-version
  namespace: default
  annotations:
    policy.open-cluster-management.io/standards: null
    policy.open-cluster-management.io/categories: null
    policy.open-cluster-management.io/controls: null
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
      apiVersion: policy.open-cluster-management.io/v1
      kind: ConfigurationPolicy
      metadata:
        name: policy-cluster-version
      spec:
        object-templates:
        - complianceType: musthave
          objectDefinition:
            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 1
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
```

```

metadata:
  name: binding-policy-cluster-version
  namespace: default
placementRef:
  name: placement-policy-cluster-version
  kind: PlacementRule
  apiGroup: apps.open-cluster-management.io
subjects:
- name: policy-cluster-version
  kind: Policy
  apiGroup: policy.open-cluster-management.io
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-cluster-version
  namespace: default
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
    [] # selects all clusters if not specified

```

- 1 **objectDefinition.spec.upstream** 값을 hub 클러스터 OpenShift Update Service 피연산자의 경로로 바꿉니다.

다음 단계를 완료하여 피연산자의 경로를 확인할 수 있습니다.

- a. hub 클러스터에서 **oc get get routes -A** 명령을 실행합니다.
 - b. **cincinnati**. + 피연산자의 경로는 **HOST/PORT** 필드의 값입니다.
7. 지원되는 경우 **Enforce** 를 선택합니다.
 8. **만들기** 를 선택하여 정책을 생성합니다.
 9. 관리형 클러스터 CLI에서 **ClusterVersion** 의 upstream 매개변수가 다음을 입력하여 로컬 허브 클러스터 OpenShift Update Service URL로 업데이트되었는지 확인합니다.

```
oc get clusterversion -o yaml
```

결과가 다음 콘텐츠와 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

```

1.7.10. 사용 가능한 업그레이드 보기

다음 단계를 완료하여 관리 클러스터에 사용 가능한 업그레이드 목록을 볼 수 있습니다.

1. Kubernetes Operator 콘솔을 위해 다중 클러스터 엔진에 로그인합니다.
2. 탐색 메뉴에서 **Infrastructure > Clusters** 를 선택합니다.
3. **Ready** 상태에 있는 클러스터를 선택합니다.
4. **작업** 메뉴에서 **클러스터 업그레이드** 를 선택합니다.
5. 선택적 업그레이드 경로를 사용할 수 있는지 확인합니다.

참고: 현재 버전이 로컬 이미지 저장소에 미러링되지 않은 경우 사용 가능한 업그레이드 버전이 표시되지 않습니다.

1.7.11. 채널 선택

Red Hat Advanced Cluster Management 콘솔을 사용하여 OpenShift Container Platform에서 클러스터 업그레이드 채널을 선택할 수 있습니다. 이러한 버전은 미리 레지스트리에서 사용할 수 있어야 합니다. **채널 선택** 단계를 완료하여 업그레이드 채널을 지정합니다.

1.7.12. 클러스터 업그레이드

연결이 끊긴 레지스트리를 구성한 후 Red Hat Advanced Cluster Management 및 OpenShift Update Service는 연결이 끊긴 레지스트리를 사용하여 업그레이드를 사용할 수 있는지 확인합니다. 사용 가능한 업그레이드가 표시되지 않는 경우 클러스터의 현재 수준의 릴리스 이미지와 로컬 저장소에 미러링된 하나 이상의 이후 수준이 있는지 확인합니다. 현재 클러스터 버전의 릴리스 이미지를 사용할 수 없는 경우 업그레이드를 사용할 수 없습니다.

업그레이드하려면 다음 단계를 완료합니다.

1. 콘솔에서 **Infrastructure > Clusters** 를 선택합니다.
2. 사용 가능한 업그레이드가 있는지 확인할 클러스터를 찾습니다.
3. 업그레이드할 수 있는 업그레이드가 있는 경우 **클러스터의 배포 버전** 옆에 업그레이드가 사용 가능함을 나타냅니다.
4. 클러스터의 **옵션** 메뉴를 선택하고 **클러스터 업그레이드** 를 선택합니다.
5. 업그레이드 대상 버전을 선택하고 업그레이드를 선택합니다.

관리 클러스터가 선택한 버전으로 업데이트되었습니다.

클러스터 업그레이드에 실패하면 Operator는 일반적으로 업그레이드를 몇 번 재시도하고 중지한 후 실패한 구성 요소의 상태를 보고합니다. 경우에 따라 업그레이드 프로세스가 프로세스를 완료하려는 시도를 계속 순환합니다. 업그레이드 실패 후 클러스터를 이전 버전으로 롤백하는 것은 지원되지 않습니다. 클러스터 업그레이드에 실패하는 경우 Red Hat 지원에 문의하십시오.

1.8. 설치 제거

Kubernetes용 Red Hat Advanced Cluster Management를 설치 제거할 때 제거 프로세스의 두 가지 수준인 **사용자 정의 리소스 제거** 및 **전체 Operator 설치 제거**가 표시됩니다. 제거 프로세스는 최대 20분 정도 걸릴 수 있습니다.

- 첫 번째 수준은 사용자 정의 리소스 제거이며, 이는 **MultiClusterHub** 인스턴스의 사용자 정의 리소스를 제거하는 가장 기본적인 설치 제거 유형이지만 다른 필수 **Operator** 리소스는 그대로 둡니다. 이 설치 제거 수준은 동일한 설정 및 구성 요소를 사용하여 다시 설치하려는 경우 유용합니다.
- 두 번째 수준은 사용자 정의 리소스 정의와 같은 구성 요소를 제외하고 대부분의 **Operator** 구성 요소를 제거하는 더 완전한 제거 사항입니다. 이 단계를 계속할 때 사용자 정의 리소스 제거와 함께 제거되지 않은 모든 구성 요소 및 서브스크립션을 제거합니다. 이 제거 후 사용자 정의 리소스를 다시 설치하기 전에 **Operator**를 다시 설치해야 합니다.

1.8.1. 사전 요구 사항

Red Hat Advanced Cluster Management Hub 클러스터를 제거하기 전에 해당 허브 클러스터에서 관리하는 모든 클러스터를 분리해야 합니다. hub 클러스터에서 계속 관리하는 모든 클러스터를 분리한 다음 제거를 다시 시도합니다.

- **Discovery**를 사용하는 경우 제거를 시도할 때 다음 오류가 표시될 수 있습니다.

Cannot delete MultiClusterHub resource because DiscoveryConfig resource(s) exist

Discovery를 비활성화하려면 다음 단계를 완료합니다.

- 콘솔에서 **Discovered Clusters** 테이블로 이동하여 클러스터 검색 비활성화를 클릭합니다. 서비스를 제거할지 확인합니다.
- 터미널을 사용할 수도 있습니다. 다음 명령을 실행하여 검색을 비활성화합니다.

oc delete discoveryconfigs --all --all-namespaces

- 에이전트 서비스 구성이 연결된 경우 다음 메시지가 표시될 수 있습니다.

Cannot delete MultiClusterHub resource because AgentServiceConfig resource(s) exist

- 명령줄 인터페이스를 사용하여 **AgentServiceConfig** 리소스를 비활성화하고 제거하려면 다음 단계를 완료합니다.
 - a. hub 클러스터에 로그인합니다.
 - b. 다음 명령을 입력하여 **AgentServiceConfig** 사용자 지정 리소스를 삭제합니다.

oc delete agentserviceconfig --all

- 관리 클러스터가 연결되어 있는 경우 다음 메시지가 표시될 수 있습니다. 참고: 여기에는 자체 관리 허브 클러스터인 **local-cluster**가 포함되지 않습니다.

Cannot delete MultiClusterHub resource because ManagedCluster resource(s) exist

클러스터 분리에 대한 자세한 내용은 [클러스터 생성 소개에서 공급자의 정보를 선택하여 관리에서 클러스터 제거](#) 섹션을 참조하십시오.

- **Observability**가 있는 경우 다음이 표시될 수 있습니다.

Cannot delete MultiClusterHub resource because MultiClusterObservability resource(s) exist

- 터미널을 사용하여 **MultiClusterObservability** 사용자 정의 리소스를 비활성화하고 제거하려면 다음 절차를 참조하십시오.
 - a. **hub** 클러스터에 로그인합니다.
 - b. 다음 명령을 입력하여 **MultiClusterObservability** 사용자 정의 리소스를 삭제합니다.

```
oc delete mco observability
```

- 콘솔을 사용하여 **MultiClusterObservability** 사용자 정의 리소스를 제거하려면 다음 절차를 참조하십시오.
 - a. **MultiClusterObservability** 사용자 정의 리소스가 설치된 경우 **MultiClusterObservability**의 탭을 선택합니다.
 - b. **MultiClusterObservability** 사용자 정의 리소스의 옵션 메뉴를 선택합니다.
 - c. **MultiClusterObservability** 삭제를 선택합니다.
리소스를 삭제하면 Red Hat Advanced Cluster Management Hub 클러스터의 **open-cluster-management-observability** 네임스페이스의 Pod와 모든 관리 클러스터의 **open-cluster-management-addon-observability** 네임스페이스의 Pod가 제거됩니다.

참고: 관찰 기능을 제거한 후 오브젝트 스토리지는 영향을 받지 않습니다.

1.8.2. 명령을 사용하여 리소스 제거

1. 아직 설치되지 않은 경우 OpenShift Container Platform CLI가 **oc** 명령을 실행하도록 구성되어 있는지 확인합니다. **oc** 명령 구성에 대한 자세한 내용은 OpenShift Container Platform 설명서에서 **OpenShift CLI 시작하기**를 참조하십시오.
2. 다음 명령을 입력하여 프로젝트 네임스페이스로 변경합니다. **namespace**를 프로젝트 네임스페이스의 이름으로 교체합니다.

```
oc project <namespace>
```

3. 다음 명령을 입력하여 **MultiClusterHub** 사용자 정의 리소스를 제거합니다.

```
oc delete multiclusterhub --all
```

4. 진행 상황을 보려면 다음 명령을 입력합니다.

```
oc get mch -o yaml
```

5. 정리 스크립트를 실행하여 나머지 잠재적 아티팩트를 제거합니다. 동일한 클러스터에서 이전 버전의 Red Hat Advanced Cluster Management로 다시 설치하려는 경우 이 정리 스크립트를 실행합니다.

- a. Helm 설치 지침에 따라 Helm CLI 바이너리 버전 3.2.0 이상을 설치합니다.
<https://helm.sh/docs/intro/install/>
- b. 다음 스크립트를 파일에 복사합니다.

```
#!/bin/bash  
ACM_NAMESPACE=<namespace>  
oc delete mch --all -n $ACM_NAMESPACE
```

```
oc delete apiservice v1.admission.cluster.open-cluster-management.io
v1.admission.work.open-cluster-management.io
oc delete clusterimageset --all
oc delete clusterrole multiclusterengines.multicluster.openshift.io-v1-admin
multiclusterengines.multicluster.openshift.io-v1-crdview
multiclusterengines.multicluster.openshift.io-v1-edit
multiclusterengines.multicluster.openshift.io-v1-view open-cluster-
management:addons:application-manager open-cluster-management:admin-
aggregate open-cluster-management:cert-policy-controller-hub open-cluster-
management:cluster-manager-admin-aggregate open-cluster-management:config-
policy-controller-hub open-cluster-management:edit-aggregate open-cluster-
management:iam-policy-controller-hub open-cluster-management:policy-
framework-hub open-cluster-management:view-aggregate
oc delete crd klusterletaddonconfigs.agent.open-cluster-management.io
placementbindings.policy.open-cluster-management.io policies.policy.open-
cluster-management.io userpreferences.console.open-cluster-management.io
discoveredclusters.discovery.open-cluster-management.io
discoveryconfigs.discovery.open-cluster-management.io
oc delete mutatingwebhookconfiguration ocm-mutating-webhook
managedclustermutators.admission.cluster.open-cluster-management.io
multicluster-observability-operator
oc delete validatingwebhookconfiguration
channels.apps.open.cluster.management.webhook.validator application-webhook-
validator multiclusterhub-operator-validating-webhook ocm-validating-webhook
multicluster-observability-operator multiclusterengines.multicluster.openshift.io
```

스크립트에서 <namespace >를 Red Hat Advanced Cluster Management가 설치된 네임스페이스 이름으로 바꿉니다.

중요: 네임스페이스가 정리 및 삭제되므로 올바른 네임스페이스를 지정해야 합니다.

- a. 스크립트를 실행하여 이전 설치에서 남아 있는 가능한 아티팩트를 제거합니다. 남아 있는 아티팩트가 없으면 리소스를 찾을 수 없는 메시지가 반환됩니다.

참고: 동일한 Red Hat Advanced Cluster Management 버전을 다시 설치하려는 경우 이 절차의 다음 단계를 건너뛰고 사용자 지정 리소스를 다시 설치할 수 있습니다. Operator의 전체 제거를 진행합니다.

6. 다음 명령을 입력하여 설치된 네임스페이스에 Red Hat Advanced Cluster Management **ClusterServiceVersion** 및 **Subscription** 을 삭제합니다. 2.x.0 값을 현재 메이저 또는 마이너 릴리스로 바꿉니다.

```
oc get csv
NAME                                DISPLAY                                VERSION  REPLACES
PHASE
advanced-cluster-management.v2.x.0  Advanced Cluster Management for Kubernetes
2.x.0          Succeeded

oc delete clusterserviceversion advanced-cluster-management.v2.x.0

oc get sub
NAME                                PACKAGE                                SOURCE                                CHANNEL
acm-operator-subscription          advanced-cluster-management          acm-custom-registry
release-2.x

oc delete sub acm-operator-subscription
```

참고: 서브스크립션 이름과 CSV 버전의 이름은 다를 수 있습니다.

1.8.3. 콘솔을 사용하여 구성 요소 삭제

Red Hat OpenShift Container Platform 콘솔을 사용하여 설치 제거할 때 Operator를 제거합니다. 콘솔을 사용하여 설치 제거하려면 다음 단계를 완료합니다.

1. OpenShift Container Platform 콘솔 탐색에서 Operator > 설치된 Operator > Advanced Cluster Manager for Kubernetes 를 선택합니다.
2. **MultiClusterHub** 사용자 정의 리소스를 제거합니다.
 - a. *Multiclusterhub*의 탭을 선택합니다.
 - b. MultiClusterHub 사용자 정의 리소스의 옵션 메뉴를 선택합니다.
 - c. MultiClusterHub 삭제를 선택합니다.
3. 명령을 사용하여 **MultiClusterHub** 인스턴스 제거의 절차에 따라 정리 스크립트를 실행합니다.
참고: 동일한 Red Hat Advanced Cluster Management 버전을 다시 설치하려는 경우 이 절차의 나머지 단계를 건너뛰고 사용자 지정 리소스를 다시 설치할 수 있습니다.
4. 설치된 Operator로 이동합니다.
5. *Options* 메뉴를 선택하고 *Uninstall operator* 를 선택하여 *Red Hat Advanced Cluster Management Operator*를 제거합니다.