



Red Hat OpenShift Container Storage 4.8

베어 메탈 인프라를 사용하여 OpenShift Container Storage 배포

베어 메탈 환경 설치 및 설정 방법

Red Hat OpenShift Container Storage 4.8 베어 메탈 인프라를 사용하여 OpenShift Container Storage 배포

베어 메탈 환경 설치 및 설정 방법

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

법적 공지

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Deploying_OpenShift_Container_Storage_using_bare_metal_infrastructure.ent file |.

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

베어 메탈 인프라에서 로컬 스토리지를 사용하기 위해 Red Hat OpenShift Container Storage 4.8을 설치하는 방법에 대한 자세한 내용은 이 문서를 참조하십시오.

차례	
보다 포괄적 수용을 위한 오픈 소스 용어 교체	3
RED HAT 문서에 관한 피드백 제공	4
PREFACE	5
1장. OPENSIFT CONTAINER STORAGE 배포 준비	6
1.1. 로컬 스토리지 장치를 사용하여 OPENSIFT CONTAINER STORAGE를 설치하기 위한 요구사항	6
1.2. RED HAT ENTERPRISE LINUX 기반 노드의 컨테이너에 대한 파일 시스템 액세스 활성화	7
1.3. VAULT에서 키 값 백엔드 경로 및 정책 활성화	8
2장. 로컬 스토리지 장치를 사용하여 OPENSIFT CONTAINER STORAGE 배포	9
2.1. LOCAL STORAGE OPERATOR 설치	9
2.2. RED HAT OPENSIFT CONTAINER STORAGE OPERATOR 설치	9
2.3. MULTUS 네트워크 생성	10
2.3.1. 네트워크 연결 정의 생성	11
2.4. 베어 메탈에서 OPENSIFT CONTAINER STORAGE 클러스터 생성	12
3장. OPENSIFT CONTAINER STORAGE 배포 확인	17
3.1. POD 상태 확인	17
3.2. OPENSIFT CONTAINER STORAGE 클러스터가 정상인지 확인	18
3.3. MULTICLOUD OBJECT GATEWAY가 정상인지 확인	19
3.4. OPENSIFT CONTAINER STORAGE 특정 스토리지 클래스가 있는지 확인	19
3.5. MULTUS 네트워킹 확인	20
4장. OPENSIFT CONTAINER STORAGE 설치 제거	23
4.1. 내부 모드에서 OPENSIFT CONTAINER STORAGE 설치 제거	23
4.1.1. 로컬 스토리지 Operator 구성 제거	32
4.2. OPENSIFT CONTAINER STORAGE에서 모니터링 스택 제거	34
4.3. OPENSIFT CONTAINER STORAGE에서 OPENSIFT CONTAINER PLATFORM 레지스트리 제거	38
4.4. OPENSIFT CONTAINER STORAGE에서 클러스터 로깅 OPERATOR 제거	39

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 이를 개선하는 방법을 알려 주십시오. 피드백을 제공하려면 다음을 수행하십시오.

- 특정 문구에 대한 간단한 주석은 다음과 같습니다.
 1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **Feedback** (피드백) 버튼이 있는지 확인합니다.
 2. 마우스 커서를 사용하여 주석 처리하려는 텍스트 부분을 강조 표시합니다.
 3. 강조 표시된 텍스트 아래에 표시되는 **피드백 추가** 팝업을 클릭합니다.
 4. 표시된 지침을 따릅니다.
- 보다 상세하게 피드백을 제출하려면 다음과 같이 Bugzilla 티켓을 생성하십시오.
 1. [Bugzilla](#) 웹 사이트로 이동하십시오.
 2. 구성 요소 섹션에서 **문서** 를 선택합니다.
 3. **Description** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
 4. **Submit Bug**를 클릭하십시오.

PREFACE

Red Hat OpenShift Container Storage 4.8은 연결되거나 연결이 끊긴 환경의 기존 Red Hat OpenShift Container Platform (RHOCP) 베어 메탈 클러스터에 프록시 환경에 대한 즉시 지원과 함께 배포를 지원합니다.

베어 메탈에서 내부 및 외부 Openshift Container Storage 클러스터가 모두 지원됩니다. [배포 요구 사항에 대한 자세한 내용은 배포 계획 및 OpenShift Container Storage 배포 준비를 참조하십시오.](#)

OpenShift Container Storage를 배포하려면 환경에 적절한 배포 프로세스를 따르십시오.

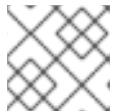
- 내부 모드
 - 로컬 스토리지 장치를 사용하여 배포
- 외부 모드

1장. OPENSIFT CONTAINER STORAGE 배포 준비

로컬 스토리지 장치를 사용하여 OpenShift Container Platform에 OpenShift Container Storage를 배포할 때 내부 클러스터 리소스를 생성할 수 있는 옵션이 제공됩니다. 이로 인해 기본 서비스의 내부 프로비저닝이 생성되어 애플리케이션에서 추가 스토리지 클래스를 사용할 수 있습니다.

로컬 스토리지를 사용하여 Red Hat OpenShift Container Storage 배포를 시작하기 전에 리소스 요구 사항을 충족해야 합니다. 자세한 내용은 [로컬 스토리지 장치를 사용하여 OpenShift Container Storage를 설치하기 위한 요구 사항](#)을 참조하십시오.

- 작업자 노드의 Red Hat Enterprise Linux 기반 호스트에서 파일 시스템 액세스를 활성화합니다. [Red Hat Enterprise Linux 기반 노드의 컨테이너에 대한 파일 시스템 액세스를 활성화합니다](#) .



참고

RHCOS(Red Hat Enterprise Linux CoreOS)에 대해 이 단계를 건너뛰십시오.

- 선택 사항: 외부 키 관리 시스템(KMS)을 사용하여 클러스터 전체 암호화를 활성화하려면 다음을 수행합니다.
 - 토큰이 있는 정책이 존재하고 Vault에 키 값 백엔드 경로가 활성화되어 있는지 확인합니다. [자격 증명 모음의 키 값 백엔드 경로 및 정책을 참조하십시오](#).
 - Vault 서버에서 서명된 인증서를 사용 중인지 확인합니다.

위의 문제를 해결한 후 지정된 순서의 섹션을 따르십시오.

1. [Local Storage Operator 설치](#) .
2. [Red Hat OpenShift Container Storage Operator 설치](#) .
3. [베어 메탈에서 OpenShift Container Storage 클러스터 생성](#) .

1.1. 로컬 스토리지 장치를 사용하여 OPENSIFT CONTAINER STORAGE를 설치하기 위한 요구사항

노드 요구 사항

클러스터는 각각 로컬에 연결된 스토리지 장치가 있는 3개 이상의 OpenShift Container Platform 작업자 노드로 구성되어야 합니다.

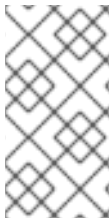
- 선택한 3개의 노드 각각에는 OpenShift Container Storage에서 사용할 수 있는 원시 블록 장치가 하나 이상 있어야 합니다.
- 사용하는 장치는 비어 있어야 합니다. 디스크에 PV(물리 볼륨), VG(볼륨 그룹) 또는 LV(논리 볼륨)가 디스크에 남아 있지 않아야 합니다.

자세한 내용은 계획 가이드의 [리소스 요구 사항](#) 섹션을 참조하십시오.

Arbiter 확장 클러스터 요구 사항 [기술 프리뷰]

이 경우 단일 클러스터는 Arbiter의 위치로 세 번째 영역으로 두 영역으로 확장됩니다. 현재 OpenShift Container Platform 온프레미스에 배포하기 위한 기술 프리뷰 기능입니다.

자세한 요구 사항 및 지침은 [ECDSA -DR 확장 클러스터용 OpenShift Container Storage](#) 구성을 참조하십시오.



참고

스케일링 논리가 충돌하므로 유연한 스케일링과 중재자를 모두 활성화할 수 없습니다. 유연성 확장을 사용하면 OpenShift Container Storage 클러스터에 한 번에 하나의 노드를 추가할 수 있습니다. 반면 arbiter 클러스터에서는 2개의 데이터 영역 각각에 하나 이상의 노드를 추가해야 합니다.

압축 모드 요구 사항

OpenShift Container Storage는 모든 워크로드가 3개의 강력한 마스터 노드에서 실행되는 3노드 OpenShift 소형 베어 메탈 클러스터에 설치할 수 있습니다. 작업자 또는 스토리지 노드가 없습니다.

압축 모드로 OpenShift Container Platform을 구성하려면 [3-노드 클러스터 구성 및 에지 배포를 위한 3-노드 아키텍처](#) 제공을 참조하십시오.

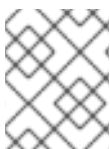
최소 시작 노드 요구 사항 [기술 프리뷰]

표준 배포 리소스 요구 사항이 충족되지 않는 경우 OpenShift Container Storage 클러스터는 최소 구성으로 배포됩니다.

자세한 내용은 계획 가이드의 [리소스 요구 사항](#) 섹션을 참조하십시오.

1.2. RED HAT ENTERPRISE LINUX 기반 노드의 컨테이너에 대한 파일 시스템 액세스 활성화

UPI(사용자 프로비저닝 인프라)의 Red Hat Enterprise Linux 기반에 작업자 노드가 있는 OpenShift Container Platform에 OpenShift Container Storage를 배포하면 기본 Ceph 파일 시스템에 대한 컨테이너 액세스 권한이 자동으로 제공되지 않습니다.



참고

RHCOS(Red Hat Enterprise Linux CoreOS)를 기반으로 하는 호스트의 경우 이 섹션을 건너뛰십시오.

절차

1. Red Hat Enterprise Linux 기반 노드에 로그인하여 터미널을 엽니다.
2. 클러스터의 각 노드에 대해 다음을 수행합니다.
 - a. 노드가 rhel-7-server-extras-rpms 리포지토리에 액세스할 수 있는지 확인합니다.

```
# subscription-manager repos --list-enabled | grep rhel-7-server
```

출력에 **rhel-7-server-rpms** 및 **rhel-7-server-extras-rpms** 가 표시되지 않거나 출력이 없는 경우 다음 명령을 실행하여 각 리포지토리를 활성화합니다.

```
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

- b. 필수 패키지를 설치합니다.

```
# yum install -y policycoreutils container-selinux
```

- c. SELinux에서 Ceph 파일 시스템의 컨테이너 사용을 영구적으로 활성화합니다.

```
# setsebool -P container_use_cephfs on
```

1.3. VAULT에서 키 값 백엔드 경로 및 정책 활성화

사전 요구 사항

- Vault에 대한 관리자 액세스.
- 나중에 변경할 수 없으므로 이름 지정 규칙을 따르는 고유한 경로 이름을 백엔드 경로로 선택합니다.

절차

1. Vault에서 KV(키/값) 백엔드 경로를 활성화합니다.
Vault KV 비밀 엔진 API의 경우 버전 1입니다.

```
$ vault secrets enable -path=ocs kv
```

Vault KV 비밀 엔진 API의 경우 버전 2입니다.

```
$ vault secrets enable -path=ocs kv-v2
```

2. 다음 명령을 사용하여 시크릿에서 쓰기 또는 삭제 작업을 수행하도록 사용자를 제한하는 정책을 생성합니다.

```
echo '
path "ocs/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}
path "sys/mounts" {
  capabilities = ["read"]
}' | vault policy write ocs -
```

3. 위 정책과 일치하는 토큰을 생성합니다.

```
$ vault token create -policy=ocs -format json
```

2장. 로컬 스토리지 장치를 사용하여 OPENSIFT CONTAINER STORAGE 배포

베어 메탈 인프라에서 로컬 스토리지 장치를 사용하여 OpenShift Container Storage를 배포하려면 다음 섹션을 따르십시오.

1. [Local Storage Operator 설치](#).
2. [Red Hat OpenShift Container Storage Operator 설치](#).
3. [베어 메탈에서 OpenShift Container Storage 클러스터 생성](#).

2.1. LOCAL STORAGE OPERATOR 설치

Red Hat OpenShift Container Platform Operator Hub를 사용하여 Local Storage Operator를 설치할 수 있습니다.

사전 요구 사항

- cluster-admin 및 Operator 설치 권한이 있는 계정을 사용하여 OpenShift Container Platform 클러스터에 액세스할 수 있습니다.

절차

1. OpenShift 웹 콘솔에 로그인합니다.
2. **Operators** → **OperatorHub** 를 클릭합니다.
3. **Filter by keyword...** 상자에 **로컬 스토리지**를 입력하여 운영자 목록에서 **Local Storage Operator**를 검색하고 클릭합니다.
4. **설치**를 클릭합니다.
5. **Operator 설치** 페이지에서 다음 옵션을 설정합니다.
 - a. 채널은 **stable-4.8**입니다.
 - b. 클러스터의 특정 네임스페이스로서의 설치 모드입니다.
 - c. **Operator** 권장 네임스페이스 **openshift-local-storage** 로 설치된 네임스페이스입니다.
 - d. 자동 승인 전략.
6. **설치**를 클릭합니다.

검증 단계

- Local Storage Operator가 **Succeeded** 로 상태가 표시되는지 확인합니다.

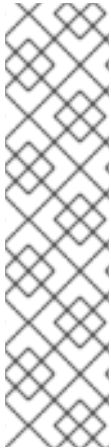
2.2. RED HAT OPENSIFT CONTAINER STORAGE OPERATOR 설치

Red Hat OpenShift Container Platform Operator Hub를 사용하여 Red Hat OpenShift Container Storage Operator를 설치할 수 있습니다.

사전 요구 사항

사전 요구 사항

- cluster-admin 및 Operator 설치 권한이 있는 계정을 사용하여 OpenShift Container Platform 클러스터에 액세스할 수 있습니다.
- Red Hat OpenShift Container Platform 클러스터에 작업자 노드가 3개 이상 있습니다.
- 필요한 추가 요구 사항을 충족했습니다. 자세한 내용은 [배포 계획을](#) 참조하십시오.



참고

- OpenShift Container Storage의 클러스터 수준 기본 노드 선택기를 재정의해야 하는 경우 다음 명령을 사용하여 **openshift-storage** 네임스페이스에 대한 빈 노드 선택기를 지정할 수 있습니다(이 경우 openshift-storage 네임스페이스 생성):

```
$ oc annotate namespace openshift-storage openshift.io/node-selector=
```

- 노드를 인프라로 테인트 하여 Red Hat OpenShift Container Storage 리소스만 해당 노드에 예약되도록 합니다. 이를 통해 서브스크립션 비용을 절감할 수 있습니다. 자세한 내용은 스토리지 리소스 관리 및 할당 가이드 의 [Red Hat OpenShift Container Storage 전용 작업자 노드를 사용하는](#) 방법을 참조하십시오.

절차

1. OpenShift 웹 콘솔에 로그인합니다.
2. **Operators** → **OperatorHub**를 클릭합니다.
3. Operator 목록에서 **OpenShift Container Storage**를 검색하고 클릭합니다.
4. 설치를 클릭합니다.
5. **Operator 설치** 페이지에서 다음 옵션을 설정합니다.
 - a. 채널은 **stable-4.8**입니다.
 - b. 클러스터의 특정 네임스페이스로서의 설치 모드입니다.
 - c. **Operator** 권장 네임스페이스 **openshift-storage**로 설치된 네임스페이스입니다. 네임스페이스 **openshift-storage**가 없으면 운영자 설치 중에 생성됩니다.
 - d. 자동 또는 수동 승인 전략
 - e. 설치를 클릭합니다.

자동 업데이트를 선택하면 OLM(Operator Lifecycle Manager)이 개입 없이 Operator의 실행 중인 인스턴스를 자동으로 업그레이드합니다.

수동 업데이트를 선택하면 OLM에서 업데이트 요청을 생성합니다. 클러스터 관리자는 Operator를 새 버전으로 업데이트하려면 해당 업데이트 요청을 수동으로 승인해야 합니다.

검증 단계

- OpenShift Container Storage Operator에 설치에 성공한 녹색 눈금이 표시되는지 확인합니다.

2.3. MULTUS 네트워크 생성

OpenShift Container Platform은 Multus CNI 플러그인을 사용하여 CNI 플러그인 체인을 허용합니다. 클러스터 설치 중에 기본 Pod 네트워크를 구성할 수 있습니다. 기본 네트워크는 클러스터의 모든 일반 네트워크 트래픽을 처리합니다. 사용 가능한 CNI 플러그인을 기반으로 추가 네트워크를 정의하고 이러한 네트워크 중 하나 이상을 Pod에 연결할 수 있습니다. Pod에 추가 네트워크 인터페이스를 연결하려면 인터페이스 연결 방법을 정의하는 구성을 생성해야 합니다. **NetworkAttachmentDefinition** CR(사용자 정의 리소스)을 사용하여 각 인터페이스를 지정할 수 있습니다. 각 **NetworkAttachmentDefinition** 내부의 CNI 구성은 해당 인터페이스 생성 방법을 정의합니다.

OpenShift Container Storage는 **macvlan** 라는 CNI 플러그인을 사용합니다. macvlan 기반 추가 네트워크를 생성하면 호스트의 pod가 실제 네트워크 인터페이스를 사용하여 해당 호스트의 다른 호스트 및 포드와 통신할 수 있습니다. macvlan 기반 추가 네트워크에 연결된 각 pod에는 고유 한 MAC 주소가 제공됩니다.

2.3.1. 네트워크 연결 정의 생성

Multus를 활용하려면 올바른 네트워킹 구성으로 이미 작동 중인 클러스터가 필요합니다. 자세한 내용은 [Multus 구성에 대한 권장 네트워크 구성 및 요구 사항을 참조하십시오](#). 지금 생성된

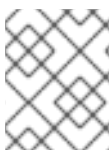
NetworkAttachmentDefinition (NAD)은 나중에 스토리지 클러스터 설치 중에 선택할 수 있습니다. 이는 스토리지 클러스터보다 먼저 생성해야 하는 이유입니다.

계획 가이드에 설명된 대로 생성하는 Multus 네트워크는 OpenShift Container Storage 트래픽에 대해 보류한 사용 가능한 네트워크 인터페이스 수에 따라 달라집니다. 모든 스토리지 트래픽을 두 인터페이스(기본 OpenShift SDN에 사용되는 하나의 인터페이스) 중 하나로 분리하거나 스토리지 트래픽을 클라이언트 스토리지 트래픽(공용) 및 스토리지 복제 트래픽(프라이빗 또는 클러스터)으로 추가로 분리할 수 있습니다.

다음은 동일한 인터페이스에서 공용 및 클러스터의 모든 스토리지 트래픽에 대한

NetworkAttachmentDefinition의 예입니다. 예약 가능한 모든 노드에 하나의 추가 인터페이스가 필요합니다(별도 네트워크 인터페이스의 OpenShift 기본 SDN).

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-public-cluster
  namespace: openshift-storage
spec:
  config: '{
    "cniVersion": "0.3.1",
    "type": "macvlan",
    "master": "ens2",
    "mode": "bridge",
    "ipam": {
      "type": "whereabouts",
      "range": "192.168.1.0/24"
    }
  }'
```



참고

모든 네트워크 인터페이스 이름은 Multus 네트워크에 연결된 모든 노드에서 동일해야 합니다(즉, **ocs-public-cluster**의 경우 **ens 2**).

다음은 별도의 Multus 네트워크의 스토리지 트래픽에 대한 **NetworkAttachmentDefinitions**의 예로, 복제 트래픽용 클라이언트 스토리지 트래픽 및 클러스터의 공용입니다. OSD 포드를 호스팅하는 OpenShift 노드에 두 개의 추가 인터페이스가 필요하며, 다른 모든 예약 가능한 노드에서 1개의 추가 인터페이스가 필요합니다(별도 네트워크 인터페이스의 OpenShift 기본 SDN).

```

apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-public
  namespace: openshift-storage
spec:
  config: '{
    "cniVersion": "0.3.1",
    "type": "macvlan",
    "master": "ens2",
    "mode": "bridge",
    "ipam": {
      "type": "whereabouts",
      "range": "192.168.1.0/24"
    }
  }'

```

NetworkAttachmentDefinition의 예 :

```

apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: ocs-cluster
  namespace: openshift-storage
spec:
  config: '{
    "cniVersion": "0.3.1",
    "type": "macvlan",
    "master": "ens3",
    "mode": "bridge",
    "ipam": {
      "type": "whereabouts",
      "range": "192.168.2.0/24"
    }
  }'

```



참고

모든 네트워크 인터페이스 이름은 Multus 네트워크에 연결된 모든 노드에서 동일해야 합니다(즉, **ocs-public**의 경우 **ens2**, **ocs-cluster**의 경우 **ens3**).

2.4. 베타 메탈에서 OPENSIFT CONTAINER STORAGE 클러스터 생성

사전 요구 사항

- 로컬 스토리지 장치를 사용하여 OpenShift Container Storage를 설치하기 위한 [요구 사항](#) 섹션의 모든 요구 사항이 충족되었는지 확인합니다.
- multus 지원의 기술 프리뷰 기능을 사용하려면 배포를 수행하기 전에 나중에 클러스터에 연결된 NAS(네트워크 연결 정의)를 생성해야 합니다. 자세한 내용은 [Multus 지원 및 네트워크 연결 정의 생성](#)을 참조하십시오.

절차

1. OpenShift 웹 콘솔에 로그인합니다.
2. **Operators** → 설치된 **Operator** 를 클릭하여 설치된 모든 Operator를 확인합니다.
3. 선택한 프로젝트가 **openshift-storage** 인지 확인합니다.
4. 스토리지 클러스터의 **OpenShift Container Storage** → **인스턴스 생성** 링크를 클릭합니다.
5. 모드를 **내부 연결 장치**로 선택합니다.
Local Storage Operator가 아직 설치되지 않은 경우 설치하라는 메시지가 표시됩니다. 설치를 클릭하고 [Local Storage Operator 설치](#)에서 설명하는 절차를 따르십시오.

스토리지 볼륨 세트를 필터링하여 스토리지를 사용하도록 전용 스토리지 클래스를 생성할 수 있습니다.

6. 다음 옵션 중 하나를 사용하여 디스크를 검색합니다.
 - 모든 노드에서 디스크를 검색할 모든 노드.
 - 노드를 선택하여 사용 가능한 노드의 하위 집합에서 디스크를 검색합니다.



중요

Arbiter 모드를 사용하려면 **All nodes** 옵션을 선택하지 마십시오. 대신 **노드 선택** 옵션을 사용하여 두 데이터 센터 영역에서 연결된 스토리지 장치가 있는 레이블이 지정된 노드를 선택합니다.

선택한 노드가 테인트되어 마법사에서 검색되지 않은 경우 [Red Hat 지식베이스 솔루션의 단계를 Local Storage Operator 리소스에 대한 허용 오차를 추가하는 해결 방법](#)으로 따르십시오.

선택한 노드가 집계된 30개의 CPU 및 72GiB RAM의 OpenShift Container Storage 클러스터 요구 사항과 일치하지 않으면 최소 클러스터가 배포됩니다. 최소 시작 노드 요구 사항은 [계획 가이드의 리소스 요구 사항](#) 섹션을 참조하십시오.


7. 다음을 클릭합니다.
8. 스토리지 클래스 생성
 - a. 로컬 볼륨 세트 이름을 입력합니다.
 - b. **Storage Class Name(스토리지 클래스 이름)**을 입력합니다. 기본적으로 볼륨 세트 이름은 스토리지 클래스 이름에 표시됩니다. 이름을 변경할 수도 있습니다.
 - c. 이전 단계에서 디스크 검색에 대해 선택한 노드가 **디스크별 필터** 섹션에 표시됩니다. 다음 명령 중 하나를 선택합니다.
 - 장치를 검색한 모든 노드를 선택하는 모든 노드의 디스크입니다.
 - 선택한 노드의 디스크로 장치를 검색한 노드의 하위 집합을 선택합니다.고가용성을 위해 3개의 다른 물리적 노드, 랙 또는 장애 도메인에 작업자 노드를 분산합니다.



중요

3개 이상의 가용 영역의 최소 요구 사항보다 적은 노드에 분산된 스토리지 클러스터를 생성할 때 유연한 확장 기능을 사용할 수 있습니다. 이 기능은 OpenShift Container Storage 4.7 클러스터의 새로운 배포에만 사용할 수 있으며 업그레이드된 클러스터를 지원하지 않습니다. 유연한 확장에 대한 자세한 내용은 [스토리지 확장 가이드를 참조하십시오](#).

- d. 사용 가능한 **디스크 유형** 목록에서 **SSD/NVMe** 을 선택합니다.
- e. **Advanced(고급)** 섹션을 확장하고 다음 옵션을 설정합니다.

블록 모드	블록은 기본적으로 선택됩니다.
장치 유형	디스크 유형 선택. 기본적으로 Disk(디스크) 및 Part(파트)가 선택됩니다.
디스크 크기	장치를 포함해야 하는 최소 및 최대 사용 가능한 크기입니다.  참고 장치에 대해 최소 크기를 100GB로 설정해야 합니다.
최대 디스크 제한	이는 노드에서 생성할 수 있는 최대 PV 수를 나타냅니다. 이 필드를 비워 두면 일치하는 노드에서 사용 가능한 모든 디스크에 PV가 생성됩니다.

- f. 다음을 클릭합니다. 새 스토리지 클래스 생성을 확인하는 팝업이 표시됩니다.
- g. **Yes (예)**를 클릭하여 계속합니다.

9. 용량 및 노드설정

- a. **스토리지 클래스** 선택. 기본적으로 이전 단계에서 만든 새 스토리지 클래스가 선택됩니다.
 - **선택한 노드** 는 이전 단계에서 선택한 노드를 표시합니다. 이 목록은 이전 단계에서 검색한 디스크를 반영하는 데 몇 분이 걸립니다.
- b. 다음을 클릭합니다.

10. 선택 사항: 보안 및 네트워크 구성을 설정합니다.

- a. **Enable encryption(암호화 사용)** 확인란을 선택하여 블록 및 파일 스토리지를 암호화합니다.
- b. 하나 또는 둘 다 **암호화 수준**을 선택합니다:
 - **전체 클러스터(블록 및 파일)**를 암호화 하기 위한 클러스터 전체 암호화.
 - **암호화가 활성화된 스토리지 클래스**를 사용하여 암호화된 영구 볼륨을 만드는 스토리지 클래스 암호화(블록만 해당).
- c. **Connect to an external key management service** 확인란을 선택합니다. 이는 클러스터 전체 암호화에 대해 선택 사항입니다.
 - i. **Key Management Service Provider** 는 기본적으로 **Vault** 로 설정됩니다.

- ii. Vault 서비스 이름, Vault 서버의 호스트 주소 (`http://<hostname>` 또는 `ip>`), 포트 번호 및 토큰 을 입력합니다.
- iii. 고급 설정을 확장하여 Vault 구성에 따라 추가 설정 및 인증서 세부 정보를 입력합니다
 - A. OpenShift Container Storage 전용 및 고유 백엔드 경로에 키 값 시크릿 경로를 입력합니다.
 - B. 선택 사항: TLS Server Name 및 Vault Enterprise Namespace를 입력합니다.
 - C. 각 PEM 인코딩 인증서 파일을 업로드하여 CA 인증서, 클라이언트 인증서 및 클라이언트 개인 키를 제공합니다.
 - D. 저장을 클릭합니다.
- 11. 여러 네트워크 인터페이스를 사용하려는 경우 단일 네트워크 또는 사용자 지정 (Multus) 네트워크를 사용하는 경우 **Default (SDN)**를 선택합니다.
 - a. 드롭다운에서 **Public Network Interface** (공용 네트워크 인터페이스)를 선택합니다.
 - b. 드롭다운에서 **Cluster Network Interface** (클러스터 네트워크 인터페이스)를 선택합니다.



참고

하나의 추가 네트워크 인터페이스만 사용하는 경우 공용 네트워크 인터페이스에 대해 단일 **NetworkAttachmentDefinition** (예: **ocs-public-cluster**)을 선택하고 Cluster Network Interface를 비워 둡니다.

- 12. 다음을 클릭합니다.
- 13. 구성 세부 정보를 검토합니다. 구성 설정을 수정하려면 **Back** 을 클릭하고 이전 구성 페이지로 리디렉션됩니다.
- 14. **생성**을 클릭합니다.
- 15. KV(Vault Key/Value) 시크릿 엔진 API, 버전 2가 KMS(Key Management System)로 클러스터 전체 암호화에 사용되는 경우 configmap을 편집합니다.
 - a. OpenShift 웹 콘솔에서 워크로드 → **ConfigMap**을 클릭합니다.
 - b. KMS 연결 세부 정보를 보려면 **ocs-kms-connection-details** 를 클릭합니다.
 - c. configmap을 편집합니다.
 - i. **작업 메뉴 (TI) → ConfigMap 편집**을 클릭합니다.
 - ii. **VAULT_BACKEND** 매개 변수를 **v2** 로 설정합니다.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: ocs-kms-connection-details
[...]
data:
  KMS_PROVIDER: vault
  KMS_SERVICE_NAME: vault
```

```
[...]
VAULT_BACKEND: v2
[...]
```

iii. 저장을 클릭합니다.

검증 단계

- 설치된 스토리지 클러스터의 최종 상태가 **Phase**로 표시되는지 확인하려면 다음을 수행하십시오. 녹색 눈금 표시가 있는 **ready** 입니다.
 - **Operators** → 설치된 **Operator** → 스토리지 클러스터 링크를 클릭하여 스토리지 클러스터 설치 상태를 확인합니다.
 - 또는 **Operator Details(운영자 세부 정보)** 탭에 있는 경우 **Storage Cluster(스토리지 클러스터)** 탭을 클릭하여 상태를 확인할 수 있습니다.
- 스토리지 클러스터에서 유연한 확장이 활성화되었는지 확인하려면 다음 단계를 수행합니다 (Arbiter 모드의 경우 유연한 확장이 비활성화됨).
 1. 스토리지 클러스터 탭에서 **ocs-storagecluster** 를 클릭합니다.
 2. YAML 탭에서 **spec** 섹션에서 flexibility **Scaling** 키를 검색하고 **status** 섹션에서 **failureDomain** 을 검색합니다. 유연한 확장이 true이고 **failureDomain** 이 host로 설정되면 유연한 확장 기능이 활성화됩니다.

```
spec:
flexibleScaling: true
[...]
status:
failureDomain: host
```

- OpenShift Container Storage의 모든 구성 요소가 성공적으로 설치되었는지 [확인하려면 OpenShift Container Storage 설치](#) 확인을 참조하십시오.
- OpenShift Container Storage가 성공적으로 설치되었는지 [확인하려면 OpenShift Container Storage 설치](#) 확인을 참조하십시오.

추가 리소스

- 초기 클러스터의 용량을 확장하려면 [스토리지 확장 가이드](#)를 참조하십시오.

3장. OPENSIFT CONTAINER STORAGE 배포 확인

내부 모드용 OpenShift Container Storage가 성공적으로 배포되었는지 확인하려면 다음 섹션을 따르십시오.

1. 포드 상태 확인.
2. OpenShift Container Storage 클러스터가 정상인지 확인합니다.
3. Multicloud Object Gateway가 정상인지 확인합니다.
4. OpenShift Container Storage 특정 스토리지 클래스가 있는지 확인합니다.

3.1. POD 상태 확인

OpenShift Container Storage의 포드가 running 상태인지 확인하려면 다음 절차를 따르십시오.

절차

1. OpenShift 웹 콘솔에 로그인합니다.
2. OpenShift 웹 콘솔의 왼쪽 창에서 워크로드 → 포드를 클릭합니다.
3. 프로젝트 드롭다운 목록에서 **openshift-storage** 를 선택합니다.
각 구성 요소에 대해 예상되는 Pod 수 및 노드 수에 따라 어떻게 달라지는지에 대한 자세한 내용은 표 3.1. "OpenShift Container 스토리지 클러스터에 해당하는 Pod" 을 참조하십시오.
4. **Running** 및 **Completed** 탭을 클릭하여 Pod가 실행 중이고 완료된 상태인지 확인합니다.

표 3.1. OpenShift Container 스토리지 클러스터에 해당하는 Pod

구성 요소	해당 Pod
OpenShift Container Storage Operator	<ul style="list-style-type: none"> ● ocs-operator-* (모든 작업자 노드에서 Pod 1개) ● ocs-metrics-exporter-*
Rook-ceph Operator	rook-ceph-operator-* 모든 작업자 노드에서 Pod 1개)
Multicloud Object Gateway	<ul style="list-style-type: none"> ● NooBaa-operator-* (작업자 노드에서 Pod 1개) ● NooBaa-core-* (모든 스토리지 노드에서 Pod 1개) ● NooBaa-db-pg-* (모든 스토리지 노드에 1 Pod) ● NooBaa-endpoint-* (모든 스토리지 노드에서 Pod 1개)

구성 요소	해당 Pod
MON	rook-ceph-mon-* 스토리지 노드에 분산된 3개의 Pod)
MGR	rook-ceph-mgr-* 모든 스토리지 노드에서 Pod 1개)
MDS	rook-ceph-mds-ocs-storagecluster-cephfilesystem-* 스토리지 노드에 분산된 2개의 Pod)
RGW	Rook-ceph-rgw-ocs-storagecluster-cephobjectstore-* (1개의 스토리지 노드에서 pod)
CSI	<ul style="list-style-type: none"> ● CephFS <ul style="list-style-type: none"> ○ CSI-cephfsplugin-* (1개의 작업자 노드에서 Pod) ○ CSI-cephfsplugin-provisioner-* (작업자 노드에 분산된 2 Pod) ● rbd <ul style="list-style-type: none"> ○ CSI-rbdplugin-* (각 작업자 노드에서 Pod 1개) ○ CSI-rbdplugin-provisioner-* (2 작업자 노드에 분산된 2 Pod)
rook-ceph-crashcollector	rook-ceph-crashcollector-* 각 스토리지 노드에서 Pod 1개)
OSD	<ul style="list-style-type: none"> ● Rook-ceph-osd-* (1개의 장치마다 pod) ● Rook-ceph-osd-prepare-ocs-deviceset-* (1개의 장치 용 pod)

3.2. OPENSIFT CONTAINER STORAGE 클러스터가 정상인지 확인

OpenShift Container Storage 클러스터가 정상인지 확인하려면 절차의 단계를 따르십시오.

절차

1. 스토리지 → 개요를 클릭하고 블록 및 파일 탭을 클릭합니다.

2. **상태 카드에서 Storage Cluster (스토리지 클러스터) 및 Data Resiliency (데이터 복원력)에 녹색 눈금**이 있는지 확인합니다.
3. **세부 정보 카드에서 클러스터 정보가 표시되는지** 확인합니다.

블록 및 파일 대시보드를 사용하는 OpenShift Container Storage 클러스터의 상태에 대한 자세한 내용은 [Monitoring OpenShift Container Storage](#) 를 참조하십시오.

3.3. MULTICLOUD OBJECT GATEWAY가 정상인지 확인

OpenShift Container Storage Multicloud Object Gateway가 정상인지 확인하려면 절차의 단계를 따르십시오.

절차

1. OpenShift 웹 콘솔에서 **스토리지** → **개요**를 클릭하고 **오브젝트** 탭을 클릭합니다.
2. **상태 카드에서 오브젝트 서비스 및 데이터 복원성 이 모두 Ready 상태(녹색 틱)인지** 확인합니다.
3. **세부 정보 카드에서 Multicloud Object Gateway 정보가 표시되는지** 확인합니다.

오브젝트 서비스 대시보드를 사용하는 **OpenShift Container Storage** 클러스터의 상태에 대한 자세한 내용은 [OpenShift Container Storage 모니터링](#)을 참조하십시오.

3.4. OPENSIFT CONTAINER STORAGE 특정 스토리지 클래스가 있는지 확인

클러스터에 스토리지 클래스가 있는지 확인하려면 절차의 단계를 따르십시오.

절차

1. **OpenShift 웹 콘솔에서 스토리지** → **스토리지 클래스** 를 클릭합니다.
2. **OpenShift Container Storage** 클러스터 생성을 사용하여 다음 스토리지 클래스가 생성되었는지 확인합니다.
 - **ocs-storagecluster-ceph-rbd**
 - **ocs-storagecluster-cephfs**

- **openshift-storage.noobaa.io**
- **ocs-storagecluster-ceph-rgw**

3.5. MULTUS 네트워킹 확인

Multus가 클러스터에서 작동하는지 확인하려면 **Multus** 네트워킹을 확인합니다.

절차

1. 네트워킹 구성 선택 사항에 따라 **OpenShift Container Storage Operator**는 다음 중 하나를 수행합니다.
 - 공용 네트워크 인터페이스에 대해 단일 **NetworkAttachmentDefinition** (예: **ocs-public-cluster**)만 선택하면 애플리케이션 **Pod**와 **OpenShift Container Storage** 클러스터 간의 트래픽이 이 네트워크에서 수행됩니다. 또한 클러스터는 **OSD** 간 트래픽을 복제 및 재조정하는 데 이 네트워크를 사용하도록 자체 구성됩니다.
 - **NetworkAttachmentDefinitions** (예: **ocs-public** 및 **ocs-cluster**) 모두 공용 네트워크 인터페이스 및 클러스터 네트워크 인터페이스에 대해 선택된 경우 클라이언트 스토리지 트래픽은 공용 네트워크에 있으며 클러스터 네트워크는 **OSD** 간의 복제 및 재조정 트래픽을 위한 것입니다.
2. 네트워킹 구성이 올바른지 확인하려면 다음 단계를 따르십시오.
 - a. **OpenShift** 콘솔에서 설치된 **Operator** → 스토리지 클러스터 → **ocs-storagecluster** 를 클릭합니다.
 - b. **YAML** 탭에서 **spec** 섹션에서 네트워크를 검색하고 네트워크 인터페이스 선택 사항에 구성이 올바른지 확인합니다. 이 예는 클라이언트 스토리지 트래픽을 스토리지 복제 트래픽과 분리하기 위한 것입니다.

샘플 출력:

```
[.]
spec:
  [.]
```



```
network:
provider: multus
selectors:
  cluster: openshift-storage/ocs-cluster
  public: openshift-storage/ocs-public
[..]
```

3. 명령줄 인터페이스를 사용하여 네트워크 구성이 올바른지 확인하려면 다음 명령을 실행합니다.

```
$ oc get storagecluster ocs-storagecluster \
-n openshift-storage \
-o=jsonpath='{.spec.network}'
```

샘플 출력:

```
{"provider":"multus","selectors":{"cluster":"openshift-storage/ocs-cluster","public":"openshift-storage/ocs-public"}}
```

4. **OSD Pod**가 올바른 네트워크를 사용하고 있는지 확인합니다.

- a. **openshift-storage** 네임스페이스에서 **OSD** 포드 중 하나를 사용하여 포드가 올바른 네트워크에 연결되어 있는지 확인합니다. 이 예는 클라이언트 스토리지 트래픽을 스토리지 복제 트래픽과 분리하기 위한 것입니다.



참고

둘 다 생성되는 경우 **OSD** 포드만 **Multus** 공용 및 클러스터 네트워크에 연결됩니다. 다른 모든 **OCS pod**는 **Multus** 공용 네트워크에 연결됩니다.

```
$ oc get -n openshift-storage $(oc get pods -n openshift-storage -o name -l app=rook-ceph-osd | grep 'osd-0') -
o=jsonpath='{.metadata.annotations.k8s\.v1\.cni\.cncf\.io/network-status}'
```

샘플 출력:

```
[{
  "name": "openshift-sdn",
  "interface": "eth0",
  "ips": [
    "10.129.2.30"
  ],
}
```

```

    "default": true,
    "dns": {}
  },{
    "name": "openshift-storage/ocs-cluster",
    "interface": "net1",
    "ips": [
      "192.168.2.1"
    ],
    "mac": "e2:04:c6:81:52:f1",
    "dns": {}
  },{
    "name": "openshift-storage/ocs-public",
    "interface": "net2",
    "ips": [
      "192.168.1.1"
    ],
    "mac": "ee:a0:b6:a4:07:94",
    "dns": {}
  }
]]

```

5.

OSD 포트가 명령줄 인터페이스를 사용하여 올바른 네트워크를 사용하고 있는지 확인하려면 다음 명령을 실행합니다(**jq** 유틸리티 필요).

```

$ oc get -n openshift-storage $(oc get pods -n openshift-storage -o name -l app=rook-ceph-osd | grep 'osd-0') -o=jsonpath='{.metadata.annotations.k8s\.v1\.cni\.cncf\.io/network-status}{"\n"}' | jq -r '.[].name'

```

샘플 출력:

```

openshift-sdn
openshift-storage/ocs-cluster
openshift-storage/ocs-public

```

4장. OPENSIFT CONTAINER STORAGE 설치 제거

OpenShift Container Storage를 제거하고 제거하려면 다음 섹션을 따르십시오.

1. 내부 모드에서 **OpenShift Container Storage** 설치 제거.
2. **OpenShift Container Storage**에서 모니터링 스택 제거.
3. **OpenShift Container Storage**에서 **OpenShift Container Platform** 레지스트리 제거.
4. **OpenShift Container Storage**에서 클러스터 로깅 운영자 제거.

4.1. 내부 모드에서 OPENSIFT CONTAINER STORAGE 설치 제거

이 섹션의 단계를 사용하여 **OpenShift Container Storage**를 제거합니다.

주석 제거

스토리지 클러스터의 주석은 제거 프로세스의 동작을 변경하는 데 사용됩니다. 설치 제거 동작을 정의하기 위해 스토리지 클러스터에 다음 두 개의 주석이 도입되었습니다.

- **uninstall.ocs.openshift.io/cleanup-policy: delete**
- **uninstall.ocs.openshift.io/mode: 정상**

아래 표는 이러한 주석과 함께 사용할 수 있는 다양한 값에 대한 정보를 제공합니다.

표 4.1. **uninstall.ocs.openshift.io** 제거 주석 설명

주석	현재의	Default	동작
cleanup-policy	delete	있음	Rook은 물리적 드라이브와 DataDirHostPath 를 정리합니다.

주석	현재의	Default	동작
cleanup-policy	유지	없음	Rook은 물리적 드라이브 및 DataDirHostPath 를 정리 하지 않습니다.
mode	정상	있음	Rook 및 NooBaa는 PVC와 OBC가 관리자/사용자가 제거할 때까지 제거 프로세스를 일시 중지합니다.
mode	강제	없음	Rook 및 NooBaa는 Rook 및 NooBaa를 사용하여 프로비저닝된 PVC/OBC가 각각 존재하는 경우에도 제거를 진행합니다.

다음 명령을 사용하여 주석의 값을 편집하여 정리 정책 또는 설치 제거 모드를 변경할 수 있습니다.

```
$ oc annotate storagecluster -n openshift-storage ocs-storagecluster
uninstall.ocs.openshift.io/cleanup-policy="retain" --overwrite
storagecluster.ocs.openshift.io/ocs-storagecluster annotated
```

```
$ oc annotate storagecluster -n openshift-storage ocs-storagecluster
uninstall.ocs.openshift.io/mode="forced" --overwrite
storagecluster.ocs.openshift.io/ocs-storagecluster annotated
```

사전 요구 사항

- **OpenShift Container Storage** 클러스터가 정상 상태인지 확인합니다. 리소스 또는 노드가 부족하여 일부 Pod가 성공적으로 종료되지 않으면 설치 제거 프로세스가 실패할 수 있습니다. 클러스터가 비정상인 경우 **OpenShift Container Storage**를 설치 제거하기 전에 **Red Hat** 고객 지원팀에 문의하십시오.
- 애플리케이션이 **OpenShift Container Storage**에서 제공하는 스토리지 클래스를 사용하여 PVC(영구 볼륨 클레임) 또는 개체 버킷 클레임(OBC)을 사용하지 않는지 확인합니다.
- 관리자가 생성한 사용자 지정 리소스(예: 사용자 지정 스토리지 클래스, **cephblockpools**)가 관리자가 생성한 경우 해당 리소스를 사용하는 리소스를 제거한 후 관리자가 삭제해야 합니다.

절차

1. **OpenShift Container Storage**를 사용하는 볼륨 스냅샷을 삭제합니다.

- a. 모든 네임스페이스의 볼륨 스냅샷을 나열합니다.

```
$ oc get volumesnapshot --all-namespaces
```

- b. 이전 명령의 출력에서 **OpenShift Container Storage**를 사용하는 볼륨 스냅샷을 식별하고 삭제합니다.

```
$ oc delete volumesnapshot <VOLUME-SNAPSHOT-NAME> -n <NAMESPACE>
```

2. **OpenShift Container Storage**를 사용하는 **PVC** 및 **OBC**를 삭제합니다.

기본 설치 제거 모드(허용)에서 설치 제거 프로그램은 **OpenShift Container Storage**를 사용하는 모든 **PVC** 및 **OBC**가 삭제됩니다.

PVC를 삭제하지 않고 스토리지 클러스터를 삭제하려면 **uninstall** 모드 주석을 강제 설정하고 이 단계를 건너뛸 수 있습니다. 이렇게 하면 시스템의 고립된 **PVC**와 **OBC**가 생성됩니다.

- a. **OpenShift Container Storage**를 사용하여 **OpenShift Container Platform** 모니터링 스택 **PVC**를 삭제합니다.

자세한 내용은 [4.2절. “OpenShift Container Storage에서 모니터링 스택 제거”](#)의 내용을 참조하십시오.

- b. **OpenShift Container Storage**를 사용하여 **OpenShift Container Platform** 레지스트리 **PVC**를 삭제합니다.

자세한 내용은 [4.3절. “OpenShift Container Storage에서 OpenShift Container Platform 레지스트리 제거”](#)의 내용을 참조하십시오.

- c. **OpenShift Container Storage**를 사용하여 **OpenShift Container Platform** 로깅 **PVC**를 삭제합니다.

자세한 내용은 [4.4절. “OpenShift Container Storage에서 클러스터 로깅 Operator 제거”](#)의 내용을 참조하십시오.

d.

OpenShift Container Storage를 사용하여 프로비저닝된 다른 **PVC** 및 **OBC**를 삭제합니다.

•

다음 스크립트는 **OpenShift Container Storage**를 사용하여 프로비저닝된 **PVC** 및 **OBC**를 식별하는 샘플 스크립트입니다. 스크립트는 **OpenShift Container Storage**에서 내부적으로 사용하는 **PVC**를 무시합니다.

```
#!/bin/bash

RBD_PROVISIONER="openshift-storage.rbd.csi.ceph.com"
CEPHFS_PROVISIONER="openshift-storage.cephfs.csi.ceph.com"
NOOBAA_PROVISIONER="openshift-storage.noobaa.io/obc"
RGW_PROVISIONER="openshift-storage.ceph.rook.io/bucket"

NOOBAA_DB_PVC="noobaa-db"
NOOBAA_BACKINGSTORE_PVC="noobaa-default-backing-store-noobaa-pvc"

# Find all the OCS StorageClasses
OCS_STORAGECLASSES=$(oc get storageclasses | grep -e
"$RBD_PROVISIONER" -e "$CEPHFS_PROVISIONER" -e
"$NOOBAA_PROVISIONER" -e "$RGW_PROVISIONER" | awk '{print $1}')

# List PVCs in each of the StorageClasses
for SC in $OCS_STORAGECLASSES
do
    echo
    "=====
=="
    echo "$SC StorageClass PVCs and OBCs"
    echo
    "=====
=="
    oc get pvc --all-namespaces --no-headers 2>/dev/null | grep $SC | grep -v -e
"$NOOBAA_DB_PVC" -e "$NOOBAA_BACKINGSTORE_PVC"
    oc get obc --all-namespaces --no-headers 2>/dev/null | grep $SC
    echo
done
```



참고

클라우드 플랫폼용 **RGW_PROVISIONER** 생략.

•

OBC를 삭제합니다.

```
$ oc delete obc <obc name> -n <project name>
```

- PVC를 삭제합니다.

```
$ oc delete pvc <pvc name> -n <project-name>
```



참고

클러스터에서 생성된 사용자 정의 백업 저장소, 버킷 클래스 등을 제거했는지 확인합니다.

3. **Storage Cluster** 오브젝트를 삭제하고 관련 리소스가 제거될 때까지 기다립니다.

```
$ oc delete -n openshift-storage storagecluster --all --wait=true
```

4. **uninstall.ocs.openshift.io/cleanup-policy** 가 삭제(기본값)로 설정되어 있는지 확인하고 상태가 **Completed** 인지 확인합니다.

```
$ oc get pods -n openshift-storage | grep -i cleanup
NAME                                READY STATUS RESTARTS AGE
cluster-cleanup-job-<xx>            0/1   Completed 0      8m35s
cluster-cleanup-job-<yy>            0/1   Completed 0      8m35s
cluster-cleanup-job-<zz>            0/1   Completed 0      8m35s
```

5. **/var/lib/rook** 디렉토리가 비어 있는지 확인합니다. 이 디렉터리는 **uninstall.ocs.openshift.io/cleanup-policy** 주석이 삭제(기본값)로 설정된 경우에만 비어 있습니다.

```
$ for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o jsonpath='{.items[*].metadata.name}'); do oc debug node/${i} -- chroot /host ls -l /var/lib/rook; done
```

6. 설치 시 암호화가 활성화된 경우 모든 **OpenShift Container Storage** 노드의 **OSD** 장치에서 **dm-crypt managed device-mapper** 매핑을 제거합니다.

- a. 스토리지 노드의 호스트에 디버그 포트 및 **chroot** 를 만듭니다.

```
$ oc debug node/<node name>
$ chroot /host
```

U.

장치 이름을 가져오고 **OpenShift Container Storage** 장치를 기록합니다.

```
$ dmsetup ls
ocs-deviceset-0-data-0-57snx-block-dmccrypt (253:1)
```

C.

매핑된 장치를 제거합니다.

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-0-data-0-57snx-block-dmccrypt
```



참고

권한이 충분하지 않아 위의 명령이 중단되면 다음 명령을 실행합니다.

- **CTRL+Z** 를 눌러 위의 명령을 종료합니다.

- 중단된 프로세스의 **PID**를 찾습니다.

```
$ ps -ef | grep crypt
```

- **kill** 명령을 사용하여 프로세스를 종료합니다.

```
$ kill -9 <PID>
```

- 장치 이름이 제거되었는지 확인합니다.

```
$ dmsetup ls
```

7.

네임스페이스를 삭제하고 삭제가 완료될 때까지 기다립니다. **openshift-storage** 가 활성 프로젝트인 경우 다른 프로젝트로 전환해야 합니다.

예를 들면 다음과 같습니다.

```
$ oc project default
$ oc delete project openshift-storage --wait=true --timeout=5m
```


다음 명령에서 **NotFound** 오류를 반환하면 프로젝트가 삭제됩니다.

```
$ oc get project openshift-storage
```



참고

OpenShift Container Storage를 설치 제거하는 동안 네임스페이스가 완전히 삭제되지 않고 종료 상태가 유지되는 경우 **설치 제거 중 나머지 리소스 문제 해결 및 삭제** 단계를 수행하여 네임스페이스가 종료되지 않는 오브젝트를 식별합니다.

8. 로컬 스토리지 장치를 사용하여 **OpenShift Container Storage**를 배포한 경우 로컬 스토리지 운영자 구성을 삭제합니다. [로컬 스토리지 운영자 구성 제거](#)를 참조하십시오.

9. 스토리지 노드의 레이블을 해제합니다.

```
$ oc label nodes --all cluster.ocs.openshift.io/openshift-storage-
$ oc label nodes --all topology.rook.io/rack-
```

10. 노드가 테인트된 경우 **OpenShift Container Storage** 테인트를 제거합니다.

```
$ oc adm taint nodes --all node.ocs.openshift.io/storage-
```

11. **OpenShift Container Storage**를 사용하여 프로비저닝된 모든 **PV**가 삭제되었는지 확인합니다. **Released** 상태에 **PV**가 남아 있으면 삭제합니다.

```
$ oc get pv
$ oc delete pv <pv name>
```

12. **Multicloud Object Gateway** 스토리지 클래스를 삭제합니다.

```
$ oc delete storageclass openshift-storage.noobaa.io --wait=true --timeout=5m
```

13. **CustomResourceDefinitions** 제거.

```
$ oc delete crd backingstores.noobaa.io bucketclasses.noobaa.io
cephblockpools.ceph.rook.io cephclusters.ceph.rook.io cephfilesystems.ceph.rook.io
```

```
cephnfses.ceph.rook.io cephobjectstores.ceph.rook.io cephobjectstoreusers.ceph.rook.io
noobaas.noobaa.io ocsinitializations.ocs.openshift.io storageclusters.ocs.openshift.io
cephclients.ceph.rook.io cephobjectrealms.ceph.rook.io cephobjectzonegroups.ceph.rook.io
cephobjectzones.ceph.rook.io cephrbdmirrors.ceph.rook.io --wait=true --timeout=5m
```

14.

선택 사항: 자격 증명 모음 키가 영구적으로 삭제되도록 하려면 자격 증명 모음 키와 연결된 메타데이터를 수동으로 삭제해야 합니다.



참고

Vault 키가 삭제된 것으로 표시되고 **OpenShift Container Storage** 설치 중에 영구적으로 삭제되지 않기 때문에 **Vault** 키가 **KMS(Key Management System)**를 사용한 클러스터 전체 암호화에 사용되는 경우에만 이 단계를 실행합니다. 필요한 경우 나중에 언제든지 복원할 수 있습니다.

a.

자격 증명 모음의 키를 나열합니다.

```
$ vault kv list <backend_path>
```

<backend_path>

암호화 키가 저장된 자격 증명 모음의 경로입니다.

예를 들면 다음과 같습니다.

```
$ vault kv list kv-v2
```

출력 예:

```
Keys
-----
NOOBAA_ROOT_SECRET_PATH/
rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
rook-ceph-osd-encryption-key-ocs-deviceset-thin-1-data-0sq227
rook-ceph-osd-encryption-key-ocs-deviceset-thin-2-data-0xzszb
```

b.

자격 증명 모음 키와 연결된 메타데이터를 나열합니다.

```
$ vault kv get kv-v2/<key>
```

MCG(Multicloud Object Gateway) 키의 경우 다음을 수행합니다.

```
$ vault kv get kv-v2/NOOBAA_ROOT_SECRET_PATH/<key>
```

<key>

암호화 키입니다.

예를 들면 다음과 같습니다.

```
$ vault kv get kv-v2/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
```

출력 예:

```
=====  
=====  
Key      Value  
---      -  
created_time  2021-06-23T10:06:30.650103555Z  
deletion_time 2021-06-23T11:46:35.045328495Z  
destroyed     false  
version       1
```

c.

메타데이터를 삭제합니다.

```
$ vault kv metadata delete kv-v2/<key>
```

MCG 키의 경우:

```
$ vault kv metadata delete kv-v2/NOOBAA_ROOT_SECRET_PATH/<key>
```

<key>

암호화 키입니다.

예를 들면 다음과 같습니다.

```
$ vault kv metadata delete kv-v2/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-  
data-0m27q8
```

출력 예:

```
Success! Data deleted (if it existed) at: kv-v2/metadata/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
```

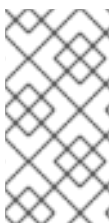
- d. 이러한 단계를 반복하여 모든 자격 증명 모음 키와 연결된 메타데이터를 삭제합니다.

15. **OpenShift Container Storage**가 완전히 설치 제거되었는지 확인하려면 **OpenShift Container Platform Web Console**에서 다음을 수행합니다.

- a. **Storage**(스토리지)를 클릭합니다.
- b. **Storage**(스토리지)에 **Overview** (개요)가 더 이상 표시되지 않는지 확인합니다.

4.1.1. 로컬 스토리지 Operator 구성 제거

로컬 스토리지 장치를 사용하여 **OpenShift Container Storage**를 배포한 경우에만 이 섹션의 지침을 사용합니다.



참고

localvolume 리소스만 사용하는 **OpenShift Container Storage** 배포의 경우 8단계를 참조하십시오.

절차

1. **LocalVolumeSet** 및 **OpenShift Container Storage**에서 사용 중인 해당 **StorageClassName** 을 식별합니다.
2. **LocalVolumeSet** 을 제공하는 **StorageClass** 로 **SC** 변수를 설정합니다.

```
$ export SC="<StorageClassName>"
```

3. **LocalVolumeSet** 을 삭제합니다.

```
$ oc delete localvolumesets.local.storage.openshift.io <name-of-volumeset> -n openshift-
local-storage
```

4. 지정된 **StorageClassName** 의 로컬 스토리지 **PV**를 삭제합니다.

```
$ oc get pv | grep $SC | awk '{print $1}' | xargs oc delete pv
```

5. **StorageClassName** 을 삭제합니다.

```
$ oc delete sc $SC
```

6. **LocalVolumeSet** 에서 생성한 심볼릭 링크를 삭제합니다.

```
[[ ! -z $SC ]] && for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o
jsonpath='{.items[*].metadata.name}'); do oc debug node/${i} -- chroot /host rm -rfv
/mnt/local-storage/${SC}/; done
```

7. **LocalVolumeDiscovery** 삭제.

```
$ oc delete localvolumediscovery.local.storage.openshift.io/auto-discover-devices -n
openshift-local-storage
```

8. **LocalVolume** 리소스 제거(있는 경우).

다음 단계를 사용하여 현재 또는 이전 **OpenShift Container Storage** 버전에서 **PV**를 프로비저닝하는 데 사용되는 **LocalVolume** 리소스를 제거합니다. 또한 클러스터의 다른 테넌트에서 이러한 리소스를 사용하지 않는지 확인합니다.

각 로컬 볼륨에 대해 다음을 수행합니다.

- a. **OpenShift Container Storage**에서 사용하는 **LocalVolume** 및 해당 **StorageClassName** 을 식별합니다.

- b. 변수 **LV**를 **LocalVolume**의 이름으로 설정하고 변수 **SC**를 **StorageClass**의 이름으로 설정합니다.

예를 들면 다음과 같습니다.

```
$ LV=local-block
$ SC=localblock
```

c.

로컬 볼륨 리소스를 삭제합니다.

```
$ oc delete localvolume -n local-storage --wait=true $LV
```

d.

나머지 **PV** 및 스토리지 클래스가 있는 경우 삭제합니다.

```
$ oc delete pv -l storage.openshift.com/local-volume-owner-name=${LV} --wait --
timeout=5m
$ oc delete storageclass $SC --wait --timeout=5m
```

e.

해당 리소스의 스토리지 노드에서 아티팩트를 정리합니다.

```
$ [[ ! -z $SC ]] && for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o
jsonpath='{ .items[*].metadata.name }'); do oc debug node/${i} -- chroot /host rm -rfv
/mnt/local-storage/${SC}/; done
```

출력 예:

```
Starting pod/node-xxx-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
Starting pod/node-yyy-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
Starting pod/node-zzz-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
```

4.2. OPENSIFT CONTAINER STORAGE에서 모니터링 스택 제거

이 섹션을 사용하여 **OpenShift Container Storage**에서 모니터링 스택을 정리합니다.

모니터링 스택 구성의 일부로 생성된 **PVC**는 **openshift-monitoring** 네임스페이스에 있습니다.

사전 요구 사항

- **PVC**는 **OpenShift Container Platform** 모니터링 스택을 사용하도록 구성됩니다.

자세한 내용은 [모니터링 스택 구성](#)을 참조하십시오.

절차

1.

openshift-monitoring 네임스페이스에서 현재 실행 중인 **Pod** 및 **PVC**를 나열합니다.

```
$ oc get pod,pvc -n openshift-monitoring
```

NAME	READY	STATUS	RESTARTS	AGE
pod/alertmanager-main-0	3/3	Running	0	8d
pod/alertmanager-main-1	3/3	Running	0	8d
pod/alertmanager-main-2	3/3	Running	0	8d
pod/cluster-monitoring-operator-84457656d-pkrxm	1/1	Running	0	8d
pod/grafana-79ccf6689f-2ll28	2/2	Running	0	8d
pod/kube-state-metrics-7d86fb966-rvd9w	3/3	Running	0	8d
pod/node-exporter-25894	2/2	Running	0	8d
pod/node-exporter-4dsd7	2/2	Running	0	8d
pod/node-exporter-6p4zc	2/2	Running	0	8d
pod/node-exporter-jbjvg	2/2	Running	0	8d
pod/node-exporter-jj4t5	2/2	Running	0	6d18h
pod/node-exporter-k856s	2/2	Running	0	6d18h
pod/node-exporter-rf8gn	2/2	Running	0	8d
pod/node-exporter-rmb5m	2/2	Running	0	6d18h
pod/node-exporter-zj7kx	2/2	Running	0	8d
pod/openshift-state-metrics-59dbd4f654-4clng	3/3	Running	0	8d
pod/prometheus-adapter-5df5865596-k8dzn	1/1	Running	0	7d23h
pod/prometheus-adapter-5df5865596-n2gj9	1/1	Running	0	7d23h
pod/prometheus-k8s-0	6/6	Running	1	8d
pod/prometheus-k8s-1	6/6	Running	1	8d
pod/prometheus-operator-55cfb858c9-c4zd9	1/1	Running	0	6d21h
pod/telemeter-client-78fc8fc97d-2rgfp	3/3	Running	0	8d

NAME	STATUS	VOLUME
------	--------	--------

CAPACITY	ACCESS MODES	STORAGECLASS	AGE
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-0	Bound	pvc-0d519c4f-15a5-11ea-baa0-026d231574aa	40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-1	Bound	pvc-0d5a9825-15a5-11ea-baa0-026d231574aa	40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-2	Bound	pvc-0d6413dc-15a5-11ea-baa0-026d231574aa	40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-0	Bound	pvc-0b7c19b0-15a5-11ea-baa0-026d231574aa	40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-1	Bound	pvc-0b8aed3f-15a5-11ea-baa0-026d231574aa	40Gi RWO ocs-storagecluster-ceph-rbd 8d

2.

모니터링 구성 맵을 편집합니다.

```
$ oc -n openshift-monitoring edit configmap cluster-monitoring-config
```

3.

다음 예제와 같이 **OpenShift Container Storage** 스토리지 클래스를 참조하는 **config** 섹션을 제거하여 저장합니다.

편집하기 전에


```
.  
. .  
apiVersion: v1  
data:  
  config.yaml: |  
    alertmanagerMain:  
      volumeClaimTemplate:  
        metadata:  
          name: my-alertmanager-claim  
        spec:  
          resources:  
            requests:  
              storage: 40Gi  
          storageClassName: ocs-storagecluster-ceph-rbd  
  prometheusK8s:  
    volumeClaimTemplate:  
      metadata:  
        name: my-prometheus-claim  
      spec:  
        resources:  
          requests:  
            storage: 40Gi  
        storageClassName: ocs-storagecluster-ceph-rbd  
kind: ConfigMap  
metadata:  
  creationTimestamp: "2019-12-02T07:47:29Z"  
  name: cluster-monitoring-config  
  namespace: openshift-monitoring  
  resourceVersion: "22110"  
  selfLink: /api/v1/namespaces/openshift-monitoring/configmaps/cluster-monitoring-config  
  uid: fd6d988b-14d7-11ea-84ff-066035b9efa8  
. . .
```

편집 후

```

.
.
.
apiVersion: v1
data:
  config.yaml: |
kind: ConfigMap
metadata:
  creationTimestamp: "2019-11-21T13:07:05Z"
  name: cluster-monitoring-config
  namespace: openshift-monitoring
  resourceVersion: "404352"
  selfLink: /api/v1/namespaces/openshift-monitoring/configmaps/cluster-monitoring-config
  uid: d12c796a-0c5f-11ea-9832-063cd735b81c
.
.
.

```

이 예에서는 **alertmanagerMain** 및 **prometheusK8s** 모니터링 구성 요소가 **OpenShift Container Storage PVC**를 사용하고 있습니다.

4. 관련 **PVC**를 삭제합니다. 스토리지 클래스를 사용하는 모든 **PVC**를 삭제하십시오.

```
$ oc delete -n openshift-monitoring pvc <pvc-name> --wait=true --timeout=5m
```

4.3. OPENSIFT CONTAINER STORAGE에서 OPENSIFT CONTAINER PLATFORM 레지스트리 제거

이 섹션을 사용하여 **OpenShift Container Storage**에서 **OpenShift Container Platform** 레지스트리를 정리합니다. 대체 스토리지를 구성하려면 [이미지 레지스트리](#)를 참조하십시오.

OpenShift Container Platform 레지스트리 구성의 일부로 생성된 **PVC**는 **openshift-image-registry** 네임스페이스에 있습니다.

사전 요구 사항

- **OpenShift Container Storage PVC**를 사용하도록 이미지 레지스트리가 구성되어 있어야 합니다.

절차

1.

configs.imageregistry.operator.openshift.io 오브젝트를 편집하고 스토리지 섹션에서 콘텐츠를 제거합니다.

```
$ oc edit configs.imageregistry.operator.openshift.io
```

편집하기 전에

```
.
.
.
storage:
  pvc:
    claim: registry-cephfs-rwx-pvc
.
.
.
```

편집 후

```
.
.
.
storage:
  emptyDir: {}
.
.
.
```

이 예에서는 PVC를 **registry-cephfs-rwx-pvc** 라고 하며 이제 안전하게 삭제할 수 있습니다.

2.

PVC를 삭제합니다.

```
$ oc delete pvc <pvc-name> -n openshift-image-registry --wait=true --timeout=5m
```

4.4. OPENSIFT CONTAINER STORAGE에서 클러스터 로깅 OPERATOR 제거

OpenShift Container Storage에서 클러스터 로깅 Operator를 정리하려면 절차의 단계를 따르십시오.

클러스터 로깅 Operator 구성의 일부로 생성된 PVC는 **openshift-logging** 네임스페이스에 있습니다.

사전 요구 사항

- **OpenShift Container Storage PVC**를 사용하려면 클러스터 로깅 인스턴스를 구성해야 합니다.

절차

1. 네임스페이스에서 **ClusterLogging** 인스턴스를 제거합니다.

```
$ oc delete clusterlogging instance -n openshift-logging --wait=true --timeout=5m
```

이제 **openshift-logging** 네임스페이스의 **PVC**를 삭제할 수 있습니다.

2. **PVC**를 삭제합니다.

```
$ oc delete pvc <pvc-name> -n openshift-logging --wait=true --timeout=5m
```