Red Hat OpenShift Data Foundation 4.11

OpenShift Data Foundation 문제 해결

OpenShift Data Foundation 문제 해결 방법

Last Updated: 2023-10-05

Red Hat OpenShift Data Foundation 4.11 OpenShift Data Foundation 문제 해결

OpenShift Data Foundation 문제 해결 방법

법적 공지

초록

Red Hat OpenShift Data Foundation 문제 해결에 대한 자세한 내용은 이 문서를 참조하십시오.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체 3

RED HAT 문서에 관한 피드백 제공 4

1장. 개요 5

2장. MUST-GATHER를 사용하여 로그 파일 및 진단 정보 다운로드 6

3장. 문제 해결에 일반적으로 필요한 로그 9

4장. OPENSHIFT DATA FOUNDATION 게시 배포의 클러스터 수준 기본 노드 선택기 덮어쓰기 12

5장. 암호화 토큰이 삭제되거나 만료됨 13

6장. OPENSHIFT DATA FOUNDATION에서 경고 및 오류 문제 해결 14
6.1. 경고 및 오류 해결 14
6.2. 클러스터 상태 문제 해결 22
6.3. NOOBAA BUCKET 오류 상태 해결 22
6.4. NOOBAA BUCKET EXCEEDING QUOTA STATE 해결 23
6.5. NOOBAA BUCKET 용량 또는 할당량 상태 해결 24
6.6. POD 복구 24
6.7. EBS 볼륨 분리에서 복구 24
6.8. ROOK-CEPH-OPERATOR의 디버그 로그 활성화 및 비활성화 24

7장. LOCAL STORAGE OPERATOR 배포 확인 26

8장. 실패한 CEPH OBJECT STORAGE 장치 제거 27
8.1. CEPH 클러스터 상태 확인 27
8.2. 동적으로 프로비저닝된 RED HAT OPENSHIFT DATA FOUNDATION에서 실패 또는 원하지 않는 CEPH OSD 제거 27
8.3. 로컬 스토리지 장치를 사용하여 프로비저닝된 CEPH OSD 또는 실패한 CEPH OSD 제거 29
8.4. 실패하거나 원하지 않는 CEPH OSD를 제거하는 동안 CEPHOSD:OSD.0 오류 문제 해결 31

9장. 설치 제거 중 나머지 리소스 문제 해결 및 삭제 32

10장. 외부 모드에서 CEPHFS PVC 생성 문제 해결 34

11장. OPENSHIFT DATA FOUNDATION에서 모니터 POD 복원 37
11.1. MULTICLOUD 오브젝트 게이트웨이 복원 43

12장. OPENSHIFT DATA FOUNDATION에서 CEPH-MONITOR 쿼럼 복원 45

13장. RED HAT OPENSHIFT DATA FOUNDATION 콘솔 플러그인 활성화 51

14장. OPENSHIFT DATA FOUNDATION 구성 요소에 대한 리소스 변경 53
14.1. ROOK-CEPH POD에서 CPU 및 메모리 리소스 변경 53
14.2. MCG의 리소스 튜닝 54

15장. 글로벌 POD 네트워킹을 수동으로 활성화하여 OVS-MULTITENANT 플러그인을 사용하여 ODF-CONSOLE 에 액세스 56

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. 개선할 내용에 대해 알려주십시오. 피드백을 보내주시려면 다음을 확인하십시오.

- 특정 문구에 대한 간단한 의견 작성 방법은 다음과 같습니다.
  1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **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 이동합니다.
  2. **구성 요소** 섹션에서 **설명서를** 선택합니다.
  3. **설명** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
  4. **버그 제출**을 클릭합니다.

1장. 개요

OpenShift Data Foundation 문제 해결은 관리자가 Red Hat OpenShift Data Foundation 클러스터의 문제를 해결하고 수정하는 방법을 이해하는 데 도움이 되도록 작성되었습니다.

대부분의 문제 해결 작업은 수정 또는 해결 방법에 중점을 둡니다. 이 문서는 관리자가 발생할 수 있는 오류를 기반으로 한 장으로 나뉩니다.

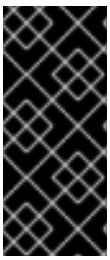
- 2장. *must-gather를 사용하여 로그 파일 및 진단 정보 다운로드* OpenShift Data Foundation에서 must-gather 유틸리티를 사용하는 방법을 보여줍니다.
- 3장. *문제 해결에 일반적으로 필요한 로그* OpenShift Data Foundation에 대해 일반적으로 필요한 로그 파일을 가져오는 방법을 보여줍니다.
- 6장. *OpenShift Data Foundation에서 경고 및 오류 문제 해결* 발생한 오류를 식별하고 필요한 작업을 수행하는 방법을 보여줍니다.

주의

Red Hat은 잘못된 명령을 실행할 때 데이터가 손실될 수 있으므로 OpenShift Data Foundation 클러스터에서 Ceph 명령 실행을 지원하지 않습니다(Red Hat 지원 또는 Red Hat 설명서에 명시되어 있지 않음). 이 경우 Red Hat 지원팀은 상업적으로 합리적인 노력으로만 제공할 수 있으며 데이터 손실 시 모든 데이터를 복원할 수 없습니다.

2장. MUST-GATHER를 사용하여 로그 파일 및 진단 정보 다운로드

Red Hat OpenShift Data Foundation이 문제를 자동으로 해결할 수 없는 경우 사용자 또는 Red Hat 지원이 문제를 검토하고 솔루션을 확인할 수 있도록 **must-gather** 툴을 사용하여 로그 파일 및 진단 정보를 수집합니다.



중요

Red Hat OpenShift Data Foundation이 외부 모드로 배포되면 **must-gather** 는 OpenShift Data Foundation 클러스터에서만 로그를 수집하고 외부 Red Hat Ceph Storage 클러스터에서 디버그 데이터 및 로그를 수집하지 않습니다. 외부 Red Hat Ceph Storage 클러스터에서 디버그 로그를 수집하려면 Red Hat Ceph Storage 문제 해결 가이드 를 참조하고 Red Hat Ceph Storage Administrator에 문의하십시오.

사전 요구 사항

- 선택 사항: OpenShift Data Foundation이 연결이 끊긴 환경에 배포된 경우 개별 **must-gather** 이미지를 연결이 끊긴 환경에서 사용 가능한 미러 레지스트리에 미러링해야 합니다.

```
$ oc image mirror registry.redhat.io/odf4/ocs-must-gather-rhel8:v4.11 <local-registry>/odf4/ocs-must-gather-rhel8:v4.11 [--registry-config=<path-to-the-registry-config>] [--insecure=true]
```

<local-registry>

연결이 끊긴 OpenShift Container Platform 클러스터에 사용 가능한 로컬 이미지 미러 레지스트리입니다.

<path-to-the-registry-config>

레지스트리 자격 증명의 경로입니다. 기본값은 **~/.docker/config.json** 입니다.

--insecure

미러 레지스트리가 비보안인 경우에만 이 플래그를 추가합니다.

자세한 내용은 Red Hat 지식베이스 솔루션을 참조하십시오.

  - Redhat Openshift 레지스트리 간 이미지를 미러링하는 방법
  - 개인 레지스트리가 비보안인 경우 OpenShift 이미지 저장소를 미러링하지 못했습니다.

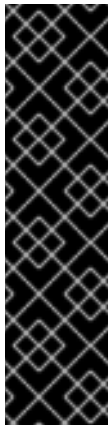
절차

- OpenShift Data Foundation 클러스터에 연결된 클라이언트에서 **must-gather** 명령을 실행합니다.

```
$ oc adm must-gather --image=registry.redhat.io/odf4/ocs-must-gather-rhel8:v4.11 --dest-dir=<directory-name>
```

<directory-name>

데이터를 쓸 디렉터리의 이름입니다.Is the name of the directory where you want to write the data to.



중요

연결이 끊긴 환경 배포의 경우 **--image** 매개변수의 이미지를 미러링된 **must-gather** 이미지로 교체합니다.

```
$ oc adm must-gather --image=<local-registry>/odf4/ocs-must-gather-rhel8:v4.11 --dest-dir=<directory-name>
```

<local-registry>

연결이 끊긴 OpenShift Container Platform 클러스터에 사용 가능한 로컬 이미지 미러 레지스트리입니다.

이 명령은 지정된 디렉터리에서 다음 정보를 수집합니다.

- 모든 Red Hat OpenShift Data Foundation 클러스터 관련 CR(사용자 정의 리소스)과 네임스페이스.
- 모든 Red Hat OpenShift Data Foundation 관련 Pod 로그입니다.
- Status, Cluster health 및 기타와 같은 일부 표준 Ceph 명령의 출력입니다.

명령 변경

- 하나 이상의 마스터 노드가 **Ready** 상태가 아닌 경우 **--node-name** 을 사용하여 **must-gather** Pod를 안전하게 예약할 수 있도록 **준비** 중인 마스터 노드를 제공합니다.

```
$ oc adm must-gather --image=registry.redhat.io/odf4/ocs-must-gather-rhel8:v4.11 --dest-dir=_<directory-name>_ --node-name=_<node-name>_
```

- 특정 시간에서 정보를 수집하려는 경우:
  - 수집된 로그의 상대적 기간을 5초 또는 2일 이내에 지정하려면=< **duration> 이후 /usr/bin/gather** 를 추가합니다.

```
$ oc adm must-gather --image=registry.redhat.io/odf4/ocs-must-gather-rhel8:v4.11 --dest-dir=_<directory-name>_ /usr/bin/gather since=<duration>
```

  - 로그를 수집할 특정 시간을 지정하려면 **/usr/bin/gather since-time=<rfc3339-timestamp** > 를 추가합니다.

```
$ oc adm must-gather --image=registry.redhat.io/odf4/ocs-must-gather-rhel8:v4.11 --dest-dir=_<directory-name>_ /usr/bin/gather since-time=<rfc3339-timestamp>
```

다음과 같이 이러한 명령에서 예제 값을 바꿉니다.

<node-name>

하나 이상의 마스터 노드가 **Ready** 상태가 아닌 경우 이 매개변수를 사용하여 여전히 **Ready** 상태에 있는 마스터 노드의 이름을 제공합니다. 이렇게 하면 **must-gather** Pod가 준비되지 않은 마스터 노드에 예약되지 않도록 하여 예약 오류가 발생하지 않습니다.

<directory-name>

must-gather 에서 수집한 정보를 저장할 디렉터리입니다.

<duration>

정보를 상대 기간(예: **5** 시간 전부터 시작)으로 수집할 기간을 지정합니다.

<rfc3339-timestamp>

정보를 수집할 시간을 RFC 3339 타임 스탬프 (예: **2020-11-10T04:00:00+00:00**:00:00:00)로 설정합니다(마지 2020년 11월 11일 오전 4시부터 시작).

3장. 문제 해결에 일반적으로 필요한 로그

OpenShift Data Foundation 문제 해결에 일반적으로 사용되는 로그 중 일부가 나열되며 해당 로그를 생성하는 명령과 함께 나열됩니다.

- 특정 Pod의 로그를 생성합니다.

```
$ oc logs <pod-name> -n <namespace>
```

- Ceph 또는 OpenShift Data Foundation 클러스터에 대한 로그를 생성합니다.

```
$ oc logs rook-ceph-operator-<ID> -n openshift-storage
```

중요

현재 rook-ceph-operator 로그는 오류에 대한 정보를 제공하지 않으며 이 로그는 문제 해결에 제한적인 역할을 하며 rook-ceph-operator의 디버그 로그 활성화 및 비활성화 를 참조하십시오.

- app-pod의 PVC 마운트에서 문제를 감지하기 위해 cephfs 또는 rbd와 같은 플러그인 Pod에 대한 로그를 생성합니다.

```
$ oc logs csi-cephfsplugin-<ID> -n openshift-storage -c csi-cephfsplugin
```

```
$ oc logs csi-rbdplugin-<ID> -n openshift-storage -c csi-rbdplugin
```

  - CSI Pod의 모든 컨테이너에 대한 로그를 생성하려면 다음을 수행합니다.

```
$ oc logs csi-cephfsplugin-<ID> -n openshift-storage --all-containers
```

```
$ oc logs csi-rbdplugin-<ID> -n openshift-storage --all-containers
```

- PVC가 **BOUND** 상태가 아닌 경우 문제를 감지하기 위해 cephfs 또는 rbd 프로비저너 Pod에 대한 로그를 생성합니다.

```
$ oc logs csi-cephfsplugin-provisioner-<ID> -n openshift-storage -c csi-cephfsplugin
```

```
$ oc logs csi-rbdplugin-provisioner-<ID> -n openshift-storage -c csi-rbdplugin
```

  - CSI Pod의 모든 컨테이너에 대한 로그를 생성하려면 다음을 수행합니다.

```
$ oc logs csi-cephfsplugin-provisioner-<ID> -n openshift-storage --all-containers
```

```
$ oc logs csi-rbdplugin-provisioner-<ID> -n openshift-storage --all-containers
```

- cluster-info 명령을 사용하여 OpenShift Data Foundation 로그를 생성합니다.

```
$ oc cluster-info dump -n openshift-storage --output-directory=<directory-name>
```

- Local Storage Operator를 사용하는 경우 cluster-info 명령을 사용하여 로그를 생성할 수 있습니다.

```
$ oc cluster-info dump -n openshift-local-storage --output-directory=<directory-name>
```

- OpenShift Data Foundation Operator 로그 및 이벤트를 확인합니다.
  - Operator 로그를 확인하려면 다음을 수행합니다.

```
# oc logs <ocs-operator> -n openshift-storage
```

<ocs-operator>

```
# oc get pods -n openshift-storage | grep -i "ocs-operator" | awk '{print $1}'
```

  - Operator 이벤트를 확인하려면 다음을 수행합니다.

```
# oc get events --sort-by=metadata.creationTimestamp -n openshift-storage
```

- OpenShift Data Foundation Operator 버전 및 채널을 가져옵니다.

```
# oc get csv -n openshift-storage
```

출력 예:

```
NNAME                         DISPLAY                       VERSION   REPLACES   PHASE
mcg-operator.v4.11.0          NooBaa Operator               4.11.0               Succeeded
ocs-operator.v4.11.0          OpenShift Container Storage   4.11.0               Succeeded
odf-csi-addons-operator.v4.11.0   CSI Addons                    4.11.0               Succeeded
odf-operator.v4.11.0          OpenShift Data Foundation     4.11.0               Succeeded
```

```
# oc get subs -n openshift-storage
```

출력 예:

```
NAME                                              PACKAGE                SOURCE
CHANNEL
mcg-operator-stable-4.11-redhat-operators-openshift-marketplace   mcg-operator
redhat-operators   stable-4.11
ocs-operator-stable-4.11-redhat-operators-openshift-marketplace   ocs-operator
redhat-operators   stable-4.11
odf-csi-addons-operator                                odf-csi-addons-operator   redhat-operators
stable-4.11
odf-operator                                           odf-operator           redhat-operators   stable-
4.11
```

- 설치 계획이 생성되었는지 확인합니다.

```
# oc get installplan -n openshift-storage
```

- OpenShift Data Foundation을 업데이트한 후 구성 요소의 이미지를 확인합니다.
  - 이미지가 실행 중인지 확인하는 구성 요소의 노드를 확인합니다.

- 이미지가 실행 중인지 확인하려는 구성 요소의 노드를 확인합니다.

```
# oc get pods -o wide | grep <component-name>
```

예를 들면 다음과 같습니다.

```
# oc get pods -o wide | grep rook-ceph-operator
```

출력 예:

```
rook-ceph-operator-566cc677fd-bjqnb 1/1 Running 20 4h6m 10.128.2.5 rook-ceph-operator-566cc677fd-bjqnb 1/1 Running 20 4h6m 10.128.2.5 dell-r440-12.gsslab.pnq2.redhat.com <none> <none>

<none> <none>
```

Dell-r440-12.gsslab.pnq2.redhat.com 은 **node-name** 입니다.

- 이미지 ID를 확인합니다.

```
# oc debug node/<node name>
```

<node-name>

이미지가 실행 중인지 확인하려는 구성 요소의 Pod가 실행되는 노드의 이름입니다.

```
# chroot /host
```

```
# crictl images | grep <component>
```

예를 들면 다음과 같습니다.

```
# crictl images | grep rook-ceph
```

IMAGEID 를 기록해 두고 Rook Ceph Operator 페이지의 **다이제스트** ID에 매핑합니다.

추가 리소스

- must-gather 사용

4장. OPENSHIFT DATA FOUNDATION 게시 배포의 클러스터 수준 기본 노드 선택기 덮어쓰기

OpenShift Data Foundation에 클러스터 수준 기본 노드 선택기를 사용하는 경우 CSI 데몬 세트에서 생성된 Pod는 선택기와 일치하는 노드에서만 시작할 수 있습니다. 선택기와 일치하지 않는 노드의 OpenShift Data Foundation을 사용하려면 명령줄 인터페이스에서 다음 단계를 수행하여 **클러스터 전체 기본 노드 선택기** 를 재정의합니다.

절차

1. openshift-storage 네임스페이스의 빈 노드 선택기를 지정합니다.

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

2. DaemonSet에서 생성한 원래 Pod를 삭제합니다.

```
oc delete pod -l app=csi-cephfsplugin -n openshift-storage
oc delete pod -l app=csi-rbdplugin -n openshift-storage
```

5장. 암호화 토큰이 삭제되거나 만료됨

키 관리 시스템의 암호화 토큰이 삭제되거나 만료되는 경우 이 절차를 사용하여 토큰을 업데이트합니다.

사전 요구 사항

- 삭제되거나 만료된 토큰과 동일한 정책이 있는 새 토큰이 있는지 확인합니다.

절차

1. OpenShift Container Platform 웹 콘솔에 로그인합니다.
2. **워크로드 → 시크릿** 을 클릭합니다.
3. 클러스터 전체 암호화에 사용되는 **ocs-kms-token** 을 업데이트하려면 다음을 수행합니다.
   a. **프로젝트를 openshift-storage** 로 설정합니다.
   b. **ocs-kms-token → 작업 → 시크릿 편집** 을 클릭합니다.
   c. 암호화 토큰 파일을 **Value** 필드에 드래그 앤 드롭하거나 업로드합니다. 토큰은 복사 및 붙여넣을 수 있는 파일 또는 텍스트일 수 있습니다.
   d. **저장**을 클릭합니다.
4. 암호화된 영구 볼륨이 있는 지정된 프로젝트 또는 네임스페이스의 **ceph-csi-kms-token** 을 업데이트하려면 다음을 수행합니다.
   a. 필요한 **프로젝트를** 선택합니다.
   b. **ceph-csi-kms-token → 작업 → 시크릿 편집** 을 클릭합니다.
   c. 암호화 토큰 파일을 **Value** 필드에 드래그 앤 드롭하거나 업로드합니다. 토큰은 복사 및 붙여넣을 수 있는 파일 또는 텍스트일 수 있습니다.
   d. **저장**을 클릭합니다.

참고

토큰은 **ceph-csi-kms-token** 을 사용하여 암호화된 모든 PVC를 삭제한 경우에만 삭제할 수 있습니다.

6장. OPENSHIFT DATA FOUNDATION에서 경고 및 오류 문제 해결

6.1. 경고 및 오류 해결

Red Hat OpenShift Data Foundation은 여러 일반적인 오류 시나리오를 탐지하고 자동으로 해결할 수 있습니다. 그러나 일부 문제는 관리자의 개입이 필요합니다.

현재 실행 중인 오류를 확인하려면 다음 위치 중 하나를 확인하십시오.

- **관찰 → 경고 →** 반복 옵션
- **홈 → 개요 → 클러스터** 탭
- **Storage → Data Foundation → Storage System →** Storage System → pop up → **Overview → Block and File** 탭의 *스토리지 시스템* 링크
- **Storage → Data Foundation → Storage System →** *storage system* link in the pop up → **Overview → Object** tab

표시된 오류를 복사하고 다음 섹션에서 검색하여 심각도 및 해결 방법을 확인합니다.

이름:CephMonVersionMismatch

Message:실행중인 여러 버전의 스토리지 서비스가 있습니다.

설명:{{ $value }} 다른 버전의 Ceph Mon 구성 요소가 실행 중입니다.

심각도: 경고

해결 방법 : 수정

프로시저: 사용자 인터페이스 및 로그를 검사하고 업데이트가 진행 중인지 확인합니다.

- 업데이트 진행 중인 경우 이 경고는 일시적인 경고입니다.
- 업데이트가 진행 중이 아니면 업그레이드 프로세스를 다시 시작합니다.

이름:CephOSDVersionMismatch

Message:실행중인 여러 버전의 스토리지 서비스가 있습니다.

설명:{{ $value }} 서로 다른 버전의 Ceph OSD 구성 요소가 실행 중입니다.

심각도: 경고

해결 방법 : 수정

프로시저: 사용자 인터페이스 및 로그를 검사하고 업데이트가 진행 중인지 확인합니다.

- 업데이트 진행 중인 경우 이 경고는 일시적인 경고입니다.
- 업데이트가 진행 중이 아니면 업그레이드 프로세스를 다시 시작합니다.

이름:CephClusterCriticallyFull

Message:스토리지 클러스터는 매우 심각하게 가득 차 있으며 즉각적인 확장이 필요합니다.

설명:스토리지 클러스터 사용률은 85%로 증가했습니다.

심각도: 일반

해결 방법 : 수정

프로시저: 불필요한 데이터를 제거하거나 클러스터를 확장합니다.

이름:CephClusterNearFull

Fix:Storage 클러스터가 가득 차 있습니다. 확장이 필요합니다.

설명:스토리지 클러스터 사용률은 75%를 갖습니다.

심각도: 경고

해결 방법 : 수정

프로시저: 불필요한 데이터를 제거하거나 클러스터를 확장합니다.

이름:NooBaaBucketErrorState

오류:NooBaa Bucket이 오류 상태입니다.

설명:NooBaa 버킷 {{ $labels.bucket_name }}은 6m 이상의 오류 상태입니다.

심각도: 경고

해결 방법: 문제 해결

절차: NooBaa Bucket 오류 상태복구

이름:NooBaaNamespaceResourceErrorState

메시지:NooBaa 네임 스페이스 리소스가 오류 상태입니다.

설명:NooBaa 네임스페이스 리소스 {{ $labels.namespace_resource_name }}은 5m 이상의 오류 상태입니다.

심각도: 경고

해결 방법 : 수정

절차: NooBaa Bucket 오류 상태복구

이름:**NooBaaNamespaceBucketErrorState**

오류:**NooBaa Namespace Bucket이 오류 상태입니다.**

설명:**NooBaa 네임 스페이스 버킷 {{ $labels.bucket_name }}은 5m 이상의 오류 상태입니다.**

심각도: 경고

해결 방법 : 수정

절차:NooBaa Bucket 오류 상태복구

Name: **NooBaaBucketExceedingQuotaState**

메시지:**NooBaa Bucket이 내부 할당량 상태에**있습니다.

설명:**NooBaa 버킷 {{ $labels.bucket_name }}은 할당량을 초과합니다 - {{ printf "%0.f" $value }}% 사용된 메시지: NooBaa Bucket Is In Exceeding Quota State**

심각도: 경고

해결 방법 : 수정

절차:NooBaa Bucket Exceeding Quota 상태조정

이름:**NooBaaBucketLowCapacityState**

메시지:**NooBaa Bucket은 용량의 낮은 상태에**있습니다.

설명:**NooBaa 버킷 {{ $labels.bucket_name }}은 {{ printf "0.f" $value }}%를 사용합니다.**

심각도: 경고

해결 방법 : 수정

절차:NooBaa Bucket 용량 또는 할당량 상태조정

이름:**NooBaaBucketNoCapacityState**

메시지:**NooBaa Bucket은 용량 상태가 아닙니다.**

설명:**NooBaa 버킷 {{ $labels.bucket_name }}은 모든 용량을 사용합니다.**

심각도: 경고

해결 방법 : 수정

절차:NooBaa Bucket 용량 또는 할당량 상태조정

Name: **NooBaaBucketReachingQuotaState**

메시지:NooBaa Bucket이 할당량 상태에있습니다.

설명:NooBaa 버킷 {{ $labels.bucket_name }}은 {{ printf "0.0f" $value }}%를 사용하고 있습니다.

심각도: 경고

해결 방법 : 수정

절차: NooBaa Bucket 용량 또는 할당량 상태조정

이름:NooBaResourceErrorState

오류:NooBaa 리소스가 오류 상태에있습니다.

설명:NooBaa 리소스 {{ $labels.resource_name }}은 6m 이상의 오류 상태입니다.

심각도: 경고

해결 방법: 문제 해결

절차: NooBaa Bucket 오류 상태복구

Name: **NooBaaSystemCapacityWarning100**

메시지:NooBaa System Approached 용량

설명:NooBaa 시스템이 용량에 접근하여 사용량은 100 %입니다.

심각도: 경고

해결 방법 : 수정

절차: NooBaa Bucket 용량 또는 할당량 상태조정

Name: **NooBaaSystemCapacityWarning85**

메시지:NooBaa 시스템은 용량과 관련이 있습니다.

설명:NooBaa 시스템이 용량에 접근하고 있으며 사용량은 85% 이상입니다.

심각도: 경고

해결 방법 : 수정

절차: NooBaa Bucket 용량 또는 할당량 상태조정

Name: **NooBaaSystemCapacityWarning95**

메시지:NooBaa 시스템은 용량과 관련이 있습니다.

설명:NooBaa 시스템이 용량에 접근하고 있으며 사용량은 95% 이상입니다.

심각도: 경고

해결 방법 : 수정

절차:NooBaa Bucket 용량 또는 할당량 상태조정

이름:CephMdsMissingReplicas

Message:스토리지 메타데이터 서비스에 대한 복제본 부족.

설명: 'Minimum required replicas for storage metadata service not available.

스토리지 클러스터 작동에 영향을 미칠 수 있습니다.

심각도: 경고

해결 방법 :Red Hat 지원 문의

절차:

1. 경고 및 Operator 상태를 확인합니다.
2. 문제를 식별할 수 없는 경우 Red Hat 지원에 문의하십시오.

이름:CephMgrIsAbsent

Message:스토리지 메트릭 수집기 서비스를 더 이상 사용할 수 없습니다.

Description:Ceph Manager가 Prometheus 대상 검색에서 사라졌습니다.

심각도: 심각

해결 방법 :Red Hat 지원 문의

절차:

1. 사용자 인터페이스를 검사하고 로그인하여 업데이트가 진행 중인지 확인합니다.
   - 업데이트 진행 중인 경우 이 경고는 일시적인 경고입니다.
   - 업데이트가 진행 중이 아니면 업그레이드 프로세스를 다시 시작합니다.
2. 업그레이드가 완료되면 경고 및 Operator 상태를 확인합니다.
3. 문제가 지속되거나 식별할 수 없는 경우 Red Hat 지원에 문의하십시오.

이름:CephNodeDown

Message:Storage 노드 {{ $labels.node }}가 다운되었습니다.

설명:Storage 노드 {{ $labels.node }}가 다운되었습니다. 즉시 노드를 확인하십시오.

심각도: 심각

해결 방법 : Red Hat 지원 문의

절차:

1. 작동이 중지된 노드와 해당 원인을 확인합니다.
2. 노드를 복구하려면 적절한 작업을 수행합니다. 노드를 복구할 수 없는 경우:
   - Red Hat OpenShift Data Foundation용 스토리지 노드배치를 참조하십시오.
   - Red Hat 지원에 문의.

이름:CephClusterErrorState

Message:Storage 클러스터가 오류 상태입니다.

설명:스토리지 클러스터는 10m 이상에 대한 오류 상태입니다.

심각도: 심각

해결 방법 : Red Hat 지원 문의

절차:

1. 경고 및 Operator 상태를 확인합니다.
2. 문제를 식별할 수 없는 경우 must-gather를 사용하여 로그 파일 및 진단 정보를 다운로드합니다.
3. must-gather의 출력 내용이 첨부되어 Red Hat 지원을 통해 지원티켓을 엽니다.

이름:CephClusterWarningState

Message:스토리지 클러스터가 degraded 상태입니다.

설명:스토리지 클러스터는 10m 이상에 대한 경고 상태입니다.

심각도: 경고

해결 방법 : Red Hat 지원 문의

절차:

1. 경고 및 Operator 상태를 확인합니다.
2. 문제를 식별할 수 없는 경우 must-gather를 사용하여 로그 파일 및 진단 정보를 다운로드합니다.
3. must-gather의 출력 내용이 첨부되어 Red Hat 지원을 통해 지원티켓을 엽니다.

Name: **CephDataRecoveryTakingTooLong**

메시지:데이터 복구 속도가 느립니다.

설명:데이터 복구가 너무 오래 활성화되었습니다.

심각도: 경고

해결 방법 :Red Hat 지원 문의

이름:CephOSDDiskNotResponding

메시지:디스크가 응답하지 않음

설명:디스크 장치 {{ $labels.device }}에 응답하지 않는 호스트 {{ $labels.host }}.

심각도: 심각

해결 방법 :Red Hat 지원 문의

Name:**CephOSDDiskUnavailable**

Message:**Disk not accessible**

설명:디스크 장치 {{ $labels.device }}는 호스트 {{ $labels.host }}에서 액세스할 수 없습니다.

심각도: 심각

해결 방법 :Red Hat 지원 문의

이름:CephPGRepairTakingTooLong

Message: 발견된 **자동 복구 문제**

설명:자체 복구 작업이 너무 오래 걸립니다.

심각도: 경고

해결 방법 :Red Hat 지원 문의

Name: **CephMonHighNumberOfLeaderChanges**

Message:**Storage Cluster에서 최근 여러 리더의 변경 사항을 확인했습니다.**

설명:'Ceph 모니터 "{{ $labels.job }}": 인스턴스 {{ $labels.instance }}은 최근 분당 {{ $value printf "%.2f" }} 리더가 변경되었습니다.

심각도: 경고

해결 방법 :Red Hat 지원 문의

Name: **CephMonQuorumAtRisk**

Message : 위험한 스토리지 쿼럼

설명:스토리지 클러스터 쿼럼이 낮습니다.

심각도: 심각

해결 방법 :Red Hat 지원 문의

Name:**ClusterObjectStoreState**

Message:**Cluster Object Store가 비정상 상태입니다. Ceph** 클러스터 상태를 확인하십시오.

설명:Cluster 오브젝트 저장소는 **15초** 이상 비정상 상태입니다. **Ceph** 클러스터 상태를 확인하십시오.

심각도: 심각

해결 방법 :Red Hat 지원 문의

절차:

- **CephObjectStore** CR 인스턴스를 확인합니다.
- Red Hat 지원에 문의.

이름:CephOSDFlapping

Message:**Storage 데몬 osd.x가 지난 5분 동안 5번 다시 시작되었습니다. Pod 이벤트 또는 Ceph 상태를 확인하여 원인** 을 확인하십시오.

설명:스토리지 OSD는 5분 내에 5회 이상 재시작 합니다.

심각도: 심각

해결 방법 :Red Hat 지원 문의

Name: **OdfPoolMirroringImageHealth**

메시지:마이버링 이미지 (PV) 풀의 <pool-name>은 1m 이상의 Warning 상태에 있습니다. 미러링이 예상대로 작동하지 않을 수 있습니다.

설명: 하나 또는 몇 가지 응용 프로그램에 대한 재해 복구가 실패합니다.

심각도: 경고

해결 방법 :Red Hat 지원 문의

Name: **OdfMirrorDaemonStatus**

Message:**Mirror daemon이 비정상입니다**.

설명: 전체 클러스터에 대해 재해 복구가 실패합니다. 미러 데몬은 1m 이상 비정상 상태에 있습니다. 이 클러스터의 미러링이 예상대로 작동하지 않습니다.

심각도: 심각

해결 방법 :Red Hat 지원 문의

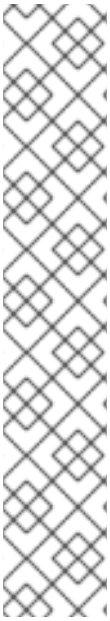
6.2. 클러스터 상태 문제 해결

Red Hat Ceph Storage 클러스터에서 OpenShift Data Foundation 사용자 인터페이스에서 표시되는 유한한 상태 메시지 집합이 있습니다. 이들은 고유 식별자를 가진 상태 검사로 정의됩니다. 식별자는 도구를 사용하여 상태 점검을 이해할 수 있도록 하고 의미를 반영하는 방식으로 제공할 수 있도록 설계된 의사-human-readable 문자열입니다. 자세한 내용과 문제 해결을 위해 아래의 상태 코드를 클릭합니다.

상태 코드	설명
MON_DISK_LOW	하나 이상의 Ceph 모니터가 디스크 공간보다 낮습니다.

6.2.1. MON_DISK_LOW

이 경고는 monitor 데이터베이스를 백분율로 저장하는 파일 시스템의 사용 가능한 공간이 **mon_data_avail_warn** (기본값: 80%)으로 떨어지면 트리거됩니다. 이는 시스템의 다른 프로세스 또는 사용자가 모니터에 사용되는 것과 동일한 파일 시스템을 채우고 있음을 나타낼 수 있습니다. 또한 모니터의 데이터베이스가 많음을 나타낼 수도 있습니다.



참고

파일 시스템의 경로는 mons 배포에 따라 다릅니다. mon이 **storagecluster.yaml** 에 배포된 경로를 찾을 수 있습니다.

경로 예:

- PVC 경로를 통해 배포된 Mon은 **/var/lib/ceph/mon**
- hostpath를 통해 배치된 mon: **/var/lib/rook/mon**

공간을 지우려면 파일 시스템의 높은 사용량 파일을 보고 삭제할 파일을 선택합니다. 파일을 보려면 다음을 실행합니다.

```
# du -a <path-in-the-mon-node> |sort -n -r |head -n10
```

& **lt;path-in-the-mon-node** >를 mons가 배포된 파일 시스템의 경로로 바꿉니다.

6.3. NOOBAA BUCKET 오류 상태 해결

절차

1. OpenShift 웹 콘솔에서 **스토리지 → 데이터 생성을 클릭합니다**.
2. **개요** 탭의 **상태** 카드에서 **스토리지 시스템**을 클릭한 다음 해당 팝업에서 스토리지 시스템 링크를 클릭합니다.
3. **오브젝트** 탭을 클릭합니다.
4. **세부 정보** 카드의 **시스템 이름** 필드에 있는 링크를 클릭합니다.
5. 왼쪽 창에서 **Buckets** 옵션을 클릭하고 오류 상태의 버킷을 검색합니다. 오류의 버킷이 네임스페이스 버킷인 경우 **Namespace Buckets** 창을 클릭해야 합니다.
6. **Bucket Name** 을 클릭합니다. 버킷에서 오류가 발생했습니다.
7. 버킷의 특정 오류에 따라 다음 중 하나 또는 둘 다를 수행합니다.
   a. 공간 관련 오류:
      i. 왼쪽 창에서 **리소스** 옵션을 클릭합니다.
      ii. 오류 상태의 리소스를 클릭합니다.
      iii. 더 많은 에이전트를 추가하여 리소스를 확장합니다.
   b. 리소스 상태 오류의 경우:
      i. 왼쪽 창에서 **리소스** 옵션을 클릭합니다.
      ii. 오류 상태의 리소스를 클릭합니다.
      iii. 연결 오류는 백업 서비스를 사용할 수 없으며 복원해야 함을 의미합니다.
      iv. 액세스/권한 오류의 경우 연결의 **액세스 키 및 시크릿 키** 를 업데이트합니다.

6.4. NOOBAA BUCKET EXCEEDING QUOTA STATE 해결

A NooBaa Bucket을 해결하기 위해 할당량 상태 오류 발생 시다음 중 하나를 수행합니다.

- 버킷의 일부 데이터를 정리합니다.
- 다음 단계를 수행하여 버킷 할당량을 늘립니다.
  1. OpenShift 웹 콘솔에서 **스토리지 → 데이터 생성을 클릭합니다**.
  2. **개요** 탭의 **상태** 카드에서 **스토리지 시스템**을 클릭한 다음 해당 팝업에서 스토리지 시스템 링크를 클릭합니다.
  3. **오브젝트** 탭을 클릭합니다.
  4. **세부 정보** 카드의 **시스템 이름** 필드에 있는 링크를 클릭합니다.
  5. 왼쪽 창에서 **Buckets** 옵션을 클릭하고 오류 상태의 버킷을 검색합니다.
  6. **Bucket Name** 을 클릭합니다. 버킷에서 오류가 발생했습니다.
  7. **Bucket Policies → Edit Quota** 을 클릭하고 할당량을 늘립니다.

6.5. NOOBAA BUCKET 용량 또는 할당량 상태 해결

절차

1. OpenShift 웹 콘솔에서 **스토리지 → 데이터 생성을 클릭합니다**.
2. **개요** 탭의 **상태** 카드에서 **스토리지 시스템**을 클릭한 다음 해당 팝업에서 스토리지 시스템 링크를 클릭합니다.
3. **오브젝트** 탭을 클릭합니다.
4. **세부 정보** 카드의 **시스템 이름** 필드에 있는 링크를 클릭합니다.
5. 왼쪽 창에서 **Resources** 옵션을 클릭하고 PV 풀 리소스를 검색합니다.
6. 용량 상태가 낮은 PV 풀 리소스의 경우 **리소스 이름을 클릭합니다**.
7. 풀 구성을 편집하고 에이전트 수를 늘립니다.

6.6. POD 복구

첫 번째 노드(say **NODE1**)가 일부 문제로 인해 NotReady 상태가 되면RWO(RWO) 액세스 모드에서 PVC를 사용하는 호스트 Pod를 두 번째 노드(NODE2 ay **NODE2**)로 이동하려고 하지만 다중 연결 오류로 인해 중단되었습니다. 이 경우 다음 단계를 사용하여 MON, OSD 및 애플리케이션 Pod를 복구할 수 있습니다.

절차

1. **NODE1** 의 전원을 NODE1(AWS 또는 vSphere 측에서) 끄고 **NODE1** 이 완전히 꺼졌는지 확인합니다.
2. 다음 명령을 사용하여 **NODE1** 에서 Pod를 강제로 삭제합니다.

```
$ oc delete pod <pod-name> --grace-period=0 --force
```

6.7. EBS 볼륨 분리에서 복구

OSD 디스크가 상주하는 OSD 또는 MON 탄력적 블록 스토리지(EBS) 볼륨이 작업자 Amazon EC2 인스턴스에서 분리되면 볼륨이 하나 또는 2분 내에 자동으로 다시 연결됩니다. 그러나 OSD 포드는 **CrashLoopBackOff** 상태로 전환합니다. 포드를 복구하여 **Running** 상태로 되돌리려면 EC2 인스턴스를 다시 시작해야 합니다.

6.8. ROOK-CEPH-OPERATOR의 디버그 로그 활성화 및 비활성화

rook-ceph-operator의 디버그 로그를 활성화하여 문제 해결에 도움이 되는 오류에 대한 정보를 가져옵니다.

절차

디버그 로그 활성화

1. rook-ceph-operator의 configmap을 편집합니다.

```
$ oc edit configmap rook-ceph-operator-config
```

2. **rook-ceph-operator-config** yaml 파일의 **ROOK_LOG_LEVEL: DEBUG** 매개 변수를 추가하여 rook-ceph-operator의 디버그 로그를 활성화합니다.

```
…
data:
  # The logging level for the operator: INFO | DEBUG
  ROOK_LOG_LEVEL: DEBUG
```

이제 rook-ceph-operator 로그가 디버그 정보로 구성됩니다.

디버그 로그 비활성화

1. rook-ceph-operator의 configmap을 편집합니다.

```
$ oc edit configmap rook-ceph-operator-config
```

2. **rook-ceph-operator-config** yaml 파일에 **ROOK_LOG_LEVEL: INFO** 매개변수를 추가하여 rook-ceph-operator에 대한 디버그 로그를 비활성화합니다.

```
…
data:
  # The logging level for the operator: INFO | DEBUG
  ROOK_LOG_LEVEL: INFO
```

7장. LOCAL STORAGE OPERATOR 배포 확인

Local Storage Operator가 포함된 Red Hat OpenShift Data Foundation 클러스터는 로컬 스토리지 장치를 사용하여 배포됩니다. OpenShift Data Foundation이 있는 기존 클러스터가 로컬 스토리지 장치를 사용하여 배포되었는지 확인하려면 다음 절차를 사용하십시오.

사전 요구 사항

- OpenShift Data Foundation은 **openshift-storage** 네임스페이스에 설치되고 실행됩니다.

절차

OpenShift Data Foundation 클러스터의 PVC(영구 볼륨 클레임)와 연결된 스토리지 클래스를 확인하여 클러스터가 로컬 스토리지 장치를 사용하여 배포되었는지 확인할 수 있습니다.

1. 다음 명령을 사용하여 OpenShift Data Foundation 클러스터의 PVC와 연결된 스토리지 클래스를 확인합니다.

```
$ oc get pvc -n openshift-storage
```

2. 출력을 확인합니다. Local Storage Operator가 있는 클러스터의 경우 **ocs-deviceset** 과 연결된 PVC는 스토리지 클래스 **localblock** 을 사용합니다. 출력은 다음과 유사합니다.

```
NAME                      STATUS   VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS              AGE
db-noobaa-db-0            Bound    pvc-d96c747b-2ab5-47e2-b07e-1079623748d8   50Gi
RWO           ocs-storagecluster-ceph-rbd   114s
ocs-deviceset-0-0-lzfrd   Bound    local-pv-7e70c77c                          1769Gi     RWO
localblock                2m10s
ocs-deviceset-1-0-7rggl   Bound    local-pv-b19b3d48                          1769Gi     RWO
localblock                2m10s
ocs-deviceset-2-0-znhk8   Bound    local-pv-e9f22cdc                          1769Gi     RWO
localblock                2m10s
```

추가 리소스

- VMware에서 로컬 스토리지 장치를 사용하여 OpenShift Data Foundation 배포
- Deploying OpenShift Data Foundation using local storage devices on Red Hat Virtualization
- 베어 메탈에서 로컬 스토리지 장치를 사용하여 OpenShift Data Foundation 배포
- Deploying OpenShift Data Foundation using local storage devices on IBM Power

8장. 실패한 CEPH OBJECT STORAGE 장치 제거

실패하거나 원하지 않는 Ceph OSD(Object Storage Devices)는 스토리지 인프라의 성능에 영향을 미칩니다. 따라서 스토리지 클러스터의 안정성과 복원력을 개선하려면 실패하거나 원하지 않는 Ceph OSD를 제거해야 합니다.

Ceph OSD가 실패하거나 원하지 않는 경우 다음을 제거합니다.

1. Ceph 상태를 확인합니다.
   자세한 내용은 Ceph 클러스터의 정상 확인에서 참조하십시오 .
2. OSD의 프로비저닝에 따라 실패하거나 원하지 않는 Ceph OSD를 제거합니다.
   다음 내용을 참조하십시오.
   - 동적으로 프로비저닝된 Red Hat OpenShift Data Foundation에서 실패하거나 원하지 않는 Ceph OSD 제거.
   - 로컬 스토리지 장치를 사용하여 프로비저닝된 Ceph OSD를 제거하지 못했습니다 .

로컬 디스크를 사용하는 경우 이전 OSD를 제거한 후 이러한 디스크를 재사용할 수 있습니다.

8.1. CEPH 클러스터 상태 확인

스토리지 상태는 **블록** 및 **파일** 및 **오브젝트 대시보드에** 표시됩니다.

절차

1. OpenShift 웹 콘솔에서 **스토리지 → 데이터 생성을 클릭합니다**.
2. **개요** 탭의 **상태** 카드에서 **스토리지 시스템**을 클릭한 다음 해당 팝업에서 스토리지 시스템 링크를 클릭합니다.
3. **블록 및 파일** 탭의 **상태** 카드에서 *스토리지 클러스터*에 녹색 체크 표시가 있는지 확인합니다.
4. **세부 정보** 카드에서 클러스터 정보가 표시되는지 확인합니다.

8.2. 동적으로 프로비저닝된 RED HAT OPENSHIFT DATA FOUNDATION에서 실패 또는 원하지 않는 CEPH OSD 제거

프로세스의 단계에 따라 동적으로 프로비저닝된 Red Hat OpenShift Data Foundation에서 실패하거나 원하지 않는 Ceph OSD를 제거합니다.

중요

클러스터 축소는 Red Hat 지원 팀을 통해서만 지원됩니다.

주의

- Ceph 구성 요소가 정상 상태가 아닌 경우 OSD를 제거하면 데이터가 손실될 수 있습니다.
- 두 개 이상의 OSD를 동시에 제거하면 데이터가 손실됩니다.

사전 요구 사항

- Ceph가 정상인지 확인합니다. 자세한 내용은 Ceph 클러스터의 정상 확인을 참조하십시오 .
- 경고가 실행되지 않거나 다시 빌드 프로세스가 진행 중인지 확인합니다.

절차

1. OSD 배포를 축소합니다.

```
# oc scale deployment rook-ceph-osd-<osd-id> --replicas=0
```

2. Ceph OSD를 제거할 **osd-prepare** Pod를 가져옵니다.

```
# oc get deployment rook-ceph-osd-<osd-id> -oyaml | grep ceph.rook.io/pvc
```

3. **osd-prepare** Pod를 삭제합니다.

```
# oc delete -n openshift-storage pod rook-ceph-osd-prepare-<pvc-from-above-command>-<pod-suffix>
```

4. 클러스터에서 실패한 OSD를 제거합니다.

```
# failed_osd_id=<osd-id>

# oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=$<failed_osd_id> | oc create -f -
```

여기서 **FAILED_OSD_ID** 는 **rook-ceph-osd** 접두사 직후 포드 이름의 정수입니다.

5. 로그를 확인하여 OSD가 성공적으로 제거되었는지 확인합니다.

```
# oc logs -n openshift-storage ocs-osd-removal-$<failed_osd_id>-<pod-suffix>
```

6. 선택 사항: OpenShift Container Platform의 **ocs- osd-removal-job Pod에서 삭제하는 데 cephosd:osd.0이 좋지 않아** 오류가 발생하는 경우 **cephosd:osd.0 오류 문제 해결을 참조하십시오. 실패하거나 원하지 않는 Ceph OSD**를 제거하는 동안 제거되지 않음을 참조하십시오.

7. OSD 배포를 삭제합니다.

```
# oc delete deployment rook-ceph-osd-<osd-id>
```

검증 단계

- OSD가 성공적으로 삭제되었는지 확인하려면 다음을 실행합니다.

```
# oc get pod -n openshift-storage ocs-osd-removal-$<failed_osd_id>-<pod-suffix>
```

이 명령은 상태를 **Completed** 로 반환해야 합니다.

8.3. 로컬 스토리지 장치를 사용하여 프로비저닝된 CEPH OSD 또는 실패한 CEPH OSD 제거

절차의 단계에 따라 로컬 스토리지 장치를 사용하여 프로비저닝된 실패 또는 원하지 않는 Ceph를 제거할 수 있습니다.

중요

클러스터 축소는 Red Hat 지원 팀을 통해서만 지원됩니다.

주의

- Ceph 구성 요소가 정상 상태가 아닌 경우 OSD를 제거하면 데이터가 손실될 수 있습니다.
- 두 개 이상의 OSD를 동시에 제거하면 데이터가 손실됩니다.

사전 요구 사항

- Ceph가 정상인지 확인합니다. 자세한 내용은 Ceph 클러스터의 정상 확인을 참조하십시오 .
- 경고가 실행되지 않거나 다시 빌드 프로세스가 진행 중인지 확인합니다.

절차

1. 강제로 OSD 배포의 복제본을 0으로 확장하여 OSD를 축소합니다. 실패로 OSD가 이미 다운된 경우 이 단계를 건너뛸 수 있습니다.

```
# oc scale deployment rook-ceph-osd-<osd-id> --replicas=0
```

2. 클러스터에서 실패한 OSD를 제거합니다.

```
# failed_osd_id=<osd_id>

# oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=$<failed_osd_id> |
oc create -f -
```

여기서 **FAILED_OSD_ID** 는 **rook-ceph-osd** 접두사 직후 포드 이름의 정수입니다.

3. 로그를 확인하여 OSD가 성공적으로 제거되었는지 확인합니다.

```
# oc logs -n openshift-storage ocs-osd-removal-$<failed_osd_id>-<pod-suffix>
```

4. 선택 사항: OpenShift Container Platform의 **ocs- osd-removal-job Pod에서 삭제하는 데 cephosd:osd.0이 좋지 않아** 오류가 발생하는 경우 **cephosd:osd.0 오류 문제 해결을 참조하십시오. 실패하거나 원하지 않는 Ceph OSD**를 제거하는 동안 제거되지 않음을 참조하십시오.

5. 실패한 OSD와 관련된 PVC(영구 볼륨 클레임) 리소스를 삭제합니다.

   a. 실패한 OSD와 연결된 **PVC** 를 가져옵니다.

   ```
   # oc get -n openshift-storage -o yaml deployment rook-ceph-osd-<osd-id> | grep ceph.rook.io/pvc
   ```

   b. PVC와 연결된 PV(**영구 볼륨**)를 가져옵니다.

   ```
   # oc get -n openshift-storage pvc <pvc-name>
   ```

   c. 실패한 장치 이름을 가져옵니다.

   ```
   # oc get pv <pv-name-from-above-command> -oyaml | grep path
   ```

   d. 실패한 OSD와 **관련된 준비 Pod** 를 가져옵니다.

   ```
   # oc describe -n openshift-storage pvc ocs-deviceset-0-0-nvs68 | grep Mounted
   ```

   e. 연결된 PVC를 제거하기 전에 **osd-prepare Pod** 를 삭제합니다.

   ```
   # oc delete -n openshift-storage pod <osd-prepare-pod-from-above-command>
   ```

   f. 실패한 OSD와 연결된 **PVC** 를 삭제합니다.

   ```
   # oc delete -n openshift-storage pvc <pvc-name-from-step-a>
   ```

6. **LocalVolume 사용자 정의 리소스** (CR)에서 실패한 장치 항목을 제거합니다.

   a. 실패한 장치를 사용하여 노드에 로그인합니다.

   ```
   # oc debug node/<node_with_failed_osd>
   ```

   b. 실패한 장치 이름에 대해 /dev/disk/by-id/<id>를 기록합니다.

   ```
   # ls -alh /mnt/local-storage/localblock/
   ```

7. 선택 사항: Local Storage Operator가 OSD 프로비저닝에 사용되는 경우 {osd-id}로 머신에 로그인하고 장치 심볼릭 링크를 제거합니다.

```
# oc debug node/<node_with_failed_osd>
```

   a. 실패한 장치 이름에 대한 OSD 심볼릭 링크를 가져옵니다.

   ```
   # ls -alh /mnt/local-storage/localblock
   ```

b. 심볼릭 링크를 제거합니다.

```
# rm /mnt/local-storage/localblock/<failed-device-name>
```

8. OSD와 연결된 PV를 삭제합니다.

```
# oc delete pv <pv-name>
```

검증 단계

- OSD가 성공적으로 삭제되었는지 확인하려면 다음을 실행합니다.

```
#oc get pod -n openshift-storage ocs-osd-removal-$<failed_osd_id>-<pod-suffix>
```

이 명령은 상태를 **Completed** 로 반환해야 합니다.

8.4. 실패하거나 원하지 않는 CEPH OSD를 제거하는 동안CEPHOSD:OSD.0 오류 문제 해결

OpenShift Container Platform의 **ocs- osd-removal-job Pod에서 제거하기 위해 cephosd:osd.0이 좋지 않아** 오류가 발생하면 **FORCE_OSD_REMOVAL** 옵션으로 OSD 제거 작업을 실행하여 OSD를 삭제한 상태로 이동합니다.

```
# oc process -n openshift-storage ocs-osd-removal -p FORCE_OSD_REMOVAL=true -p
FAILED_OSD_IDS=$<failed_osd_id> | oc create -f -
```

참고

FORCE_OSD_REMOVAL 옵션을 모든 PG가 활성 상태인 경우에만 사용해야 합니다. 그렇지 않은 경우 PG는 백 채우기를 완료하거나 활성 상태인지 확인해야 합니다.

9장. 설치 제거 중 나머지 리소스 문제 해결 및 삭제

때때로 Operator에서 관리하는 사용자 정의 리소스 중 일부는 필요한 모든 정리 작업을 수행했지만 종료자가 완료될 때까지 "Terminating" 상태가 될 수 있습니다. 이러한 경우 이러한 리소스를 강제로 제거해야 합니다. 이렇게 하지 않으면 모든 제거 단계를 수행한 후에도 리소스가 "Terminating" 상태로 유지됩니다.

1. openshift-storage 네임스페이스가 삭제 시 Terminating 상태인지 확인합니다.

```
$ oc get project -n <namespace>
```

출력:

```
NAME              DISPLAY NAME   STATUS
openshift-storage                Terminating
```

2. 명령 출력의 **STATUS** 섹션에서 **NamespaceFinalizersRemaining** 및 **NamespaceContentRemaining** 메시지를 확인하고 나열된 각 리소스에 대해 다음 단계를 수행합니다.

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

출력 예:

```
status:
  conditions:
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: All resources successfully discovered
    reason: ResourcesDiscovered
    status: "False"
    type: NamespaceDeletionDiscoveryFailure
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: All legacy kube types successfully parsed
    reason: ParsedGroupVersions
    status: "False"
    type: NamespaceDeletionGroupVersionParsingFailure
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: All content successfully deleted, may be waiting on finalization
    reason: ContentDeleted
    status: "False"
    type: NamespaceDeletionContentFailure
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: 'Some resources are remaining: cephobjectstoreusers.ceph.rook.io has
      1 resource instances'
    reason: SomeResourcesRemain
    status: "True"
    type: NamespaceContentRemaining
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: 'Some content in the namespace has finalizers remaining:
cephobjectstoreuser.ceph.rook.io
      in 1 resource instances'
    reason: SomeFinalizersRemain
    status: "True"
    type: NamespaceFinalizersRemaining
```

3. 이전 단계에 나열된 나머지 리소스를 모두 삭제합니다.
   삭제할 각 리소스에 대해 다음을 수행합니다.

   a. 제거해야 하는 리소스의 오브젝트 유형을 가져옵니다. 위의 출력에서 메시지를 확인합니다.
   예:

   메시지: 네임스페이스의 일부 콘텐츠에는 종료자가 남아 있습니다.
   cephobjectstoreuser.ceph.rook.io

   여기에서 **cephobjectstoreuser.ceph.rook.io** 는 오브젝트 종류입니다.

   b. 오브젝트 유형에 해당하는 오브젝트 이름을 가져옵니다.

   ```
   $ oc get  <Object-kind> -n  <project-name>
   ```

   예:

   ```
   $ oc get cephobjectstoreusers.ceph.rook.io -n openshift-storage
   ```

   출력 예:

   ```
   NAME                          AGE
   noobaa-ceph-objectstore-user   26h
   ```

   c. 리소스를 패치합니다.

   ```
   $ oc patch -n <project-name> <object-kind>/<object-name> --type=merge -p
   '{"metadata": {"finalizers":null}}'
   ```

   예제:

   ```
   $ oc patch -n openshift-storage cephobjectstoreusers.ceph.rook.io/noobaa-ceph-
   objectstore-user \
   --type=merge -p '{"metadata": {"finalizers":null}}'
   ```

   출력:

   ```
   cephobjectstoreuser.ceph.rook.io/noobaa-ceph-objectstore-user patched
   ```

4. openshift-storage 프로젝트가 삭제되었는지 확인합니다.

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

   출력:

   ```
   Error from server (NotFound): namespaces "openshift-storage" not found
   ```

   문제가 지속되면 Red Hat 지원팀 에 문의하십시오.

10장. 외부 모드에서 CEPHFS PVC 생성 문제 해결

4.1.1보다 낮은 버전에서 Red Hat Ceph Storage 클러스터를 최신 릴리스로 업데이트하고 새로 배포된 클러스터가 아닌 경우 외부 모드에서 CephFS PVC 생성을 활성화하려면 Red Hat Ceph Storage 클러스터에서 CephFS 풀의 애플리케이션 유형을 수동으로 설정해야 합니다.

1. CephFS pvc가 **Pending** 상태로 설정되어 있는지 확인합니다.

```
# oc get pvc -n <namespace>
```

출력 예:

```
NAME                    STATUS    VOLUME
CAPACITY  ACCESS MODES    STORAGECLASS                  AGE
ngx-fs-pxknkcix20-pod     Pending
                    ocs-external-storagecluster-cephfs  28h
[...]
```

2. 해당 pvc에 대한 이벤트를 보려면 **describe** 출력을 확인합니다.
예상되는 오류 메시지는 **cephfs_metadata/csi.volumes.default/csi.volume.pvc-xxxxxxxx-xxxx-xxxx-xxxxxxxx: (1) Operation not allowed**입니다.

```
# oc describe pvc ngx-fs-pxknkcix20-pod -n nginx-file
```

출력 예:

```
Name:          ngx-fs-pxknkcix20-pod
Namespace:     nginx-file
StorageClass:  ocs-external-storagecluster-cephfs
Status:        Pending
Volume:
Labels:        <none>
Annotations:   volume.beta.kubernetes.io/storage-provisioner: openshift-
storage.cephfs.csi.ceph.com
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:    Filesystem
Mounted By:    ngx-fs-oyoe047v2bn2ka42jfgg-pod-hqhzf
Events:
 Type     Reason              Age                  From
Message
 ----     ------              ----                 ----
-------
 Warning  ProvisioningFailed  107m (x245 over 22h)  openshift-
storage.cephfs.csi.ceph.com_csi-cephfsplugin-provisioner-5f8b66cc96-hvcqp_6b7044af-
c904-4795-9ce5-bf0cf63cc4a4
 (combined from similar events): failed to provision volume with StorageClass "ocs-external-
storagecluster-cephfs": rpc error: code = Internal desc = error (an error (exit status 1)
occurred while
 running rados args: [-m 192.168.13.212:6789,192.168.13.211:6789,192.168.13.213:6789 --
id csi-cephfs-provisioner --keyfile=stripped -c /etc/ceph/ceph.conf -p cephfs_metadata
getomapval
 csi.volumes.default csi.volume.pvc-1ac0c6e6-9428-445d-bbd6-1284d54ddb47 /tmp/omap-
```

```
get-186436239 --namespace=csi]) occurred, command output streams is ( error getting
omap value
 cephfs_metadata/csi.volumes.default/csi.volume.pvc-1ac0c6e6-9428-445d-bbd6-
1284d54ddb47: (1) Operation not permitted)
```

3. < **cephfs 메타데이터 풀 name>(here cephfs_metadata**) 및 < **cephfs 데이터 풀 name** >(여기re **cephfs_data**)의 설정을 확인합니다. 명령을 실행하려면 Red Hat Ceph Storage 클라이언트 노드에 사전 구성된 **jq** 가 필요합니다.

```
# ceph osd pool ls detail --format=json | jq '.[] | select(.pool_name| startswith("cephfs")) |
.pool_name, .application_metadata' "cephfs_data"
{
  "cephfs": {}
}
"cephfs_metadata"
{
   "cephfs": {}
}
```

4. CephFS 풀의 애플리케이션 유형을 설정합니다.

   - Red Hat Ceph Storage 클라이언트 노드에서 다음 명령을 실행합니다.

     ```
     # ceph osd pool application set <cephfs metadata pool name> cephfs metadata cephfs
     ```

     ```
     # ceph osd pool application set <cephfs data pool name> cephfs data cephfs
     ```

5. 설정이 적용되었는지 확인합니다.

```
 # ceph osd pool ls detail --format=json | jq '.[] | select(.pool_name| startswith("cephfs")) |
.pool_name, .application_metadata' "cephfs_data"
 {
   "cephfs": {
     "data": "cephfs"
  }
 }
 "cephfs_metadata"
 {
   "cephfs": {
     "metadata": "cephfs"
   }
 }
```

6. CephFS PVC 상태를 다시 확인합니다. 이제 PVC가 **Bound** 상태에 있어야 합니다.

```
# oc get pvc -n <namespace>
```

출력 예:

```
NAME                    STATUS    VOLUME
CAPACITY  ACCESS MODES    STORAGECLASS                  AGE
ngx-fs-pxknkcix20-pod     Bound     pvc-1ac0c6e6-9428-445d-bbd6-1284d54ddb47
1Mi       RWO             ocs-external-storagecluster-cephfs  29h
[...]
```

-

11장. OPENSHIFT DATA FOUNDATION에서 모니터 POD 복원

3개가 모두 다운되면 모니터 Pod를 복원하고 OpenShift Data Foundation이 모니터 Pod를 자동으로 복구할 수 없는 경우 모니터 Pod를 복원합니다.

절차

1. **rook-ceph-operator** 및 **ocs** Operator 배포를 축소합니다.

```
# oc scale deployment rook-ceph-operator --replicas=0 -n openshift-storage
```

```
# oc scale deployment ocs-operator --replicas=0 -n openshift-storage
```

2. **openshift-storage** 네임스페이스에 모든 배포 백업을 생성합니다.

```
# mkdir backup
```

```
# cd backup
```

```
# oc project openshift-storage
```

```
# for d in $(oc get deployment|awk -F' ' '{print $1}'|grep -v NAME); do echo $d;oc get deployment $d -o yaml > oc_get_deployment.${d}.yaml; done
```

3. OSD 배포를 패치하여 **livenessProbe** 매개 변수를 제거하고 **sleep** 로 명령 매개 변수를 사용하여 실행합니다.

```
# for i in $(oc get deployment -l app=rook-ceph-osd -oname);do oc patch ${i} -n openshift-storage --type='json' -p '[{"op":"remove", "path":"/spec/template/spec/containers/0/livenessProbe"}]' ; oc patch ${i} -n openshift-storage -p '{"spec": {"template": {"spec": {"containers": [{"name": "osd", "command": ["sleep", "infinity"], "args": []}]}}}}' ; done
```

4. 모든 OSD에서 **monstore** 클러스터 맵을 검색합니다.

   a. **recover_mon.sh** 스크립트를 생성합니다.

```
#!/bin/bash
ms=/tmp/monstore

rm -rf $ms
mkdir $ms

for osd_pod in $(oc get po -l app=rook-ceph-osd -oname -n openshift-storage); do

  echo "Starting with pod: $osd_pod"

  podname=$(echo $osd_pod|sed 's/pod\///g')
  oc exec $osd_pod -- rm -rf $ms
  oc cp $ms $podname:$ms

  rm -rf $ms
```

```
  mkdir $ms

  echo "pod in loop: $osd_pod ; done deleting local dirs"

  oc exec $osd_pod -- ceph-objectstore-tool --type bluestore --data-path
/var/lib/ceph/osd/ceph-$(oc get $osd_pod -ojsonpath='{
.metadata.labels.ceph_daemon_id }') --op update-mon-db --no-mon-config --mon-store-
path $ms
  echo "Done with COT on pod: $osd_pod"

  oc cp $podname:$ms $ms

  echo "Finished pulling COT data from pod: $osd_pod"
done
```

b. **recover_mon.sh** 스크립트를 실행합니다.

```
# chmod +x recover_mon.sh
```

```
# ./recover_mon.sh
```

5. MON 배포를 패치하고 명령 매개 변수를 **sleep** 로 사용하여 실행합니다.

a. MON 배포를 편집합니다.

```
# for i in $(oc get deployment -l app=rook-ceph-mon -oname);do oc patch ${i} -n
openshift-storage -p '{"spec": {"template": {"spec": {"containers": [{"name": "mon",
"command": ["sleep", "infinity"], "args": []}]}}}}'; done
```

b. MON 배포를 패치하여 **initialDelaySeconds** 를 늘립니다.

```
# oc get deployment rook-ceph-mon-a -o yaml | sed "s/initialDelaySeconds:
10/initialDelaySeconds: 2000/g" | oc replace -f -
```

```
# oc get deployment rook-ceph-mon-b -o yaml | sed "s/initialDelaySeconds:
10/initialDelaySeconds: 2000/g" | oc replace -f -
```

```
# oc get deployment rook-ceph-mon-c -o yaml | sed "s/initialDelaySeconds:
10/initialDelaySeconds: 2000/g" | oc replace -f -
```

6. 이전에 검색한 **monstore** 를 **mon-a** Pod에 복사합니다.

```
# oc cp /tmp/monstore/ $(oc get po -l app=rook-ceph-mon,mon=a -oname |sed
's/pod\///g'):/tmp/
```

7. MON 포드로 이동하여 검색된 **monstore** 의 소유권을 변경합니다.

```
# oc rsh $(oc get po -l app=rook-ceph-mon,mon=a -oname)
```

```
# chown -R ceph:ceph /tmp/monstore
```

8. **mon db** 를 다시 빌드하기 전에 인증 키 템플릿 파일을 복사합니다.

```
# oc rsh $(oc get po -l app=rook-ceph-mon,mon=a -oname)
```

```
# cp /etc/ceph/keyring-store/keyring /tmp/keyring
```

```
# cat /tmp/keyring
 [mon.]
   key = AQCleqldWqm5IhAAgZQbEzoShkZV42RiQVffnA==
   caps mon = "allow *"
 [client.admin]
   key = AQCmAKld8J05KxAArOWeRAw63gAwwZO5o75ZNQ==
   auid = 0
   caps mds = "allow *"
   caps mgr = "allow *"
   caps mon = "allow *"
   caps osd = "allow *”
```

9. 각 시크릿에서 다른 모든 Ceph 데몬(MGR, MDS, RGW, Crash, CSI 프로비저너)의 인증 키를 식별합니다.

```
# oc get secret rook-ceph-mds-ocs-storagecluster-cephfilesystem-a-keyring -ojson  | jq
.data.keyring | xargs echo | base64 -d

[mds.ocs-storagecluster-cephfilesystem-a]
key = AQB3r8VgAtr6OhAAVhhXpNKqRTuEVdRoxG4uRA==
caps mon = "allow profile mds"
caps osd = "allow *"
caps mds = "allow"
```

인증 키 파일의 예: **/etc/ceph/ceph.client.admin.keyring**:

```
[mon.]
 key = AQDxTF1hNgLTNxAAi51cCojs01b4I5E6v2H8Uw==
 caps mon = "allow "
[client.admin]
        key = AQDxTF1hpzguOxAA0sS8nN4udoO35OEbt3bqMQ==
        caps mds = "allow " caps mgr = "allow *" caps mon = "allow *" caps osd = "allow *"
[mds.ocs-storagecluster-cephfilesystem-a] key =
AQCKTV1horgjARAA8aF/BDh/4+eG4RCNBCl+aw== caps mds = "allow" caps mon = "allow
profile mds" caps osd = "allow *" [mds.ocs-storagecluster-cephfilesystem-b] key =
AQCKTV1hN4gKLBAA5emIVq3ncV7AMEM1c1RmGA== caps mds = "allow" caps mon =
"allow profile mds" caps osd = "allow *" [client.rgw.ocs.storagecluster.cephobjectstore.a] key
= AQCOkdBixmpiAxAA4X7zjn6SGTI9c1MBflszYA== caps mon = "allow rw" caps osd =
"allow rwx" [mgr.a] key = AQBOTV1hGYOEORAA87471+eIZLZtptfkcHvTRg== caps mds =
"allow *" caps mon = "allow profile mgr" caps osd = "allow *" [client.crash] key =
AQBOTV1htO1aGRAAe2MPYcGdiAT+Oo4CNPSF1g== caps mgr = "allow rw" caps mon =
"allow profile crash" [client.csi-cephfs-node] key =
AQBOTV1hiAtuBBAAaPPBVgh1AqZJlDeHWdoFLw== caps mds = "allow rw" caps mgr =
"allow rw" caps mon = "allow r" caps osd = "allow rw tag cephfs *=" [client.csi-cephfs-
provisioner] key = AQBNTV1hHu6wMBAAzNXZv36aZJuE1iz7S7GfeQ== caps mgr = "allow
rw" caps mon = "allow r" caps osd = "allow rw tag cephfs metadata="
[client.csi-rbd-node]
 key = AQBNTV1h+LnkIRAAWnpIN9bUAmSHOvJ0EJXHRw==
 caps mgr = "allow rw"
 caps mon = "profile rbd"
```

```
 caps osd = "profile rbd"
[client.csi-rbd-provisioner]
 key = AQBNTV1hMNcsExAAvA3gHB2qaY33LOdWCvHG/A==
 caps mgr = "allow rw"
 caps mon = "profile rbd"
 caps osd = "profile rbd"
```

중요

- **client.csi** 관련 인증 키의 경우 이전 인증 키 파일 출력을 참조하고 각 OpenShift Data Foundation 시크릿에서 키를 가져온 후 기본 **대문자** 를 추가합니다.
- OSD 인증 키는 자동으로 복구 후 추가됩니다.

10. **mon-a** pod로 이동하여 **monstore** 에 **monmap** 이 있는지 확인합니다.

    a. **mon-a** Pod로 이동합니다.

    ```
    # oc rsh $(oc get po -l app=rook-ceph-mon,mon=a -oname)
    ```

    b. **monstore** 에 **monmap** 이 있는지 확인합니다.

    ```
    # ceph-monstore-tool /tmp/monstore get monmap -- --out /tmp/monmap
    ```

    ```
    # monmaptool /tmp/monmap --print
    ```

11. 선택 사항: **monmap** 이 없는 경우 새 **monmap** 을 만듭니다.

    ```
    # monmaptool --create --add <mon-a-id> <mon-a-ip> --add <mon-b-id> <mon-b-ip> --add
    <mon-c-id> <mon-c-ip> --enable-all-features --clobber /root/monmap --fsid <fsid>
    ```

    <mon-a-id>

    mon-a Pod의 ID입니다.

    <mon-a-ip>

    는 **mon-a** 포드의 IP 주소입니다.

    <mon-b-id>

    mon-b Pod의 ID입니다.

    <mon-b-ip>

    는 **mon-b** 포드의 IP 주소입니다.

    <mon-c-id>

    mon-c Pod의 ID입니다.

    <mon-c-ip>

    는 **mon-c** Pod의 IP 주소입니다.

    <fsid>

    는 파일 시스템 ID입니다.

12. **monmap** 을 확인합니다.

```
# monmaptool /root/monmap --print
```

13. **monmap** 을 가져옵니다.

중요

이전에 생성된 **인증 키** 파일을 사용합니다.

```
# ceph-monstore-tool /tmp/monstore rebuild -- --keyring /tmp/keyring --monmap
/root/monmap
```

```
# chown -R ceph:ceph /tmp/monstore
```

14. 이전 **store.db** 파일의 백업을 생성합니다.

```
# mv /var/lib/ceph/mon/ceph-a/store.db /var/lib/ceph/mon/ceph-a/store.db.corrupted
```

```
# mv /var/lib/ceph/mon/ceph-b/store.db /var/lib/ceph/mon/ceph-b/store.db.corrupted
```

```
# mv /var/lib/ceph/mon/ceph-c/store.db /var/lib/ceph/mon/ceph-c/store.db.corrupted
```

15. rebuild **store.db** 파일을 **monstore** 디렉터리에 복사합니다.

```
# mv /tmp/monstore/store.db /var/lib/ceph/mon/ceph-a/store.db
```

```
# chown -R ceph:ceph /var/lib/ceph/mon/ceph-a/store.db
```

16. **monstore** 디렉터리를 다시 빌드한 후 local에서 MON 포드의 나머지 부분에 **store.db** 파일을 복사합니다.

```
# oc cp $(oc get po -l app=rook-ceph-mon,mon=a -oname | sed
's/pod\///g'):/var/lib/ceph/mon/ceph-a/store.db /tmp/store.db
```

```
# oc cp /tmp/store.db $(oc get po -l app=rook-ceph-mon,mon=<id> -oname | sed
's/pod\///g'):/var/lib/ceph/mon/ceph-<id>
```

<id>

MON Pod의 ID입니다.

17. MON 포드의 나머지 부분으로 이동하고 복사된 **monstore** 의 소유권을 변경합니다.

```
# oc rsh $(oc get po -l app=rook-ceph-mon,mon=<id> -oname)
```

```
# chown -R ceph:ceph /var/lib/ceph/mon/ceph-<id>/store.db
```

<id>

MON Pod의 ID입니다.

18. 패치된 변경 사항을 되돌립니다.

- MON 배포의 경우:

```
# oc replace --force -f <mon-deployment.yaml>
```

<mon-deployment.yaml>

MON 배포 yaml 파일

- OSD 배포의 경우:

```
# oc replace --force -f <osd-deployment.yaml>
```

<osd-deployment.yaml>

OSD 배포 yaml 파일

- MGR 배포의 경우:

```
# oc replace --force -f <mgr-deployment.yaml>
```

<mgr-deployment.yaml>

MGR 배포 yaml 파일

중요

MON, MGR 및 OSD Pod가 실행 중인지 확인합니다.

19. **rook-ceph-operator** 및 **ocs-operator** 배포를 확장합니다.

```
# oc -n openshift-storage scale deployment ocs-operator --replicas=1
```

검증 단계

1. Ceph 상태를 확인하여 CephFS가 실행 중인지 확인합니다.

```
# ceph -s
```

출력 예:

```
cluster:
  id:     f111402f-84d1-4e06-9fdb-c27607676e55
  health: HEALTH_ERR
          1 filesystem is offline
          1 filesystem is online with fewer MDS than max_mds
          3 daemons have recently crashed

  services:
    mon: 3 daemons, quorum b,c,a (age 15m)
    mgr: a(active, since 14m)
    mds: ocs-storagecluster-cephfilesystem:0
    osd: 3 osds: 3 up (since 15m), 3 in (since 2h)

  data:
```

```
pools:   3 pools, 96 pgs
objects: 500 objects, 1.1 GiB
usage:   5.5 GiB used, 295 GiB / 300 GiB avail
pgs:     96 active+clean
```

2. MCG(Multicloud Object Gateway) 상태를 확인합니다. 활성 상태여야 하며 backingstore 및 버킷 클래스는 **Ready** 상태여야 합니다.

```
noobaa status -n openshift-storage
```

중요

MCG가 활성 상태가 아니며 backstore 및 bucketclass가 **Ready** 상태가 아닌 경우 모든 MCG 관련 Pod를 다시 시작해야 합니다. 자세한 내용은 11.1절. "Multicloud 오브젝트 게이트웨이 복원"의 내용을 참조하십시오.

11.1. MULTICLOUD 오브젝트 게이트웨이 복원

MCG(Multicloud Object Gateway)가 활성 상태가 아니며 backingstore 및 버킷 클래스가 **Ready** 상태가 아닌 경우 모든 MCG 관련 포드를 재시작하고 MCG 상태가 다시 실행되고 있는지 확인해야 합니다.

절차

1. MCG와 관련된 모든 Pod를 다시 시작합니다.

```
# oc delete pods <noobaa-operator> -n openshift-storage
```

```
# oc delete pods <noobaa-core> -n openshift-storage
```

```
# oc delete pods <noobaa-endpoint> -n openshift-storage
```

```
# oc delete pods <noobaa-db> -n openshift-storage
```

<noobaa-operator>

MCG Operator의 이름입니다.

<noobaa-core>

MCG 코어 Pod의 이름입니다.

<noobaa-endpoint>

MCG 엔드 포인트의 이름입니다.

<noobaa-db>

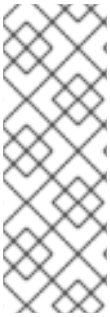
MCG db Pod의 이름입니다.

2. RADOS 개체 게이트웨이(RGW)가 구성된 경우 Pod를 다시 시작합니다.

```
# oc delete pods <rgw-pod> -n openshift-storage
```

<rgw-pod>

RGW Pod의 이름입니다.



참고

OpenShift Container Platform 4.11에서는 복구 후 RBD PVC가 애플리케이션 포드에 마운트되지 않습니다. 따라서 애플리케이션 Pod를 호스팅하는 노드를 재시작해야 합니다. 애플리케이션 Pod를 호스팅하는 노드 이름을 가져오려면 다음 명령을 실행합니다.

```
# oc get pods <application-pod> -n <namespace> -o yaml | grep nodeName
  nodeName: node_name
```

12장. OPENSHIFT DATA FOUNDATION에서 CEPH-MONITOR 쿼럼 복원

경우에 따라 **ceph-mons** 에 쿼럼이 손실될 수 있습니다. **mons** 에서 쿼럼을 다시 구성할 수 없는 경우 쿼럼을 다시 수행하는 수동 절차가 있습니다. 유일한 요구 사항은 적어도 하나의 **몬** 이 건강해야합니다. 다음 단계에서는 쿼럼에서 비정상 **mons** 를 제거하고 단일 **mon** 을 사용하여 쿼럼을 다시 형성한 다음 쿼럼을 원래 크기로 다시 가져올 수 있습니다.

예를 들어, 3개의 **mons** 가 있고 쿼럼이 손실된 경우 쿼럼에서 두 개의 bad **mons** 를 제거하고, 쿼럼에서 유일한 **mon** 임을 알리고, good **mon** 을 다시 시작해야 합니다.

절차

1. **monmap** 을 수정할 때 **mons** 가 실패하지 않도록 **rook-ceph-operator** 를 중지합니다.

```
# oc -n openshift-storage scale deployment rook-ceph-operator --replicas=0
```

2. 새 **monmap** 을 삽입합니다.

> **주의**
>
> **monmap** 을 매우 신중하게 삽입해야합니다. 잘못 실행하면 클러스터가 영구적으로 삭제될 수 있습니다. Ceph **monmap** 은 **mon** 쿼럼을 추적합니다. **monmap** 은 건강한 레몬을 포함하도록 업데이트되었습니다. 이 예에서 정상적인 mon은 **rook-ceph-mon-b** 이지만 비정상 **mons** 는 **rook-ceph-mon-a** 및 **rook-ceph-mon-c** 입니다.

a. 현재 **rook-ceph-mon-b** 배포 백업을 수행합니다.

```
# oc -n openshift-storage get deployment rook-ceph-mon-b -o yaml > rook-ceph-mon-b-deployment.yaml
```

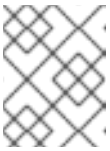
b. YAML 파일을 열고 **mon** 컨테이너의 **명령** 및 **인수를** 복사합니다(다음 예제의 컨테이너 목록 참조). 이를 위해서는 **monmap** 변경이 필요합니다.

```
[...]
  containers:
  - args:
    - --fsid=41a537f2-f282-428e-989f-a9e07be32e47
    - --keyring=/etc/ceph/keyring-store/keyring
    - --log-to-stderr=true
    - --err-to-stderr=true
    - --mon-cluster-log-to-stderr=true
    - '--log-stderr-prefix=debug '
    - --default-log-to-file=false
    - --default-mon-cluster-log-to-file=false
    - --mon-host=$(ROOK_CEPH_MON_HOST)
    - --mon-initial-members=$(ROOK_CEPH_MON_INITIAL_MEMBERS)
    - --id=b
```

```
    - --setuser=ceph
    - --setgroup=ceph
    - --foreground
    - --public-addr=10.100.13.242
    - --setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db
    - --public-bind-addr=$(ROOK_POD_IP)
    command:
    - ceph-mon
[...]
```

c. 복사된 **명령** 과 **args** 필드를 정리하여 다음과 같이 붙여넣을 수 있는 명령을 형성합니다.

```
# ceph-mon \
    --fsid=41a537f2-f282-428e-989f-a9e07be32e47 \
    --keyring=/etc/ceph/keyring-store/keyring \
    --log-to-stderr=true \
    --err-to-stderr=true \
    --mon-cluster-log-to-stderr=true \
    --log-stderr-prefix=debug \
    --default-log-to-file=false \
    --default-mon-cluster-log-to-file=false \
    --mon-host=$ROOK_CEPH_MON_HOST \
    --mon-initial-members=$ROOK_CEPH_MON_INITIAL_MEMBERS \
    --id=b \
    --setuser=ceph \
    --setgroup=ceph \
    --foreground \
    --public-addr=10.100.13.242 \
    --setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db \
    --public-bind-addr=$ROOK_POD_IP
```



참고

ROOK_CEPH_MONH_MON_INITIAL_MEMBERS 및 ROOK_POD_IP 에 전달되는 변수에 대해 **--log-stderr-prefix** 플래그와 괄호를 제거해야 합니다.

d. **rook-ceph-mon-b** Deployment를 패치하여 mon pod를 삭제하지 않고 이 **mon** 의 작동을 중지합니다.

```
# oc -n openshift-storage patch deployment rook-ceph-mon-b  --type='json' -p
'[{"op":"remove", "path":"/spec/template/spec/containers/0/livenessProbe"}]'

# oc -n openshift-storage patch deployment rook-ceph-mon-b -p '{"spec": {"template":
{"spec": {"containers": [{"name": "mon", "command": ["sleep", "infinity"], "args": []}]}}}}'
```

e. **mon-b** Pod에서 다음 단계를 수행합니다.

i. 정상적인 **mon** 의 Pod에 연결하고 다음 명령을 실행합니다.

```
# oc -n openshift-storage exec -it <mon-pod> bash
```

ii. 변수를 설정합니다.

```
# monmap_path=/tmp/monmap
```

iii. 좋은 **mon** 배포에서 ceph mon 명령을 붙여넣고 **--extract- monmap =${ mon map_path}** 플래그를 추가하여 **monmap** 을 파일에 추출합니다.

```
# ceph-mon \
    --fsid=41a537f2-f282-428e-989f-a9e07be32e47 \
    --keyring=/etc/ceph/keyring-store/keyring \
    --log-to-stderr=true \
    --err-to-stderr=true \
    --mon-cluster-log-to-stderr=true \
    --log-stderr-prefix=debug \
    --default-log-to-file=false \
    --default-mon-cluster-log-to-file=false \
    --mon-host=$ROOK_CEPH_MON_HOST \
    --mon-initial-members=$ROOK_CEPH_MON_INITIAL_MEMBERS \
    --id=b \
    --setuser=ceph \
    --setgroup=ceph \
    --foreground \
    --public-addr=10.100.13.242 \
    --setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db \
    --public-bind-addr=$ROOK_POD_IP \
    --extract-monmap=${monmap_path}
```

iv. **monmap** 의 내용을 검토합니다.

```
# monmaptool --print /tmp/monmap
```

v. **mon map** 에서 나쁜 마우스를 제거하십시오.

```
# monmaptool ${monmap_path} --rm <bad_mon>
```

이 예제에서는 **mon0** 및 **mon2** 를 제거합니다.

```
# monmaptool ${monmap_path} --rm a
# monmaptool ${monmap_path} --rm c
```

vi. 다음과 같이 ceph **mon** 명령을 붙여넣고 **--inject- monmap =${ mon map_path}** 플래그를 추가하여 수정된 **monmap** 을 양호한 mon에 삽입합니다.

```
# ceph-mon \
    --fsid=41a537f2-f282-428e-989f-a9e07be32e47 \
    --keyring=/etc/ceph/keyring-store/keyring \
    --log-to-stderr=true \
    --err-to-stderr=true \
    --mon-cluster-log-to-stderr=true \
    --log-stderr-prefix=debug \
    --default-log-to-file=false \
    --default-mon-cluster-log-to-file=false \
    --mon-host=$ROOK_CEPH_MON_HOST \
    --mon-initial-members=$ROOK_CEPH_MON_INITIAL_MEMBERS \
    --id=b \
    --setuser=ceph \
    --setgroup=ceph \
    --foreground \
```

```
        --public-addr=10.100.13.242 \
        --setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db \
        --public-bind-addr=$ROOK_POD_IP \
        --inject-monmap=${monmap_path}
```

vii. 쉘을 종료하여 계속합니다.

3. Rook **configmaps** 를 편집합니다.

a. 운영자가 **mons** 를 추적하는 데 사용하는 **configmap** 을 편집합니다.

```
# oc -n openshift-storage edit configmap rook-ceph-mon-endpoints
```

b. 데이터 요소에서 다음과 같은 세 가지 **mons** (또는 **moncount**에 따라 추가)가 표시되는지 확인합니다.

```
data: a=10.100.35.200:6789;b=10.100.13.242:6789;c=10.100.35.12:6789
```

c. 하나의 좋은 **몬트리트** 로 끝나는 목록의 잘못된 양말을 **삭제합니다**. 예를 들어 다음과 같습니다.

```
data: b=10.100.13.242:6789
```

d. 파일을 저장하고 종료합니다.

e. 이제 **mons** 및 기타 구성 요소에 사용되는 보안을 조정해야 합니다.

i. **변수 good_mon_id 값을 설정합니다.**

예를 들어 다음과 같습니다.

```
# good_mon_id=b
```

ii. **oc patch 명령을 사용하여 rook-ceph-config 시크릿을 패치하고 두 개의 키/값 쌍 mon_host 와 mon_initial_members 를 업데이트할 수 있습니다.**

```
# mon_host=$(oc -n openshift-storage get svc rook-ceph-mon-b -o
jsonpath='{.spec.clusterIP}')

# oc -n openshift-storage patch secret rook-ceph-config -p '{"stringData":
{"mon_host": "[v2:'"${mon_host}"':3300,v1:'"${mon_host}"':6789]",
"mon_initial_members": """${good_mon_id}"""}}'
```

참고

hostNetwork: true 를 사용하는 경우 **mon_host var**을 **mon** 이 고정 **(nodeSelector)**에 고정 된 노드 **IP**로 교체해야 합니다. 이는 해당 **"mode"**에서 생성된 **rook-ceph-mon-*** 서비스가 없기 때문입니다.

4. 레몬을 다시 시작하십시오.

변경 사항을 가져오려면 원래 **ceph- mon** 명령으로 올바른 **mon Pod**를 다시 시작해야 합니다.

a. **oc replace** 명령을 **mon** 배포 **YAML** 파일의 백업에 사용합니다.

```
# oc replace --force -f rook-ceph-mon-b-deployment.yaml
```

참고

option --force 는 배포를 삭제하고 새 배포를 생성합니다.

b. 클러스터 상태를 확인합니다.

상태에 쿼럼에 **1**개의 **mon** 이 표시되어야 합니다. 상태가 정상으로 표시되면 클러스터가 다시 정상이어야 합니다.

5. 더 이상 쿼럼에 없는 두 개의 **mon** 배포를 삭제합니다.

예를 들어 다음과 같습니다.

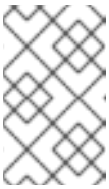
```
# oc delete deploy <rook-ceph-mon-1>
# oc delete deploy <rook-ceph-mon-2>
```

이 예에서 삭제할 배포는 **rook-ceph-mon-a** 및 **rook-ceph-mon-c** 입니다.

6. **Operator**를 다시 시작합니다.

a.

rook Operator를 다시 시작하여 클러스터의 상태 모니터링을 재개합니다.



참고

여러 리소스가 이미 존재하는 오류를 무시하는 것은 안전합니다.

```
# oc -n openshift-storage scale deployment rook-ceph-operator --replicas=1
```

Operator는 **mon** 수에 따라 쿼럼 크기를 다시 늘리기 위해 더 많은 **mons** 를 자동으로 추가합니다.

13장. RED HAT OPENSHIFT DATA FOUNDATION 콘솔 플러그인 활성화

OpenShift Data Foundation Operator를 설치한 후 자동으로 활성화되지 않은 경우 콘솔 플러그인 옵션을 활성화합니다. 콘솔 플러그인은 웹 콘솔에 포함된 사용자 정의 인터페이스를 제공합니다. **GUI(**그래픽 사용자 인터페이스**)** 또는 명령줄 인터페이스에서 콘솔 플러그인 옵션을 활성화할 수 있습니다.

사전 요구 사항

- **OpenShift** 웹 콘솔에 대한 관리자 액세스 권한이 있습니다.
- **OpenShift Data Foundation Operator**는 **openshift-storage** 네임스페이스에 설치 및 실행됩니다.

절차

사용자 인터페이스에서

1. **OpenShift** 웹 콘솔에서 **Operator** → 설치된 **Operator**를 클릭하여 설치된 모든 **Operator**를 확인합니다.
2. 선택한 프로젝트가 **openshift-storage** 인지 확인합니다.
3. **OpenShift Data Foundation Operator**를 클릭합니다.
4. 콘솔 플러그인 옵션을 활성화합니다.
   a. 세부 정보 탭의 콘솔 플러그인 아래에 있는 연필 아이콘을 클릭합니다.
   b. 사용을 선택하고 저장을 클릭합니다.

명령줄 인터페이스에서

- 콘솔 플러그인 옵션을 활성화하려면 다음 명령을 실행합니다.

```
$ oc patch console.operator cluster -n openshift-storage --type json -p '[{"op": "add",
"path": "/spec/plugins", "value": ["odf-console"]}]'
```

-

검증 단계

- 콘솔 플러그인 옵션을 활성화하면 메시지가 있는 팝업이 표시되면 **GUI**에 웹 콘솔 업데이트를 사용할 수 있습니다. 콘솔 변경 사항을 반영하려면 이 팝업 창에서 웹 콘솔 새로 고침을 클릭합니다.
  - 웹 콘솔에서 **Storage** (스토리지)로 이동하여 **Data Foundation** 이 사용 가능한지 확인합니다.

14장. OPENSHIFT DATA FOUNDATION 구성 요소에 대한 리소스 변경

OpenShift Data Foundation을 설치할 때 OpenShift Data Foundation Pod에서 사용할 수 있는 사전 정의된 리소스가 제공됩니다. I/O 로드가 높은 경우 이러한 제한을 늘려야 할 수 있습니다.

- rook-ceph Pod에서 CPU 및 메모리 리소스를 변경하려면 14.1절. “rook-ceph Pod에서 CPU 및 메모리 리소스 변경” 을 참조하십시오.
- MCG(Multicloud Object Gateway)의 리소스를 튜닝하려면 14.2절. “MCG의 리소스 튜닝” 을 참조하십시오.

14.1. ROOK-CEPH POD에서 CPU 및 메모리 리소스 변경

OpenShift Data Foundation을 설치하면 rook-ceph Pod에 대한 사전 정의된 CPU 및 메모리 리소스가 제공됩니다. 요구 사항에 따라 이러한 값을 수동으로 늘릴 수 있습니다.

다음 Pod에서 CPU 및 메모리 리소스를 변경할 수 있습니다.

- mgr
- mds
- rgw

다음 예제에서는 rook-ceph Pod에서 CPU 및 메모리 리소스를 변경하는 방법을 보여줍니다. 이 예에서는 cpu 및 memory 의 기존 MDS Pod 값이 각각 1 및 4Gi 에서 8Gi 로 증가했습니다.

1. 스토리지 클러스터를 편집합니다.

```
# oc edit storagecluster -n openshift-storage <storagecluster_name>
```

<storagecluster_name>

스토리지 클러스터의 이름을 지정합니다.

예를 들어 다음과 같습니다.

```
# oc edit storagecluster -n openshift-storage ocs-storagecluster
```

2. 스토리지 클러스터 CR(사용자 정의 리소스)에 다음 행을 추가합니다.

```
spec:
  resources:
    mds:
      limits:
        cpu: 2
        memory: 8Gi
      requests:
        cpu: 2
        memory: 8Gi
```

3. 변경 사항을 저장하고 편집기를 종료합니다.

4. 또는 **oc patch** 명령을 실행하여 **mds Pod**의 **CPU** 및 메모리 값을 변경합니다.

```
# oc patch -n openshift-storage storagecluster <storagecluster_name>
    --type merge \
    --patch '{"spec": {"resources": {"mds": {"limits": {"cpu": "2","memory": "8Gi"},"requests":
{"cpu": "2","memory": "8Gi"}}}}}'
```

<storagecluster_name>

스토리지 클러스터의 이름을 지정합니다.

예를 들어 다음과 같습니다.

```
# oc patch -n openshift-storage storagecluster ocs-storagecluster \
    --type merge \
    --patch '{"spec": {"resources": {"mds": {"limits": {"cpu": "2","memory": "8Gi"},"requests":
{"cpu": "2","memory": "8Gi"}}}}}'
```

14.2. MCG의 리소스 튜닝

MCG(Multicloud Object Gateway)의 기본 구성은 성능이 저하되고 리소스 사용에 최적화되어 있습니다. **MCG**에 대한 리소스를 조정하는 방법에 대한 자세한 내용은 NooBaa(Multicloud Object Gateway)에

대한 **Red Hat** 지식베이스 솔루션 성능 튜닝 가이드 를 참조하십시오.

15장. 글로벌 POD 네트워킹을 수동으로 활성화하여 OVS-MULTITENANT 플러그인을 사용하여 ODF-CONSOLE 에 액세스

OpenShift Container Platform에서 ovs-multitenant 플러그인을 소프트웨어 정의 네트워킹(SDN)에 사용하는 경우 다른 프로젝트의 Pod는 다른 프로젝트의 포드 및 서비스에서 패킷을 보내거나 받을 수 없습니다. 프로젝트의 Pod 네트워킹이 글로벌 상태가 아니므로 기본적으로 Pod는 네임스페이스 또는 프로젝트 간에 통신할 수 없습니다.

odf-console에 액세스하려면 openshift-console 네임스페이스의 OpenShift 콘솔 포드가 openshift-storage 네임스페이스의 OpenShift Data Foundation odf-console에 연결되어 있어야 합니다. 이는 글로벌 Pod 네트워킹을 수동으로 사용하는 경우에만 가능합니다.

문제

- OpenShift Container Platform에서'ovs-multitenant' 플러그인이 사용되는 경우 다음 메시지와 함께 odf-console 플러그인이 실패합니다.

```
GET request for "odf-console" plugin failed: Get "https://odf-console-service.openshift-storage.svc.cluster.local:9001/locales/en/plugin__odf-console.json": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
```

해결

- OpenShift Data Foundation 프로젝트의 포드 네트워킹을 글로벌로 설정합니다.

```
$ oc adm pod-network make-projects-global openshift-storage
```