



Red Hat OpenShift Container Storage 4.6

OpenShift Container Storage 故障排除

如何排除 OpenShift Container Storage 中的错误和问题

Red Hat OpenShift Container Storage 4.6 OpenShift Container Storage 故障排除

如何排除 OpenShift Container Storage 中的错误和问题

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.

法律通告

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

摘要

参阅此文档来了解与对 Red Hat OpenShift Container Storage 进行故障排除的说明。

目录

第 1 章 概述	3
第 2 章 使用 MUST-GATHER 下载日志文件和诊断信息	4
第 3 章 故障排除所需的常见日志	6
第 4 章 为 OPENSIFT CONTAINER STORAGE 部署后覆盖集群范围默认节点选择器	9
第 5 章 对 OPENSIFT CONTAINER STORAGE 中的警报和错误进行故障排除	10
5.1. 解决警报和错误	10
5.2. 解决 NOOBAA BUCKET 错误状态	17
5.3. 解决 NOOBAA BUCKET EXCEEDING QUOTA STATE 问题	17
5.4. 解决 NOOBAA BUCKET CAPACITY 或 QUOTA STATE 问题	17
5.5. 恢复 POD	18
5.6. 从 EBS 卷分离中恢复	18
第 6 章 检查 LOCAL STORAGE OPERATOR 部署	19
第 7 章 卸载过程中的故障排除和删除剩余的资源	20
第 8 章 对外部模式的 CEPHFS PVC 创建进行故障排除	22
第 9 章 在 OPENSIFT CONTAINER STORAGE 中恢复监控 POD	24

第 1 章 概述

编写 OpenShift Container Storage 故障排除后，可帮助管理员了解如何对 Red Hat OpenShift Container Storage 集群进行故障排除和修复。

大多数故障排除任务都侧重于修复或临时解决方案。本文档根据管理员可能遇到的错误分为若干章节：

- [第 2 章 使用 `must-gather` 下载日志文件和诊断信息](#) 显示如何在 OpenShift Container Storage 中使用 `must-gather` 实用程序。
- [第 3 章 故障排除所需的常见日志](#) 显示如何获取 OpenShift Container Storage 所需的日志文件。
- [第 5 章 对 OpenShift Container Storage 中的警报和错误进行故障排除](#) 如何识别遇到的错误并执行所需的操作。

第 2 章 使用 MUST-GATHER 下载日志文件和诊断信息

如果 Red Hat OpenShift Container Storage 无法自动解决问题，请使用 `must-gather` 工具来收集日志文件和诊断信息，以便或红帽支持可以查看问题，并确定解决方案。



重要

当 OpenShift Container Storage 以外部模式部署时，`must-gather` 只从 Red Hat OpenShift Container Storage 集群收集日志，且不会从外部 Red Hat Ceph Storage 集群收集调试数据和日志。要从外部 Red Hat Ceph Storage 集群收集调试日志，请参阅 Red Hat Ceph Storage [故障排除指南](#)并联系您的 Red Hat Ceph Storage 管理员。

步骤

- 从连接到 OpenShift Container Storage 集群的客户端运行 `must-gather` 命令：

```
$ oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6 --dest-dir=<directory-name>
```

这会在指定目录中收集以下信息：

- 所有 OpenShift Container Storage 集群相关的自定义资源(CR)及其命名空间。
- 与所有 OpenShift Container Storage 相关的 pod 的 Pod 日志。
- 某些标准 Ceph 命令的输出，如状态、集群运行状况等。

命令变体

- 如果一个或多个 master 节点没有处于 **Ready** 状态，请使用 `--node-name` 指定一个状态为 **Ready** 的 master 节点，以便可以安全地调度 `must-gather` pod。

```
$ oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6 --dest-dir=<directory-name> --node-name=<node-name>
```

- 如果要从特定时间收集信息：
 - 要为收集的日志指定相对时间段（例如在 5 秒内或在 2 天内），添加 `/usr/bin/gather since=<duration>`：

```
$ oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6 --dest-dir=<directory-name> /usr/bin/gather since=<duration>
```

- 要指定在以后的一个特定时间收集日志，添加 `/usr/bin/gather since-time=<rfc3339-timestamp>`：

```
$ oc adm must-gather --image=registry.redhat.io/ocs4/ocs-must-gather-rhel8:v4.6 --dest-dir=<directory-name> /usr/bin/gather since-time=<rfc3339-timestamp>
```

按如下方式替换这些命令中的示例值：

`<node-name>`

如果一个或多个 master 节点没有处于 **Ready** 状态，使用这个参数指定一个仍然处于 **Ready** 状态的 master 节点名称。这可避免调度错误，确保 **must-gather** pod 没有调度到未就绪的 master 节点上。

<directory-name>

must-gather 收集的信息的目录。

<duration>

指定收集信息的时长（相对时长），例如 **5h**（代表从 5 小时以前开始）。

<rfc3339-timestamp>

指定收集信息的时常（RFC 3339 时间戳），例如 **2020-11-10T04:00:00+00:00**（代表从 2020 年 11 月 11 日 4am UTC 开始）。

第 3 章 故障排除所需的常见日志

下方列出了一些用于对 OpenShift Container Storage 进行故障排除的日志，以及生成它们的命令。

- 为特定 pod 生成日志：

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

- 为 Ceph 或 OpenShift Container Storage 集群生成日志：

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

- 为 cephfs 或 rbd 等插件 pod 生成日志，以检测 app-pod 挂载中的任何问题：

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

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

- 为 CSI pod 中的所有容器生成日志：

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

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

- 为 cephfs 或 rbd provisioner pod 生成日志，以检测 PVC 不处于 **BOUND** 状态的问题：

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

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

- 为 CSI pod 中的所有容器生成日志：

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

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

- 使用 cluster-info 命令生成 OpenShift Container Storage 日志：

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

- 检查 OpenShift Container Storage operator 日志和事件。

- 检查 Operator 日志：

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

```
<ocs-operator>
```

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

- 检查 Operator 事件：

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

- 获取 OpenShift Container Storage operator 版本和频道。

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

输出示例：

```
NAME                DISPLAY VERSION      REPLACES
PHASE
ocs-operator.v4.6.2  OpenShift Container Storage 4.6.2
Succeeded
```

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

输出示例：

```
NAME      PACKAGE      SOURCE
CHANNEL
ocs-operator ocs-operator redhat-operators
stable-4.6
```

- 确认已创建了安装计划。

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

- 在更新 OpenShift Container Storage 后验证组件的镜像。

- 检查您要在其上验证镜像运行的组件 pod 的节点。

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

例如：

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

输出示例：

```
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 是 node-name。

- 检查镜像 ID。

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

<node-name>

是您要验证镜像运行的组件 pod 的节点名称。

```
# chroot /host
```

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

例如：

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

输出示例：

```
IMAGE                                TAG
  IMAGEID      SIZE
registry.redhat.io/ocs4/rook-ceph-rhel8-operator@sha256:5600a36370df4  <none>
5600a36370df4  1.55GB
```

记录 **IMAGEID**，并将其映射到 [Rook Ceph Operator](#) 页面中的 **Digest ID**。

其他资源

- [使用 must-gather](#)

第 4 章 为 OPENSIFT CONTAINER STORAGE 部署后覆盖集群范围默认节点选择器

当 Openshift Container Storage 使用集群范围默认节点选择器时，CSI daemonset 生成的 pod 只能在与选择器匹配的节点上启动。要从与选择器不匹配的节点使用 Openshift Container Storage，请在命令行界面中执行以下步骤来覆盖 **集群范围默认节点选择器**：

步骤

1. 为 openshift-storage 命名空间指定一个空白节点选择器。

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

2. 删除 DaemonSet 生成的原始 pod。

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

第 5 章 对 OPENSIFT CONTAINER STORAGE 中的警报和错误进行故障排除

5.1. 解决警报和错误

Red Hat OpenShift Container Storage 可以检测和自动解决很多常见的故障场景。但是，有些问题需要管理员介入。

要了解当前触发的错误，请查看以下位置之一：

- **Monitoring** → **Alerting** → **Firing** 选项
- **Home** → **Overview** → **Overview** 选项卡
- **Home** → **Overview** → **Persistent Storage** 标签页
- **Home** → **Overview** → **Object Service** 标签页

复制显示的错误并在以下部分搜索它以了解其严重性和解决方案：

Name:CephMonVersionMismatch

消息：运行多个版本的存储服务。

描述：运行有 {{ \$value }} 不同的 Ceph Mon 组件版本。

严重性: Warning

解决方案：修复

流程：检查用户界面和日志，并验证更新是否已进行中。

- 如果更新正在进行，则此警报是临时的。
- 如果更新没有进行，重启升级过程。

Name:CephOSDVersionMismatch

消息：运行多个版本的存储服务。

描述：运行有 {{ \$value }} 的不同版本。

严重性: Warning

解决方案：修复

流程：检查用户界面和日志，并验证更新是否已进行中。

- 如果更新正在进行，则此警报是临时的。
- 如果更新没有进行，重启升级过程。

名称 : CephClusterCriticallyFull

消息 : 存储集群非常全面, 需要立即扩展

描述 : 存储集群利用率已跨 85%。

严重性 : Critical

解决方案 : 修复

流程 : 删除不必要的数据库或扩展集群。

名称 : CephClusterNearFull

修复了:Storage 集群已满。需要进行扩展。

描述 : 存储集群利用率已跨 75%。

严重性: Warning

解决方案 : 修复

流程 : 删除不必要的数据库或扩展集群。

Name:NooBaaBucketErrorState

Message:A NooBa Bucket Is In Error State

Description:A NooBaa bucket {{ \$labels.bucket_name }} is in error state for more than 6m

严重性: Warning

解决方案 : 临时解决方案

流程 : [解决 NooBaa Bucket Error State](#)

Name: NooBaaBucketExceedingQuotaState

Message:A NooBaa Bucket Is In Exceeding Quota State

Description : A NooBaa bucket {{ \$labels.bucket_name }} its quota - {{ printf "%0.0f" \$value }}%
used used message: A NooBaa Bucket Is In Exceeding Quota State

严重性: Warning

解决方案 : 修复

流程 : [解决 NooBaa Bucket Exceeding Quota State](#)

Name:**NooBaaBucketLowCapacityState**

Message:**A NooBa Bucket Is In Low Capacity State**

Description : **A NooBaa bucket {{ \$labels.bucket_name }} 使用 {{ printf "%0.0f" \$value }}% 的容量**

严重性: Warning

解决方案 : 修复

流程 : [解决 NooBaa Bucket Capacity](#) 或 [Quota State](#)

Name:**NooBaaBucketNoCapacityState**

Message:**A NooBa Bucket Is In No Capacity State**

Description : **A NooBaa bucket {{ \$labels.bucket_name }} 正在使用其所有容量**

严重性: Warning

解决方案 : 修复

流程 : [解决 NooBaa Bucket Capacity](#) 或 [Quota State](#)

Name: **NooBaaBucketReachingQuotaState**

Message:**A NooBa Bucket Is Inaching Quota State**

Description : **A NooBaa bucket {{labels.bucket_name }} 使用 {{ printf "%0.0f" \$value }}% of its quota**

严重性: Warning

解决方案 : 修复

流程 : [解决 NooBaa Bucket Capacity](#) 或 [Quota State](#)

Name:**NooBaaResourceErrorState**

Message:**A NooBaa Resource Is In Error State**

Description : **A NooBaa resource {{labels.resource_name }} is in error state for more than 6m**

严重性: Warning

解决方案 : 临时解决方案

流程 : [解决 NooBaa Bucket Error State](#)

Name: NooBaaSystemCapacityWarning100

Message : A NooBaa System Approached it Capacity

描述 : 一个 NooBaa 系统接近其容量, 使用量为 100%

严重性: Warning

解决方案 : 修复

流程 : [解决 NooBaa Bucket Capacity](#) 或 [Quota State](#)

Name: NooBaaSystemCapacityWarning85

Message : A NooBaa System Is Approaching it Capacity

描述 : NooBaa 系统接近其容量, 使用量超过 85%

严重性: Warning

解决方案 : 修复

流程 : [解决 NooBaa Bucket Capacity](#) 或 [Quota State](#)

Name: NooBaaSystemCapacityWarning95

Message : A NooBaa System Is Approaching it Capacity

描述 : NooBaa 系统接近其容量, 使用量超过 95%

严重性: Warning

解决方案 : 修复

流程 : [解决 NooBaa Bucket Capacity](#) 或 [Quota State](#)

名称 : CephMdsMissingReplicas

Message : 存储元数据服务的 Insufficient 副本。

Description: `Minimum required replicas for storage metadata service not available.

可能会影响存储集群的工作。

严重性: Warning

解决方案 : [请联系红帽支持](#)

流程 :

1. 检查警报和操作器状态。
2. 如果无法识别该问题, 请联系[红帽支持团队](#)。

名称 : CephMgrIsAbsent

Message : 存储指标收集器服务不再可用。

Description : Ceph Manager 已经从 Prometheus 目标发现中消失。

严重性: Critical

解决方案 : [请联系红帽支持](#)

流程 :

1. 检查用户界面并记录，并验证更新是否正在进行。
 - 如果更新正在进行，则此警报是临时的。
 - 如果更新没有进行，重启升级过程。
2. 升级完成后，检查警报和 Operator 状态。
3. 如果问题持久或无法识别，[请联系红帽支持](#)。

名称 : CephNodeDown

Message : Storage node {{ \$labels.node }} down

Description:Storage node {{ \$labels.node }} down。请立即检查节点。

严重性: Critical

解决方案 : [请联系红帽支持](#)

流程 :

1. 检查哪个节点停止正常运行，并检查其原因。
2. 采取适当的操作来恢复节点。如果无法恢复节点：
 - 请参阅 [替换 OpenShift Container Storage 的存储节点](#)
 - [联系红帽支持部门](#)。

名称 : CephClusterErrorState

Message : Storage 集群处于错误状态

描述 : 存储集群处于错误状态, 状态为 10m。

严重性: Critical

解决方案 : 请联系红帽支持

流程 :

1. 检查警报和操作器状态。
2. 如果无法识别该问题, 请使用 [must-gather](#) 下载日志文件和诊断信息。
3. 向[红帽支持](#)创建一个支持问题单, 并附加 must-gather 的输出。

名称 : CephClusterWarningState

Message : Storage cluster 处于 degraded 状态

描述 : 存储集群处于警告状态, 状态为 10m。

严重性: Warning

解决方案 : 请联系红帽支持

流程 :

1. 检查警报和操作器状态。
2. 如果无法识别该问题, 请使用 [must-gather](#) 下载日志文件和诊断信息。
3. 向[红帽支持](#)创建一个支持问题单, 并附加 must-gather 的输出。

名称 : CephDataRecoveryTakingTooLong

消息 : 数据恢复速度慢

描述 : 数据恢复太长。

严重性: Warning

解决方案 : 请联系红帽支持

名称 : CephOSDDiskNotResponding

消息 : 磁盘没有响应

Description : Disk device `{{ $labels.device }}` not response on host `{{ $labels.host }}`。

严重性: Critical

解决方案 : 请联系红帽支持

名称 : **CephOSDDiskUnavailable**

Message : **Disk 无法访问**

Description : **Disk device {{ \$labels.device }} 在主机 {{ \$labels.host }} 上无法访问。**

严重性: Critical

解决方案 : [请联系红帽支持](#)

名称 : **CephPGRepairTakingTooLong**

消息 : **检测到的自修复问题**

描述 : **进行自我修复操作时间过长。**

严重性: Warning

解决方案 : [请联系红帽支持](#)

名称 : **CephMonHighNumberOfLeaderChanges**

Message : **Storage Cluster 最近看到了许多领导变化。**

Description : **'Ceph Monitor "{{ \$labels.job }}" instance {{ \$labels.instance }} have seen {{ \$value printf "%.2f" }} leader changes, recently.'**

严重性: Warning

解决方案 : [请联系红帽支持](#)

名称 : **CephMonQuorumAtRisk**

Message : **Storage quorum in risk**

描述 : **存储群集仲裁很低。**

严重性: Critical

解决方案 : [请联系红帽支持](#)

Name:**ClusterObjectStoreState**

Message:**Cluster Object Store 处于不健康状态。请检查 Ceph 集群健康状态。**

描述 : **Cluster Object Store 处于不健康状态，超过 15s。请检查 Ceph 集群健康状态。**

严重性: Critical

解决方案 : [请联系红帽支持](#)

流程 :

- 检查 **CephObjectStore** CR 实例。
- [联系红帽支持部门](#)。

5.2. 解决 NOOBAA BUCKET 错误状态

步骤

1. 登录 OpenShift Web 控制台，再单击 **Object Service**。
2. 在详细信息卡中，单击 **系统名称** 字段下的链接。
3. 在左侧窗格中，单击 **Buckets** 选项并搜索处于错误状态的存储桶。
4. 点它的 **Bucket Name**。此时会显示存储桶中遇到的错误。
5. 根据存储桶的具体错误，执行以下操作之一或两者：
 - a. 对于与空间相关的错误：
 - i. 在左侧窗格中，点 **Resources** 选项。
 - ii. 单击处于错误状态的资源。
 - iii. 通过添加更多代理来缩放资源。
 - b. 对于资源健康错误：
 - i. 在左侧窗格中，点 **Resources** 选项。
 - ii. 单击处于错误状态的资源。
 - iii. 连接错误意味着后备服务不可用，需要恢复。
 - iv. 如需访问/权限错误，请更新连接的访问密钥和机密密钥。

5.3. 解决 NOOBAA BUCKET EXCEEDING QUOTA STATE 问题

要解决 **A NooBaa Bucket Is In Exceeding Quota State** 错误，请执行以下操作之一：

- 清理存储桶上的一些数据。
- 执行以下步骤增加存储桶配额：
 1. 登录 OpenShift Web 控制台，再单击 **Object Service**。
 2. 在详细信息卡中，单击 **系统名称** 字段下的链接。
 3. 在左侧窗格中，单击 **Buckets** 选项并搜索处于错误状态的存储桶。
 4. 点它的 **Bucket Name**。此时会显示存储桶中遇到的错误。
 5. 点 **Bucket Policies** → **Edit Quota** 并增加配额。

5.4. 解决 NOOBAA BUCKET CAPACITY 或 QUOTA STATE 问题

步骤

1. 登录 OpenShift Web 控制台，再单击 **Object Service**。

2. 在详细信息卡中，单击 **系统名称** 字段下的链接。
3. 在左侧窗格中，单击 **Resources** 选项并搜索 PV 池资源。
4. 对于容量低状态的 PV 池资源，请单击其 **资源名称**。
5. 编辑池配置并增加代理数量。

5.5. 恢复 POD

当第一个节点(例如 **NODE1**)因为出现问题而变为 NotReady 状态时，使用 ReadWriteOnce(RWO)访问模式的 PVC 的托管 pod 会尝试移到第二个节点（例如 **NODE2**），但由于 multi-attach 错误而卡住。在这种情况下，您可以通过下列步骤恢复 MON、OSD 和应用容器集：

步骤

1. 关闭 **NODE1**（从 AWS 或 vSphere 端）并确保 **NODE1** 完全关闭。
2. 使用以下命令，强制删除 **NODE1** 上的 pod：

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

5.6. 从 EBS 卷分离中恢复

当 OSD 磁盘所驻留的 OSD 或 MON 弹性块存储(EBS)卷与工作程序 Amazon EC2 实例分离时，该卷会在一两分钟内自动重新附加。但是，OSD 容器集进入 **CrashLoopBackOff** 状态。若要将 pod 恢复并恢复为 **Running** 状态，您必须重新启动 EC2 实例。

第 6 章 检查 LOCAL STORAGE OPERATOR 部署

带有 Local Storage Operator 的 OpenShift Container Storage 集群使用本地存储设备部署。要找出带有 OpenShift Container Storage 的现有集群是否使用本地存储设备部署，请使用以下步骤：

先决条件

- OpenShift Container Storage 在 **openshift-storage** 命名空间内安装并运行。

步骤

通过检查与 OpenShift Container Storage 集群的持久性卷声明(PVC)关联的存储类，您可以确定集群是否使用本地存储设备部署。

1. 使用以下命令检查与 OpenShift Container Storage 集群的 PVC 关联的存储类：

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

2. 检查输出。对于使用 Local Storage Operator 的集群，与 **ocs-deviceset** 关联的 PVC 使用存储类 **localblock**。输出结果类似如下：

NAME	STATUS	VOLUME	CAPACITY	ACCESS
db-noobaa-db-0	Bound	pvc-d96c747b-2ab5-47e2-b07e-1079623748d8	50Gi	
ocs-deviceset-0-0-lzfrd	Bound	local-pv-7e70c77c	1769Gi	RWO
ocs-deviceset-1-0-7rggl	Bound	local-pv-b19b3d48	1769Gi	RWO
ocs-deviceset-2-0-znhk8	Bound	local-pv-e9f22cdc	1769Gi	RWO

其它资源

- [使用 AWS 上的本地存储设备部署 OpenShift Container Storage](#)
- [使用 VMware 上的本地存储设备部署 OpenShift Container Storage](#)

第 7 章 卸载过程中的故障排除和删除剩余的资源

有时，由 Operator 管理的一些自定义资源可能会处于 "Terminating" 状态，等待终结器完成，尽管您执行了所有必要的清理任务。在这种情况下，您需要强制删除这些资源。如果不这样做，资源仍会处于 "Terminating" 状态，即使您执行了所有卸载步骤。

1. 检查 openshift-storage 命名空间在删除时是否处于 Terminating 状态。

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

输出：

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

2. 在命令输出的 **STATUS** 部分检查 **NamespaceFinalizersRemaining** 和 **NamespaceContentRemaining** 信息，并对列出的每个资源执行下一步。

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

输出示例：

```
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. 删除上一步中列出的所有剩余资源。

对于要删除的每个资源，请执行以下操作：

- a. 获取需要删除的资源对象类型。查看以上输出中的消息。

示例：

```
message : 命名空间中的一些内容有余的终结器 : cephobjectstoreuser.ceph.rook.io
```

其中 `cephobjectstoreuser.ceph.rook.io` 是对象类型。

- b. 获取与对象类型对应的对象名称。

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

示例：

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

输出示例：

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

- c. 修补资源。

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

例如：

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

输出：

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

4. 验证 `openshift-storage` 项目是否已删除。

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

输出：

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

如果问题仍然存在，请[联系红帽支持团队](#)。

第 8 章 对外部模式的 CEPHFS PVC 创建进行故障排除

如果您已将红帽 Ceph 存储集群从低于 4.1.1 的版本更新为最新版本，且不是全新部署的集群，您必须在红帽 Ceph 存储集群中手动设置 CephFS 池的应用类型，以外部模式启用 CephFS PVC 创建。

1. 检查 CephFS pvc 处于 **Pending** 状态。

```
$ oc get pvc
```

输出示例：

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

2. 检查 **describe** 输出，以查看相应 pvc 的事件。

预期错误消息是 **cephfs_metadata/csi.volumes.default/csi.volume.pvc-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx:(1)Operation not allowed**

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

输出示例：

```
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-cephfspugin-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. 检查 **<cephfs metadata pool name>** (这里是 **cephfs_metadata**) 和 **<cephfs data pool name>** (这里是 **cephfs_data**)。为了运行命令，需要在 Red Hat Ceph Storage 客户端节点中预先安装 **jq**。

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

4. 设置 CephFS 池的应用类型。

- 在 Red Hat Ceph Storage 客户端节点中运行以下命令：

```
# 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. 验证是否应用了设置。

```
# 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. 再次检查 CephFS PVC 状态。PVC 现在处于 **Bound** 状态。

```
# oc get pvc
```

输出示例：

```
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
[...]
```

第 9 章 在 OPENSIFT CONTAINER STORAGE 中恢复监控 POD

如果所有三个容器集都停止，以及 OpenShift Container Storage 无法自动恢复监控 pod，则恢复监控 pod。

步骤

1. 缩减 **rook-ceph-operator** 和 **ocs operator** 部署。

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

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

2. 为 **openshift-storage** 命名空间中的所有部署创建备份。

```
# 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. 修补 OSD 部署以移除 **livenessProbe** 参数，再以命令参数作为 **sleep** 状态运行它。

```
# 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. 从所有 OSD 检索 **monstore** 集群映射。

- a. 创建 **restore_mon.sh** 脚本。

```
#!/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/podV//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. 运行 **restore_mon.sh** 脚本。

```

# chmod +x recover_mon.sh

# ./recover_mon.sh

```

5. 修补 MON 部署，并使用命令参数作为 **sleep** 状态运行它。

- a. 编辑 MON 部署。

```

# 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. 修补 MON 部署，以增加 **initialDelaySeconds**。

```

# 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. 将之前检索到的 **monstore** 复制到 **mon-a** pod。

```

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

```

7. 导航到 MON 容器集，再更改检索到的 **monstore** 的所有权。

```

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

```

```

# chown -R ceph:ceph /tmp/monstore

```

8. 在重建 MON DB 之前设置适当的能力。

```
# 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 = AQCleqldWqm5lhAAgZQbEzoShkZV42RiQVffnA==
caps mon = "allow *"
[client.admin]
key = AQCmAKld8J05KxAArOWeRAw63gAwwZO5o75ZNQ==
aid = 0
caps mds = "allow *"
caps mgr = "allow *"
caps mon = "allow *"
caps osd = "allow *"
```

9. 从对应的 secret 识别所有其他 Ceph 守护进程（MGR、MDS、RGW、Crash、CSI 和 CSI 置备程序）的密钥环。

```
# 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"
```

keyring 文件示例：**/etc/ceph/ceph.client.admin.keyring**：

```
[mon.]
key = AQDxTF1hNgLTNxAAi51cCojs01b4I5E6v2H8Uw==
caps mon = "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.admin] key =
AQDxTF1hpzguOxAA0sS8nN4udoO35OEbt3bqMQ== caps mds = "allow *" caps mgr =
"allow *" caps mon = "allow *" caps osd = "allow *" [client.crash] key =
AQBOTV1htO1aGRAAe2MPYcGdiAT+Oo4CNPSF1g== caps mgr = "allow rw" caps mon =
"allow profile crash" [client.csi-cephfs-node] key =
AQBOTV1hiAtuBBAAaPPBVgh1AqZJIDeHWdoFLw== 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"
[mgr.a]
key = AQBOTV1hGYOERAA87471+eIZLZtptfkchVTRg==
caps mds = "allow *"
caps mon = "allow profile mgr"
caps osd = "allow *"
```



重要

- 对于 **client.csi** 相关的密钥环，请在从其相应的 OpenShift Container Storage secret 获取密钥后添加默认 **caps**。
- OSD 密钥环会在恢复后自动添加。

10. 进入到 **mon-a** 容器集。
验证 **monstore** 有 **monmap**。

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

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

如果缺少 **monmap**，则创建新的 **monmap**。

```
# 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>

是 mon-a pod 的 ID

<mon-a-ip>

是 mon-a pod 的 IP 地址

<mon-b-id>

是 mon-b pod 的 ID

<mon-b-ip>

是 mon-b pod 的 IP 地址

<mon-c-id>

是 mon-c pod 的 ID

<mon-c-ip>

是 mon-c pod 的 IP 地址

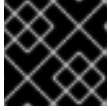
<fsid>

是文件系统 ID

11. 验证 **monmap**。

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

12. 导入 **monmap**。

**重要**

使用之前创建的 **keyring** 文件。

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

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

13. 创建旧 **store.db** 文件的备份。

```
# 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
```

14. 将重新构建 **store.db** 文件复制到 **monstore** 目录。

```
# 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
```

15. 在重建了 **monstore** 目录后，将 **store.db** 文件从本地复制到 MON 容器集的其余部分。

```
# 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>

是 MON Pod 的 ID

16. 前往 MON 容器集的其余部分，再更改复制的 **monstore** 的所有权。

```
# 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>

是 MON Pod 的 ID

17. 恢复补丁的更改。

- 对于 MON 部署：

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

<mon-deployment.yaml>

是 MON 部署 yaml 文件

- 对于 OSD 部署：

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

<osd-deployment.yaml>

是 OSD 部署 yaml 文件

- 对于 MGR 部署：

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

<mgr-deployment.yaml>

是 MGR 部署 yaml 文件



重要

确保 MON、MGR 和 OSD 容器集已启动并在运行。

18. 扩展 **rook-ceph-operator** 和 **ocs-operator** 部署。

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

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

恢复 CephFS

检查 Ceph 状态，以确认 CephFS 正在运行。

```
# ceph -s
```

输出示例：

```
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
```

如果文件系统离线或者缺少 MDS 服务，请执行以下步骤：

1. 缩减 **rook-ceph-operator** 和 **ocs operator** 部署。

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

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

2. 修补 MDS 部署以移除 **livenessProbe** 参数，并使用命令参数运行该参数作为 **sleep** 状态。

```
# for i in $(oc get deployment -l app=rook-ceph-mds -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": "mds", "command": ["sleep", "infinity"], "args": []}]}}}' ; done
```

3. 恢复 CephFS.

```
# ceph fs reset ocs-storagecluster-cephfilesystem --yes-i-really-mean-it
```

如果 **reset** 命令失败，请强制使用数据和元数据池创建默认文件系统，然后重置它。



注意

如果 **cephfilesystem** 缺失，则 **reset** 命令可能会失败。

```
# ceph fs new ocs-storagecluster-cephfilesystem ocs-storagecluster-cephfilesystem-metadata ocs-storagecluster-cephfilesystem-data0 --force
```

```
# ceph fs reset ocs-storagecluster-cephfilesystem --yes-i-really-mean-it
```

4. 替换 MDS 部署。

```
# oc replace --force -f oc_get_deployment.rook-ceph-mds-ocs-storagecluster-cephfilesystem-a.yaml
```

```
# oc replace --force -f oc_get_deployment.rook-ceph-mds-ocs-storagecluster-cephfilesystem-b.yaml
```

5. 扩展 **rook-ceph-operator** 和 **ocs-operator** 部署。

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

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

6. 检查 CephFS 状态。

```
# ceph fs status
```

状态应当为 **active**。

重要

如果连接到使用 CephFS 持久性卷声明(PVC)的部署的应用容器集在恢复 CephFS 后处于 **CreateContainerError** 状态，请重启应用容器集。

```
# oc -n <namespace> delete pods <cephfs-app-pod>
```

<namespace>

是项目的命名空间

<cephfs-app-pod>

是 CephFS 应用 pod 的名称

恢复 Multicloud 对象网关

在恢复 MON pod 后，检查 Multicloud 对象网关(MCG)状态，它应处于活动状态，并且后备存储和 bucket 类应当处于 **Ready** 状态。如果没有，重启所有与 MCG 相关的 pod，并检查 MCG 状态以确认 MCG 是否已备份并运行。

1. 检查 MCG 状态。

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

2. 重启与 MCG 相关的所有 pod。

```
# 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>

是 MCG operator 的名称

<noobaa-core>

是 MCG 内核 pod 的名称

<noobaa-endpoint>

是 MCG 端点的名称

<noobaa-db>

是 MCG db pod 的名称

3. 如果配置了 RADOS 对象网关(RGW)，请重新启动容器集。

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

<rgw-pod>

是 RGW pod 的名称

