Red Hat OpenShift Data Foundation 4.15

OpenShift Data Foundation 故障排除

有关 OpenShift Data Foundation 故障排除的说明

Last Updated: 2024-02-19

Red Hat OpenShift Data Foundation 4.15 OpenShift Data Foundation 故障排除

有关 OpenShift Data Foundation 故障排除的说明

法律通告

摘要

阅读本文档，了解有关 Red Hat OpenShift Data Foundation 故障排除的说明。

目录

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 CTO Chris Wright 的信息。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请告诉我们如何让它更好。

要提供反馈，请创建一个 Bugzilla ticket：

1. 进入 Bugzilla 网站。
2. 在 **Component** 部分中，选择 **文档**。
3. 在 **Description** 中输入您要提供的信息。包括文档相关部分的链接。
4. 点 **Submit Bug**。

第 1 章 概述

OpenShift Data Foundation 故障排除旨在帮助管理员了解如何排除故障并修复其 Red Hat OpenShift Data Foundation 集群。

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* 如何在 OpenShift Data Foundation 中使用 must-gather 实用程序。
- 第 3 章 *故障排除所需的常见日志* 如何获取 OpenShift Data Foundation 所需的日志文件。
- 第 6 章 *对 OpenShift Data Foundation 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。

警告

红帽不支持在 OpenShift Data Foundation 集群中运行 Ceph 命令（除非由红帽支持或红帽文档表示），因为在运行错误的命令时可能会导致数据丢失。在这种情况下，红帽支持团队只能提供商业合理的工作，在出现数据丢失时可能无法恢复所有数据。

第 2 章 使用 MUST-GATHER 下载日志文件和诊断信息

如果 Red Hat OpenShift Data Foundation 无法自动解决问题，请使用 **must-gather** 工具收集日志文件和诊断信息，以便您或红帽支持可以审核问题并确定解决方案。

重要

当将 Red Hat OpenShift Data Foundation 部署为外部模式时，**must-gather** 仅从 OpenShift Data Foundation 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。

先决条件

- 可选：如果 OpenShift Data Foundation 在断开连接的环境中部署，请确保将独立的 **must-gather** 镜像镜像(mirror)到断开连接的环境中可用的镜像 registry。

```
$ oc image mirror registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14 <local-registry>/odf4/odf-must-gather-rhel9:v4.14 [--registry-config=<path-to-the-registry-config>] [--insecure=true]
```

<local-registry>

是本地镜像 registry 可用于断开连接的 OpenShift Container Platform 集群。

<path-to-the-registry-config>

是 registry 凭证的路径，默认为 **~/.docker/config.json**。

--insecure

仅在镜像 registry 不安全时才添加此标志。

如需更多信息，请参阅红帽知识库解决方案：

- 如何在 Redhat Openshift registry 间镜像镜像
- 私有 registry 不安全时镜像 OpenShift 镜像存储库失败

流程

- 从连接到 OpenShift Data Foundation 集群的客户端运行 **must-gather** 命令：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14 --dest-dir=<directory-name>
```

<directory-name>

是要将数据写入的目录的名称。

重要

对于断开连接的环境部署，将 **--image** 参数中的镜像替换为镜像的 **must-gather** 镜像。

```
$ oc adm must-gather --image=<local-registry>/odf4/odf-must-gather-rhel9:v4.14 --dest-dir=<directory-name>
```

<local-registry>

是本地镜像 registry 可用于断开连接的 OpenShift Container Platform 集群。

这会在指定目录中收集以下信息：

- 所有与 Red Hat OpenShift Data Foundation 集群相关的自定义资源(CR)及其命名空间。
- 所有 Red Hat OpenShift Data Foundation 相关 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/odf4/odf-must-gather-rhel9:v4.14 --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/odf4/odf-must-gather-rhel9:v4.14 --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/odf4/odf-must-gather-rhel9:v4.14 --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>

指定收集来自 RFC 3339 时间戳的信息的时间周期，例如 **2020-11-10T04:00:00+00:00** （从 2020 年 11 月 11 日的 UTC 开始）。

第 3 章 故障排除所需的常见日志

其中列出了一些用于对 OpenShift Data Foundation 进行故障排除的常用日志，以及用于生成这些日志的命令。

- 为特定 pod 生成日志：

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

- 为 Ceph 或 OpenShift Data Foundation 集群生成日志：

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

重要

目前，rook-ceph-operator 日志不提供有关故障的任何信息，这在故障排除中可作为限制，请参阅为 rook-ceph-operator 启用和禁用 debug 日志。

- 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 Data Foundation 日志：

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

- 使用 Local Storage Operator 时，可以使用 cluster-info 命令生成日志：

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

- 检查 OpenShift Data Foundation 操作器日志和事件。
  - 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 Data Foundation 操作器版本和渠道。

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

  输出示例：

  ```
  NAME                         DISPLAY                       VERSION   REPLACES   PHASE
  mcg-operator.v4.14.0         NooBaa Operator               4.14.0               Succeeded
  ocs-operator.v4.14.0         OpenShift Container Storage   4.14.0               Succeeded
  odf-csi-addons-operator.v4.14.0   CSI Addons                    4.14.0               Succeeded
  odf-operator.v4.14.0         OpenShift Data Foundation     4.14.0               Succeeded
  ```

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

  输出示例：

  ```
  NAME                                              PACKAGE                SOURCE
  CHANNEL
  mcg-operator-stable-4.14-redhat-operators-openshift-marketplace   mcg-operator
  redhat-operators   stable-4.14
  ocs-operator-stable-4.14-redhat-operators-openshift-marketplace   ocs-operator
  redhat-operators   stable-4.14
  odf-csi-addons-operator                                odf-csi-addons-operator   redhat-operators
  stable-4.14
  odf-operator                                           odf-operator           redhat-operators   stable-
  4.14
  ```

- 确认已创建了 **安装计划**。

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

- 在更新 OpenShift Data Foundation 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
```

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

3.1. 调整日志的详细程度

调试日志消耗的空间量可能会成为严重的问题。Red Hat OpenShift Data Foundation 提供了一种调整方法，因此可以控制调试日志消耗的存储量。

要调整调试日志的详细程度，您可以调整负责容器存储接口(CSI)操作的容器的日志级别。在容器的 yaml 文件中，调整以下参数来设置日志级别：

- **CSI_LOG_LEVEL** - 默认为 **5**
- **CSI_SIDECAR_LOG_LEVEL** - 默认为 **1**

支持的值从 **0** 到 **5**。**0** 作为常规使用的日志，**5** 具有追踪级别的详细程度。

第 4 章 部署后覆盖 OPENSHIFT DATA FOUNDATION 的集群范围默认节点选择器

当将集群范围的默认节点选择器用于 OpenShift Data Foundation 时，由容器存储接口(CSI) daemonset 生成的 pod 只能在与选择器匹配的节点上启动。要能够从与选择器不匹配的节点使用 OpenShift Data Foundation，请在命令行界面中执行以下步骤来覆盖**集群范围的默认节点选择器** ：

流程

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 加密令牌已删除或过期

如果密钥管理系统的加密令牌被删除或过期，请使用这个流程更新令牌。

先决条件

- 确保您有一个与已删除或过期令牌相同的策略的新令牌

流程

1. 登录 OpenShift Container Platform Web 控制台。
2. 点 **Workloads → Secrets**
3. 更新用于集群范围加密的 **ocs-kms-token** ：
   a. 将 **Project** 设置为 **openshift-storage**。
   b. 点 **ocs-kms-token → Actions → Edit Secret**。
   c. 在 **Value** 字段中拖放或上传您的加密令牌文件。令牌可以是可复制和粘贴的文件或文本。
   d. 点 **Save**。
4. 为带有加密持久性卷的给定项目或命名空间更新 **ceph-csi-kms-token** ：
   a. 选择所需的项目。
   b. 点 **ceph-csi-kms-token → Actions → Edit Secret**。
   c. 在 **Value** 字段中拖放或上传您的加密令牌文件。令牌可以是可复制和粘贴的文件或文本。
   d. 点 **Save**。

注意

只有在所有使用 **ceph-csi-kms-token** 的加密 PVC 已被删除后，才能删除令牌。

第 6 章 对 OPENSHIFT DATA FOUNDATION 中的警报和错误进行故障排除

6.1. 解决警报和错误

Red Hat OpenShift Data Foundation 可以检测并自动解决许多常见的故障情形。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：

- **Observe → Alerting → Firing** 选项
- **Home → Overview → Cluster** 标签页
- **Storage → Data Foundation → Storage System →** *storage system* 链接，在弹出的 **→ Overview → Block and File** 标签页
- **Storage → Data Foundation → Storage System →** Storage *system* 链接，在弹出 **→ Overview → Object** 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**

Message: **运行多个存储服务版本。**

Description：**{{ $value }} 运行的 Ceph Mon 组件的不同版本。**

严重性: 警告

解决方案 ：修复

流程 ：检查用户界面和日志，并验证更新是否进行中。

- 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:**CephOSDVersionMismatch**

Message: **运行多个存储服务版本。**

Description：**{{ $value }} 运行的 Ceph OSD 组件的不同版本。**

严重性: 警告

解决方案 ：修复

流程 ：检查用户界面和日志，并验证更新是否进行中。

- 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 ：**存储集群几乎已满，需要立即扩展**

描述 ：**存储集群利用率已超过 85%。**

严重性: 关键

解决方案 ：修复

流程 ：删除不必要的数据或扩展集群。

名称 ：**CephClusterNearFull**

修复了:Storage 集群已接近满。需要进行扩展。

描述 ：**存储集群利用率已超过 75%**

严重性: 警告

解决方案 ：修复

流程 ：删除不必要的数据或扩展集群。

Name:**NooBaaBucketErrorState**

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

Description **: NooBaa bucket {{ $labels.bucket_name }}** 处于错误状态，超过 **6m**

严重性: 警告

解决方案 ：临时解决方案

步骤 ：解决 NooBaa Bucket Error State

名称:**NooBaaNamespaceResourceErrorState**

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

描述 : NooBaa 命名空间资源 {{ $labels.namespace_resource_name }} 处于错误状态，表示 **5m** 的错误状态

严重性: 警告

解决方案 ：修复

步骤 ：解决 NooBaa Bucket Error State

Name:**NooBaNamespaceBucketErrorState**

Message:**A NooBaa Namespace Bucket Is In Error State**

Description **: NooBaa 命名空间存储桶 {{ $labels.bucket_name }} 处于错误状态，超过 5m**

严重性: 警告

解决方案 ：修复

步骤 ：解决 NooBaa Bucket Error State

名称:NooBaaBucketExceedingQuotaState

Message **: NooBaa Bucket In Exceeding Quota State**

Description **: NooBaa bucket {{ $labels.bucket_name }} 超过其配额 - {{ printf "%0.0f" $value }}% 使用的消息： NooBaa Bucket Is In Exceeding Quota State**

严重性: 警告

解决方案 ：修复

步骤 ：解决 NooBaa Bucket Exceeding Quota State

Name:**NooBaaBucketLowCapacityState**

Message **: NooBaa Bucket Is In Low Capacity State**

Description **: NooBaa bucket {{ $labels.bucket_name }} 正在为其容量使用 {{ printf "%0.0f" $value }}%**

严重性: 警告

解决方案 ：修复

步骤 ：解决 NooBaa Bucket Capacity 或 Quota State

Name:**NooBaaBucketNoCapacityState**

Message **: NooBaa Bucket Is In Capacity State**

描述 : NooBaa 存储桶 {{ $labels.bucket_name }} 使用其所有容量

严重性: 警告

解决方案 ：修复

步骤 ：解决 NooBaa Bucket Capacity 或 Quota State

Name:**NooBaaBucketReachingQuotaState**

消息 ：NooBaa Bucket Is In Reaching Quota State

Description **: NooBaa bucket {{ $labels.bucket_name }} 正在为其配额使用 {{ printf "%0.0f" $value }}%**

严重性: 警告

解决方案 ：修复

步骤 ：解决 NooBaa Bucket Capacity 或 Quota State

Name:**NooBaaResourceErrorState**

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

描述 : NooBaa resource {{ $labels.resource_name }} 处于错误状态，超过 6m

严重性: 警告

解决方案 ：临时解决方案

步骤 ：解决 NooBaa Bucket Error State

Name:**NooBaaSystemCapacityWarning100**

消息 ：NooBaa System Approached Its Capacity

描述 ：NooBaa 系统接近其容量，使用量为 100%

严重性: 警告

解决方案 ：修复

步骤 ：解决 NooBaa Bucket Capacity 或 Quota State

Name:**NooBaaSystemCapacityWarning85**

Message **: NooBaa System Is Approaching it Capacity**

描述 ：NooBaa 系统接近其容量，使用时间超过 85%

严重性: 警告

解决方案 ：修复

步骤 ：解决 NooBaa Bucket Capacity 或 Quota State

名称 ：NooBaaSystemCapacityWarning95

Message **: NooBaa System Is Approaching it Capacity**

描述 ：NooBaa 系统接近其容量，使用时间超过 95%

严重性: 警告

解决方案 ：修复

步骤 ：解决 NooBaa Bucket Capacity 或 Quota State

Name:**CephMdsMissingReplicas**

Message ：用于存储元数据服务的不计副本。

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: 警告

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:**Storage 指标收集器服务不再可用。**

描述 : Ceph Manager 从 Prometheus 目标发现中消失。

严重级别: 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. 如果问题仍然存在或无法识别，请联系红帽支持。

名称 ：CephNodeDown

Message: **Storage node {{ $labels.node }} 停机**

描述:Storage node {{ $labels.node }} 停机。立即检查节点。

严重级别: 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：
   - 请参阅为 Red Hat OpenShift Data Foundation 替换存储节点
   - 联系红帽支持部门.

名称 ：CephClusterErrorState

消息:Storage cluster is in error state

Description ：**存储集群处于错误状态超过 10m。**

严重级别: 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:Storage cluster 处于 warning 状态，表示 10m 以上的警告状态。

严重性: 警告

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

Message: **Data recovery is slow**

描述 ：数据恢复时间过长。

严重性: 警告

解决方案 ：请联系红帽支持

名称 ：CephOSDDiskNotResponding

Message:**Disk not respond**

描述 : 磁盘设备 {{ $labels.device }} 未响应，在主机 {{ $labels.host }} 上。

严重级别: Critical

解决方案 ：请联系红帽支持

名称 ：CephOSDDiskUnavailable

Message:**Disk not access**

描述:**磁盘设备 {{ $labels.device }} 无法在主机 {{ $labels.host }} 上访问。**

严重级别: Critical

解决方案 ：请联系红帽支持

名称 ：CephPGRepairTakingTooLong

Message:**检测到的自助修复问题**

描述 ：执行自助服务修复操作用时过长。

严重性: 警告

解决方案 ：请联系红帽支持

Name:**CephMonHighNumberOfLeaderChanges**

Message:**Storage Cluster 最近看到很多领导变化。**

描述:**'Ceph Monitor "{{ $labels.job }}": instance {{ $labels.instance }} 已看到 {{ $value printf "%.2f" }} leader 每分钟更改。 '**

严重性: 警告

解决方案 ：请联系红帽支持

名称 ：**CephMonQuorumAtRisk**

消息 ：**存储仲裁的风险**

描述 ：**存储群集仲裁较低。**

严重级别: Critical

解决方案 ：请联系红帽支持

名称 ：**ClusterObjectStoreState**

消息:**Cluster Object Store is in unhealthy state.检查 Ceph 集群健康状况。**

描述:**Cluster Object Store is in unhealthy state for more than 15s.检查 Ceph 集群健康状况。**

严重级别: 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.

名称 ：**CephOSDFlapping**

Message: **Storage daemon osd.x has restarted 5 times in the last 5 minutes.检查 pod 事件或 Ceph 状态以查找原因。**

描述 ：**Storage OSD 在 5 分钟内重新启动超过 5 次。**

严重级别: Critical

解决方案 ：请联系红帽支持

名称 ：**OdfPoolMirroringImageHealth**

Message:**Mirroring image(PV)位于池 <pool-name> 中，超过 1m。Mirroring might not work as expected.**

描述 ：对一个或多个应用程序失败。

严重性: 警告

解决方案 ：请联系红帽支持

名称 ：**OdfMirrorDaemonStatus**

消息 ：**Mirror 守护进程不健康。**

描述 ：对整个集群进行灾难恢复失败。Mirror daemon is in unhealthy status for more than 1m.Mirroring on this cluster is not working as expected.

严重级别: Critical

解决方案 ：请联系红帽支持

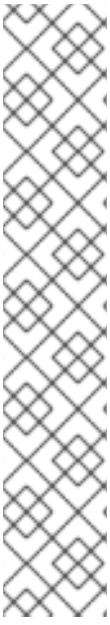
6.2. 解决集群健康问题

Red Hat Ceph Storage 可以在 OpenShift Data Foundation 用户界面中引发该显示的一系列有限健康消息。它们定义为具有唯一标识符的健康检查。标识符是一个制表伪可读字符串，旨在使工具能够理解健康检查，并以反应其含义的方式呈现它们。有关更多信息和故障排除，请单击下面的健康代码。

健康代码	描述
MON_DISK_LOW	一个或多个 Ceph 监控器在磁盘空间上较低。

6.2.1. MON_DISK_LOW

如果将 monitor 数据库存储为百分比的文件系统中的可用空间下降到 **mon_data_avail_warn** 下，则会触发此警报（默认：15%）。这可能表明系统上的某些其他进程或用户正在填满监控器使用的相同文件系统。也可能表明监控器的数据库比较大。



注意

文件系统的路径因您的 mon 部署而异。您可以找到在 **storagecluster.yaml** 中部署 mon 的路径。

路径示例：

- 通过 PVC 路径部署的 mon: **/var/lib/ceph/mon**
- 通过 hostpath 部署 mon: **/var/lib/rook/mon**

若要清除空间，请查看文件系统中的高使用量文件并选择要删除的文件。要查看文件，请运行：

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

将 **<path-in-the-mon-node>** 替换为部署 mons 的文件系统的路径。

6.3. 解决集群警报

Red Hat Ceph Storage 集群可以引发的一组有限健康警报，显示在 OpenShift Data Foundation 用户界面中。它们定义为具有唯一标识符的健康警报。标识符是一个制表伪可读字符串，旨在使工具能够理解健康检查，并以反应其含义的方式呈现它们。点健康警报以了解更多信息和故障排除。

表 6.1. 集群健康警报的类型

健康警报	概述
CephClusterCriticallyFull	存储集群利用率已超过 80%。
CephClusterErrorState	存储集群处于错误状态的时间已超过 10 分钟。
CephClusterNearFull	存储集群接近满容量。需要删除数据或集群扩展。
CephClusterReadOnly	存储集群现在是只读的，需要立即删除数据或集群扩展。
CephClusterWarningState	存储集群处于警告状态超过 10 分钟。
CephDataRecoveryTakingTooLong	Data recovery has been active for too long.
CephMdsMissingReplicas	存储元数据服务不可用所需的最小副本。可能会影响存储集群的工作。
CephMgrIsAbsent	Ceph Manager has disappeared from Prometheus target discovery.
CephMgrIsMissingReplicas	Ceph 管理器缺少副本。这会破坏健康状态报告，并将导致 **ceph status** 命令报告的一些信息缺失或过时。此外，Ceph 管理器负责一个管理器框架，旨在扩展 Ceph 的现有功能。
CephMonHighNumberOfLeaderChanges	Ceph 监控领导正在改变异常次数。
CephMonQuorumAtRisk	Storage cluster quorum is low.
CephMonQuorumLost	存储集群中的监控 pod 数量不够。
CephMonVersionMismatch	运行 Ceph Mon 组件的不同版本。
CephNodeDown	存储节点停机。立即检查节点。该警报应包含节点名称。
CephOSDCriticallyFull	后端对象存储设备(OSD)的利用率已超过 80%。立即释放一些空间或扩展存储集群或联系支持。
CephOSDDiskNotResponding	磁盘设备没有在其中一个主机上响应。
CephOSDDiskUnavailable	一个主机上无法访问磁盘设备。
CephOSDFlapping	Ceph 存储 OSD 阻塞。
CephOSDNearFull	其中一个 OSD 存储设备接近满。
CephOSDSlowOps	OSD 请求需要很长时间才能进行处理。

健康警报	概述
CephOSDVersionMismatch	运行 Ceph OSD 组件的不同版本。
CephPGRepairTakingTooLong	自我修复操作用时过长。
CephPoolQuotaBytesCriticallyExhausted	存储池配额使用量已超过 90%。
CephPoolQuotaBytesNearExhaustion	存储池配额使用量已超过 70%。
PersistentVolumeUsageCritical	持久性卷声明使用量已超过 85% 的容量。
PersistentVolumeUsageNearFull	持久性卷声明使用量已超过 75% 的容量。

6.3.1. CephClusterCriticallyFull

含义	存储集群利用率已超过 80%，并将在 85% 时变得只读。超过 85% 的利用率后，您的 Ceph 集群将变得只读。释放一些空间或立即扩展存储集群。在此警报之前，通常会看到与对象存储设备(OSD)已满或接近满相关的警报。
影响	高

诊断

扩展存储

根据集群的类型，您需要添加存储设备、节点或两者。有关更多信息，请参阅扩展存储指南。

缓解方案

删除信息

如果无法扩展集群，则需要删除信息来释放一些空间。

6.3.2. CephClusterErrorState

含义	此警报反映了存储集群在不可接受的时间处于 **ERROR** 状态，这会破坏存储可用性。检查之前触发的其他警报，并首先对这些警报进行故障排除。
影响	Critical

诊断

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
$ oc get pod | grep rook-ceph
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a rook-ceph that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod_name>
指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

- 如果分配了节点，请检查节点上的 kubelet。
- 如果正在运行的 pod 的基本健康状况，则验证节点上的节点关联性和资源可用性，请运行 Ceph 工具来获取存储组件的状态。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.3. CephClusterNearFull

含义	存储集群利用率已超过 75%，并将在 85% 时变得只读。释放一些空间或扩展存储集群。
影响	Critical

诊断

扩展存储

根据集群的类型，您需要添加存储设备、节点或两者。有关更多信息，请参阅扩展存储指南。

缓解方案

删除信息

如果无法扩展集群，则需要删除信息来释放一些空间。

6.3.4. CephClusterReadOnly

含义	存储集群利用率已超过 85%，现在将变为只读。释放一些空间或立即扩展存储集群。
影响	Critical

诊断

扩展存储

根据集群的类型，您需要添加存储设备、节点或两者。有关更多信息，请参阅扩展存储指南。

缓解方案

删除信息

如果无法扩展集群，则需要删除信息来释放一些空间。

6.3.5. CephClusterWarningState

含义	此警报反映了存储集群在不可接受的时间处于警告状态。虽然存储操作将继续在这个状态下正常工作，但建议修复错误，以便集群不会进入错误状态。检查之前可能触发的其他警报，并首先对这些警报进行故障排除。
影响	高

诊断

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
oc get pod | grep {ceph-component}
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a {ceph-component}  that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod_name>

指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

如果分配了节点，请检查节点上的 kubelet。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.6. CephDataRecoveryTakingTooLong

含义	数据恢复速度较慢。检查所有对象存储设备(OSD)是否已启动并运行。
影响	高

诊断

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
oc get pod | grep rook-ceph-osd
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a {ceph-component}  that is in the pending state, not running or
not ready
MYPOD=<pod_name>
```

<pod_name>

指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

如果分配了节点，请检查节点上的 kubelet。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.7. CephMdsMissingReplicas

含义	存储元数据服务(MDS)的最低副本不可用。MDS 负责填充元数据。MDS 服务的降级可能会影响存储集群的工作方式（与 CephFS 存储类相关），并应尽快修复。
影响	高

诊断

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
oc get pod | grep rook-ceph-mds
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a {ceph-component}  that is in the pending state, not running or
not ready
MYPOD=<pod_name>
```

<pod_name>

指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

如果分配了节点，请检查节点上的 kubelet。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.8. CephMgrIsAbsent

含义	没有 Ceph 管理器运行集群的监控。创建和删除请求应尽快解决持久性卷声明(PVC)创建和删除请求。
影响	高

诊断

- 验证 **rook-ceph-mgr** pod 失败，并在需要时重启。如果 Ceph mgr pod 重启失败，请遵循常规 pod 故障排除来解决这个问题。
  - 验证 Ceph mgr pod 失败：

```
$ oc get pods | grep mgr
```

  - 描述 Ceph mgr pod 以获取更多详细信息：

```
$ oc describe pods/<pod_name>
```

<pod_name>
指定上一步中的 **rook-ceph-mgr** pod 名称。

分析与资源问题相关的错误。

  - 删除 pod，并等待 pod 重启：

```
$ oc get pods | grep mgr
```

按照以下步骤进行常规 pod 故障排除：

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
oc get pod | grep rook-ceph-mgr
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a {ceph-component}  that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod_name>
指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

如果分配了节点，请检查节点上的 kubelet。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.9. CephMgrIsMissingReplicas

含义	要解决此警报，您需要确定 Ceph 管理器消失的原因，并在需要时重新启动。
影响	高

诊断

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
oc get pod | grep rook-ceph-mgr
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a {ceph-component}  that is in the pending state, not running or
not ready
MYPOD=<pod_name>
```

<pod_name>
指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

如果分配了节点，请检查节点上的 kubelet。

缓解方案

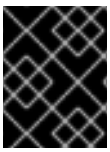
调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.10. CephMonHighNumberOfLeaderChanges

含义	在 Ceph 集群中，有一组冗余的 monitor pod，用于存储有关存储集群的重要信息。定期监控 pod 同步，以获取有关存储集群的信息。第一个监控 pod 获取最新更新的信息成为领导信息，其他监控容器集将在询问领导后启动其同步过程。在一个或多个 monitor pod 中网络连接或其他类型问题会造成不必要的变化。这种情形可能会对存储集群性能造成负面影响。
影响	Medium



重要

检查是否有网络问题。如果存在网络问题，则需要在进行以下任何故障排除步骤前升级到 OpenShift Data Foundation 团队。

诊断

1. 输出受影响监控 pod 的日志，以收集有关此问题的更多信息：

```
$ oc logs <rook-ceph-mon-X-yyyy> -n openshift-storage
```

<rook-ceph-mon-X-yyyy>

指定受影响的 monitor pod 的名称。

2. 或者，使用 Openshift Web 控制台打开受影响的监控 pod 的日志。有关可能原因的更多信息，会在日志中反映。

3. 执行常规 pod 故障排除步骤：

Pod 状态：Pending

4. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
oc get pod | grep {ceph-component}
```

5. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a {ceph-component}  that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod_name>
指定被认为是有问题的 pod 的 pod 的名称。

6. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

如果分配了节点，请检查节点上的 kubelet。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.11. CephMonQuorumAtRisk

含义	多个 MON 协同工作以提供冗余性。每个 MON 都会保留元数据的副本。集群使用 3 MON 部署，并且需要 2 个或更多 MON 上线并运行仲裁，以及运行存储操作。如果仲裁丢失，对数据的访问将面临风险。

影响	高

诊断

恢复 Ceph MON Quorum。如需更多信息，请参阅故障排除指南中的 *在 OpenShift Data Foundation 中恢复 ceph-monitor 仲裁*。https://access.redhat.com/documentation/zh-cn/red_hat_openshift_data_foundation/4.14/html-single/troubleshooting_openshift_data_foundation/index#restoring-ceph-monitor-quorum-in-openshift-data-foundation_rhodf如果恢复 Ceph MON Quorum 失败，请遵循常规 pod 故障排除来解决这个问题。

对常规 pod 故障排除执行以下操作：

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
oc get pod | grep rook-ceph-mon
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a {ceph-component}  that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod_name>
指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

如果分配了节点，请检查节点上的 kubelet。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.12. CephMonQuorumLost

含义	在 Ceph 集群中，有一组冗余的 monitor pod，用于存储有关存储集群的重要信息。定期监控 pod 同步，以获取有关存储集群的信息。第一个监控 pod 获取最新更新的信息成为领导信息，其他监控容器集将在询问领导后启动其同步过程。在一个或多个 monitor pod 中网络连接或其他类型问题会造成不必要的变化。这种情形可能会对存储集群性能造成负面影响。
影响	高

重要

检查是否有网络问题。如果存在网络问题，则需要在进行以下任何故障排除步骤前升级到 OpenShift Data Foundation 团队。

诊断

恢复 Ceph MON Quorum。如需更多信息，请参阅故障排除指南中的 *在 OpenShift Data Foundation 中恢复 ceph-monitor 仲裁*。https://access.redhat.com/documentation/zh-cn/red_hat_openshift_data_foundation/4.14/html-single/troubleshooting_openshift_data_foundation/index#restoring-ceph-monitor-quorum-in-openshift-data-foundation_rhodf如果恢复 Ceph MON Quorum 失败，请遵循常规 pod 故障排除来解决这个问题。

或者，执行常规 pod 故障排除：

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
oc get pod | grep {ceph-component}
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a {ceph-component}  that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod_name>

指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

-

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

如果分配了节点，请检查节点上的 kubelet。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.13. CephMonVersionMismatch

含义	通常，这个警报会在升级过程中触发，这需要很长时间。
影响	Medium

诊断

检查 **ocs-operator** 订阅状态和 Operator pod 健康状况，以检查 Operator 升级是否正在进行。

1. 检查 **ocs-operator** 订阅健康状况。

```
$ oc get sub $(oc get pods -n openshift-storage | grep -v ocs-operator) -n openshift-storage -o json | jq .status.conditions
```

状态条件类型是 **CatalogSourcesUnhealthy**,**InstallPlanMissing**,**InstallPlanPending**, 和 **InstallPlanFailed**。每种类型的状态应当是 **False**。

输出示例：

```
[
  {
    "lastTransitionTime": "2021-01-26T19:21:37Z",
```

```
    "message": "all available catalogsources are healthy",
    "reason": "AllCatalogSourcesHealthy",
    "status": "False",
    "type": "CatalogSourcesUnhealthy"
  }
]
```

示例输出显示 **CatalogSourcesUnHealthly** 类型的 **False** 状态，这意味着目录源处于健康状态。

2. 检查 OCS operator pod 状态，以查看正在进行中的 OCS operator 是否升级。

```
$ oc get pod -n openshift-storage | grep ocs-operator OCSOP=$(oc get pod -n openshift-storage -o custom-columns=POD:.metadata.name --no-headers | grep ocs-operator) echo $OCSOP oc get pod/${OCSOP} -n openshift-storage oc describe pod/${OCSOP} -n openshift-storage
```

如果您确定"ocs-operator"正在进行中，请等待 5 分钟，并且此警报应自行解决。如果您等待或看到不同的错误状态条件，请继续故障排除。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

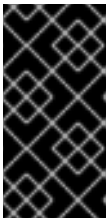
6.3.14. CephNodeDown

含义	运行 Ceph pod 的节点已停机。虽然存储操作将继续工作，因为 Ceph 旨在处理节点故障，但建议解决这个问题，以最大程度降低另一个节点停机并影响存储功能的风险。
影响	Medium

诊断

1. 列出运行和失败的所有 pod：

```
oc -n openshift-storage get pods
```



重要

确保您满足 OpenShift Data Foundation 资源要求，以便将对象存储设备(OSD) pod 调度到新节点上。这可能需要几分钟时间，因为 Ceph 集群恢复故障的数据，但现在恢复 OSD。要监视此恢复，请确保 OSD pod 正确放置到新的 worker 节点上。

2. 检查之前失败的 OSD pod 是否现在是否正在运行：

```
oc -n openshift-storage get pods
```

如果之前出现故障的 OSD pod 没有调度，请使用 **describe** 命令并检查事件，因为 pod 无法重新调度的原因。

3. 描述故障 OSD pod 的事件：

```
oc -n openshift-storage get pods | grep osd
```

4. 查找一个或多个失败的 OSD pod：

```
oc -n openshift-storage describe pods/<osd_podname_ from_the_ previous step>
```

在 events 部分中，查找失败的原因，如没有满足资源。

另外，您可以使用 **rook-ceph-toolbox** 来监控恢复。此步骤是可选的，但对大型 Ceph 集群非常有用。要访问 toolbox，请运行以下命令：

```
TOOLS_POD=$(oc get pods -n openshift-storage -l app=rook-ceph-tools -o name)
oc rsh -n openshift-storage $TOOLS_POD
```

在 rsh 命令提示符中运行以下命令，并在 io 部分监视 "recovery"：

```
ceph status
```

5. 确定是否有失败的节点。

   a. 获取 worker 节点列表，并检查节点状态：

   ```
   oc get nodes --selector='node-role.kubernetes.io/worker','!node-role.kubernetes.io/infra'
   ```

   b. 描述处于 **NotReady** 状态的节点以获取有关故障的更多信息：

   ```
   oc describe node <node_name>
   ```

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

  ```
  $ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
  ```

6.3.15. CephOSDCriticallyFull

含义	一个对象存储设备(OSD)的可以空间几乎已用完。立即扩展集群。
影响	高

诊断

删除数据以释放存储空间

您可以删除数据，集群将通过自我修复过程来解析警报。

重要

这仅适用于接近或完全完满的 OpenShift Data Foundation 集群，它们不适用于只读模式。只读模式可防止任何包含删除数据的更改，即删除持久性卷声明(PVC)、持久性卷(PV)或两者。

扩展存储容量

当前存储大小小于 1 TB

您必须首先评估扩展的功能。对于添加的每个 1TB 存储，集群需要有 3 个节点，每个节点都有最少可用 2 个 vCPU 和 8 GiB 内存。

您可以通过附加组件将存储容量增加到 4 TB，集群将通过自我修复过程解决警报。如果没有满足最低 vCPU 和内存资源要求，则需要在集群中添加 3 个额外的 worker 节点。

缓解方案

- 如果您的当前存储大小等于 4 TB，请联系红帽支持。
- 可选：运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.16. CephOSDDiskNotResponding

含义	磁盘设备没有响应。检查所有对象存储设备(OSD)是否已启动并运行。
影响	Medium

诊断

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
$ oc get pod | grep rook-ceph
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a rook-ceph that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod_name>

指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

- 如果分配了节点，请检查节点上的 kubelet。
- 如果正在运行的 pod 的基本健康状况，则验证节点上的节点关联性和资源可用性，请运行 Ceph 工具来获取存储组件的状态。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.17. CephOSDDiskUnavailable

含义	一个主机上无法访问磁盘设备，其对应的对象存储设备(OSD)被 Ceph 集群标记为 out。当 Ceph 节点无法在 10 分钟内恢复时，会引发此警报。
影响	高

诊断

确定失败的节点

1. 获取 worker 节点列表，并检查节点状态：

```
oc get nodes --selector='node-role.kubernetes.io/worker','!node-role.kubernetes.io/infra'
```

1. 描述处于 **NotReady** 状态的节点以获取有关故障的更多信息：

```
oc describe node <node_name>
```

6.3.18. CephOSDFlapping

含义	在最后 5 分钟内，存储守护进程已重启 5 次。检查 pod 事件或 Ceph 状态，以查找原因。
影响	高

诊断

按照 Red Hat Ceph Storage 故障排除指南中的 Flapping OSD 部分的步骤进行操作。

或者，按照常规 pod 故障排除的步骤进行操作：

Pod 状态：Pending

1. 检查资源问题、待处理持久性卷声明(PVC)、节点分配和 kubelet 问题：

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

```
$ oc get pod | grep rook-ceph
```

2. 将 **MYPOD** 设置为标识为问题 pod 的 pod 的变量：

```
# Examine the output for a rook-ceph that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod_name>
指定被认为是有问题的 pod 的 pod 的名称。

3. 查找资源限制或待处理的 PVC。否则，检查节点分配：

```
$ oc get pod/${MYPOD} -o wide
```

Pod 状态： NOT pending, running, but not ready

- 检查就绪度探测：

```
$ oc describe pod/${MYPOD}
```

pod 状态: NOT pending, but NOT running

- 检查应用程序或镜像问题：

```
$ oc logs pod/${MYPOD}
```

重要

- 如果分配了节点，请检查节点上的 kubelet。
- 如果正在运行的 pod 的基本健康状况，则验证节点上的节点关联性和资源可用性，请运行 Ceph 工具来获取存储组件的状态。

缓解方案

调试日志信息

- 此步骤是可选的。运行以下命令来收集 Ceph 集群的调试信息：

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14
```

6.3.19. CephOSDNearFull

含义	后端存储设备对象存储设备(OSD)的利用率在主机上已超过 75%。
影响	高

缓解方案

释放集群中的一些空间、扩展存储集群或联系红帽支持。如需有关扩展存储的更多信息，请参阅扩展存储指南。

6.3.20. CephOSDSlowOps

含义	请求速度较慢的 OSD 是每个无法在 **osd_op_complaint_time** 参数定义的时间内队列中每秒 I/O 操作(IOPS)服务的 OSD。默认情况下，此参数被设置为 30 秒。
影响	Medium

诊断

有关使用 Openshift 控制台获取有关较慢请求的更多信息。

1. 访问 OSD pod 终端，并运行以下命令：

```
$ ceph daemon osd.<id> ops
```

```
$ ceph daemon osd.<id> dump_historic_ops
```

注意

OSD 的数量在容器集名称中看到。例如，在 **rook-ceph-osd-0-5d86d4d8d4-zlqkx** 中，**<0>** 是 OSD。

缓解方案

OSD 请求缓慢的主要原因包括：

- 底层硬件或基础架构的问题，如磁盘驱动器、主机、机架或网络交换机。使用 Openshift 监控控制台查找集群资源的警报或错误。这可让您了解 OSD 中缓慢操作的根本原因。
- 与网络相关的问题。这些问题通常与 flapping OSD 相关。请参阅 Red Hat Ceph Storage 故障排除指南中的 Flapping OSD 部分

- 如果是网络问题，请升级到 OpenShift Data Foundation 团队
- 系统负载。使用 Openshift 控制台查看 OSD pod 的指标以及运行 OSD 的节点。添加或分配更多资源可以是可能的解决方案。

6.3.21. CephOSDVersionMismatch

含义	通常，这个警报会在升级过程中触发，这需要很长时间。
影响	Medium

诊断

检查 **ocs-operator** 订阅状态和 Operator pod 健康状况，以检查 Operator 升级是否正在进行。

1. 检查 **ocs-operator** 订阅健康状况。

```
$ oc get sub $(oc get pods -n openshift-storage | grep -v ocs-operator) -n openshift-storage -o json | jq .status.conditions
```

状态条件类型是 **CatalogSourcesUnhealthy**,**InstallPlanMissing**,**InstallPlanPending**, 和 **InstallPlanFailed**。每种类型的状态应当是 **False**。

输出示例：

```
[
  {
    "lastTransitionTime": "2021-01-26T19:21:37Z",
    "message": "all available catalogsources are healthy",
    "reason": "AllCatalogSourcesHealthy",
    "status": "False",
    "type": "CatalogSourcesUnhealthy"
  }
]
```

示例输出显示 **CatalogSourcesUnHealthly** 类型的 **False** 状态，这意味着目录源处于健康状态。

2. 检查 OCS operator pod 状态，以查看正在进行中的 OCS operator 是否升级。

```
$ oc get pod -n openshift-storage | grep ocs-operator OCSOP=$(oc get pod -n openshift-storage -o custom-columns=POD:.metadata.name --no-headers | grep ocs-operator) echo $OCSOP oc get pod/${OCSOP} -n openshift-storage oc describe pod/${OCSOP} -n openshift-storage
```

如果您确定"ocs-operator"正在进行中，请等待 5 分钟，并且此警报应自行解决。如果您等待或看到不同的错误状态条件，请继续故障排除。

6.3.22. CephPGRepairTakingTooLong

含义	自我修复操作用时过长。

影响	高

诊断

检查放置组(PG)是否不一致，并修复它们。有关更多信息，请参阅红帽知识库解决方案 Handle Inconsistent Placement Groups in Ceph。

6.3.23. CephPoolQuotaBytesCriticallyExhausted

含义	已达到一个或多个池，或者非常接近到达其配额。触发此错误条件的阈值由 **mon_pool_quota_crit_threshold** 配置选项控制。
影响	高

缓解方案

调整池配额。运行以下命令以完全删除或调整池配额：

```
ceph osd pool set-quota <pool> max_bytes <bytes>
```

```
ceph osd pool set-quota <pool> max_objects <objects>
```

将配额值设置为 **0** 将禁用配额。

6.3.24. CephPoolQuotaBytesNearExhaustion

含义	一个或多个池正在接近配置的全度阈值。触发此警告条件的一个阈值是 **mon_pool_quota_warn_threshold** 配置选项。
影响	高

缓解方案

调整池配额。运行以下命令以完全删除或调整池配额：

```
ceph osd pool set-quota <pool> max_bytes <bytes>
```

```
ceph osd pool set-quota <pool> max_objects <objects>
```

将配额值设置为 **0** 将禁用配额。

6.3.25. PersistentVolumeUsageCritical

含义	持久性卷声明(PVC)接近其完整容量，如果未及时处理，可能会导致数据丢失。
影响	高

缓解方案

扩展 PVC 大小以增加容量。

1. 登录 OpenShift Web 控制台。
2. 点 **Storage → PersistentVolumeClaim**。
3. 从 **Project** 下拉列表中选择 **openshift-storage**。
4. 在您要扩展的 PVC 中，点 **Action menu (⋮) → Expand PVC**。
5. 将**总大小**更新为所需的大小。
6. 点 **Expand**。

或者，您可以删除可能会占用空间的不必要的数据。

6.3.26. PersistentVolumeUsageNearFull

含义	持久性卷声明(PVC)接近其完整容量，如果未及时处理，可能会导致数据丢失。
影响	高

缓解方案

扩展 PVC 大小以增加容量。

1. 登录 OpenShift Web 控制台。
2. 点 **Storage → PersistentVolumeClaim**。
3. 从 **Project** 下拉列表中选择 **openshift-storage**。
4. 在您要扩展的 PVC 中，点 **Action menu (⋮) → Expand PVC**。
5. 将**总大小**更新为所需的大小。
6. 点 **Expand**。

或者，您可以删除可能会占用空间的不必要的数据。

6.4. 解决 NOOBAA BUCKET 错误状态

流程

1. 在 OpenShift Web 控制台中，点 **Storage → Data Foundation**。
2. 在 **Overview** 选项卡的 **Status** 卡中，点 **Storage System**，然后点弹出框中的存储系统链接。
3. 单击 **Object** 选项卡。
4. 在 **Details** 卡中，点 **System Name** 字段下的链接。

5. 在左侧窗格中，点 **Buckets** 选项并搜索处于错误状态的存储桶。如果处于错误状态的存储桶是一个命名空间存储桶，请确定点 **Namespace Buckets** 窗格。
6. 点其 **Bucket Name**。此时会显示存储桶中遇到的错误。
7. 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**。

6.5. 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. 在 OpenShift Web 控制台中，点 **Storage → Data Foundation**。
  2. 在 **Overview** 选项卡的 **Status** 卡中，点 **Storage System**，然后点弹出框中的存储系统链接。
  3. 单击 **Object** 选项卡。
  4. 在 **Details** 卡中，点 **System Name** 字段下的链接。
  5. 在左侧窗格中，点 **Buckets** 选项并搜索处于错误状态的存储桶。
  6. 点其 **Bucket Name**。此时会显示存储桶中遇到的错误。
  7. 点 **Bucket Policies → Edit Quota** 并增加配额。

6.6. 解决 NOOBAA BUCKET CAPACITY 或 QUOTA STATE 问题

流程

1. 在 OpenShift Web 控制台中，点 **Storage → Data Foundation**。
2. 在 **Overview** 选项卡的 **Status** 卡中，点 **Storage System**，然后点弹出框中的存储系统链接。
3. 单击 **Object** 选项卡。

4. 在 **Details** 卡中，点 **System Name** 字段下的链接。
5. 在左侧窗格中，点 **Resources** 选项，再搜索 PV 池资源。
6. 对于具有低容量状态的 PV 池资源，请单击 **Resource Name**。
7. 编辑池配置并增加代理数量。

6.7. 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：

流程

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
```

6.8. 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。

6.9. 为 ROOK-CEPH-OPERATOR 启用和禁用 DEBUG 日志

为 rook-ceph-operator 启用 debug 日志，以获取有助于对问题进行故障排除的失败信息。

流程

启用 debug 日志

1. 编辑 rook-ceph-operator 的 configmap。

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

2. 在 **rook-ceph-operator-config** yaml 文件中添加 **ROOK_LOG_LEVEL: DEBUG** 参数，为 rook-ceph-operator 启用调试日志。

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

现在，rook-ceph-operator 日志由 debug 信息组成。

禁用 debug 日志

1. 编辑 rook-ceph-operator 的 configmap。

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

2. 在 **rook-ceph-operator-config** yaml 文件中添加 **ROOK_LOG_LEVEL: INFO** 参数，以禁用 rook-ceph-operator 的调试日志。

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

6.10. 解决具有五个或更多节点的部署的 CEPH 监控器计数

当部署中存在三个、五个或更多个故障域（基于机架或区域的数量）时，您可以在内部模式部署中配置五个 Ceph 监控器计数。您可以将值设为三个或五个。此配置有助于通过配置 Ceph 监控器计数来提高集群的可用性。

流程

1. 在 OpenShift Web 控制台的通知面板或 Alert Center 中，会显示一个警报来指示 monitor 计数为不满时的 Ceph 监控器计数。
2. 在 **Inadequate Ceph Monitor count** 警报中，单击 **Configure**。
3. 在 **Configure Ceph Monitor** 弹出窗口中，单击 **Update** count。
   在弹出窗口中，显示推荐的 monitor 数量，具体取决于故障区域的数量。
4. 在 **Configure CephMon** 弹出窗口中，根据推荐的值更新 monitor count 值，然后单击 **Save changes**。

6.11. 不健康的阻塞节点故障排除

6.11.1. ODFRBDClientBlocked

含义	此警报表示 Ceph 在 Kubernetes 集群的特定节点上可能会阻止 RADOS 块设备(RBD)客户端。当 **ocs_rbd_client_blocklisted metric** 为节点报告了 1 时，将发生阻止行为。另外，在同一节点上存在 **CreateContainerError** 状态的 pod。阻塞列表可能会导致使用 RBD 的持久性卷声明 (PVC) 的文件系统变为只读。调查此警报非常重要，以防止对存储集群造成任何中断。
影响	高

诊断

由于多个因素（如网络或集群速度较慢）可能会出现 RBD 客户端阻止列表。在某些情况下，三个持续客户端（工作负载、镜像守护进程和 manager/scheduler）之间的专用锁争用可能会导致 blocklist。

缓解方案

1. 为被放入阻塞列表的节点添加污点：在 Kubernetes 中，请考虑污点节点，以触发 pod 驱除到另一节点。这个方法假设卸载/取消映射过程正常进行。pod 成功被驱除后，可能会取消阻塞节点，允许清除 blocklist。然后可将 pod 移到未包含的节点。

2. 重启列入阻塞列表的节点：如果污点节点并驱除 pod 没有解决阻塞的问题，则可以尝试重启列入阻塞列表的节点。此步骤可能帮助缓解导致 blocklist 并恢复正常功能的任何底层问题。

重要

及时调查并解决 blocklist 问题对于避免对存储集群有进一步影响至关重要。

第 7 章 检查 LOCAL STORAGE OPERATOR 部署

使用本地存储 Operator 的 Red Hat OpenShift Data Foundation 集群是使用本地存储设备部署的。要查找您的 OpenShift Data Foundation 的现有集群是否使用本地存储设备进行了部署，请使用以下步骤：

先决条件

- OpenShift Data Foundation 在 **openshift-storage** 命名空间上安装并运行。

流程

通过检查与 OpenShift Data Foundation 集群的持久性卷声明(PVC)关联的存储类，您可以确定您的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 Data Foundation 集群 PVC 关联的存储类：

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

2. 检查输出。对于带有 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
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
```

其它资源

- 使用 VMware 上的本地存储设备部署 OpenShift Data Foundation
- 使用 Red Hat Virtualization 上的本地存储设备部署 OpenShift Data Foundation
- 使用裸机上的本地存储设备部署 OpenShift Data Foundation
- 使用 IBM Power 上的本地存储设备部署 OpenShift Data Foundation

第 8 章 删除失败或不需要的 CEPH 对象存储设备

失败或不需要的 Ceph OSD （对象存储设备）会影响存储基础架构的性能。因此，为了提高存储集群的可靠性和弹性，您必须删除失败或不需要的 Ceph OSD。

如果您有故障或不需要的 Ceph OSD 来删除：

1. 验证 Ceph 健康状态。
   有关更多信息，请参阅：验证 Ceph 集群是否健康。
2. 根据 OSD 的调配，移除失败或不需要的 Ceph OSD。
   请参阅：
   - 在 动态置备的 Red Hat OpenShift Data Foundation 中删除失败的或不需要的 Ceph OSD 。
   - 使用本地存储设备移除失败的或不需要的 Ceph OSD。

如果使用本地磁盘，您可以在删除旧 OSD 后重复使用这些磁盘。

8.1. 验证 CEPH 集群是否健康

存储健康状况在 **Block** 和 **File** 和 **Object** 仪表板上可见。

流程

1. 在 OpenShift Web 控制台中，点 **Storage → Data Foundation**。
2. 在 **Overview** 选项卡的 **Status** 卡中，点 **Storage System**，然后点弹出框中的存储系统链接。
3. 在 **Block and File** 选项卡的 **Status** 卡中，验证 *Storage Cluster* 是否具有绿色勾号。
4. 在 **Details** 卡中，验证是否显示集群信息。

8.2. 在动态置备的 RED HAT OPENSHIFT DATA FOUNDATION 中删除失败的或不需要的 CEPH OSD

按照流程中的步骤，在动态置备的 Red Hat OpenShift Data Foundation 中删除失败或不需要的 Ceph 对象存储设备(OSD)。

重要

只有红帽支持团队才支持缩减集群。

警告

- 当 Ceph 组件没有处于健康状态时，删除 OSD 可能会导致数据丢失。
- 同时删除两个或多个 OSD 会导致数据丢失。

先决条件

- 检查 Ceph 是否健康。如需更多信息，请参阅验证 Ceph 集群是否健康。
- 确保没有触发警报，或者所有重建过程都在进行中。

流程

1. 缩减 OSD 部署。

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

2. 获取要删除的 Ceph OSD 的 **osd-prepare** pod。

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

3. 删除 **osd-prepare** pod。

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

4. 从集群移除出现故障的 OSD。

```
# failed_osd_id=<osd-id>

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

其中，**FAILED_OSD_ID** 是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

5. 通过检查日志来验证 OSD 是否已成功移除。

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

6. 可选：如果您遇到 **cephosd:osd.0 is not ok to destroy to destroy to destroy** from the **ocs-osd-removal-job** pod in OpenShift Container Platform 的错误，请参阅 对 **cephosd:osd.0** 错误进行故障排除，同时删除失败或不需要的 Ceph OSD。

7. 删除 OSD 部署。

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

验证步骤

- 要检查 OSD 是否已成功删除，请运行：

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

此命令必须将状态返回为 **Completed**。

8.3. 使用本地存储设备移除失败的或不需要的 CEPH OSD

您可以按照以下步骤使用本地存储设备删除失败或不需要的 Ceph 置备的对象存储设备(OSD)。

重要

只有红帽支持团队才支持缩减集群。

警告

- 当 Ceph 组件没有处于健康状态时，删除 OSD 可能会导致数据丢失。
- 同时删除两个或多个 OSD 会导致数据丢失。

先决条件

- 检查 Ceph 是否健康。如需更多信息，请参阅验证 Ceph 集群是否健康。
- 确保没有触发警报，或者所有重建过程都在进行中。

流程

1. 总之，通过将 OSD 部署上的副本扩展到 0 来标记 OSD 停机。如果 OSD 已因为失败而停机，您可以跳过这一步。

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

2. 从集群移除出现故障的 OSD。

```
# failed_osd_id=<osd_id>

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

其中，**FAILED_OSD_ID** 是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

3. 通过检查日志来验证 OSD 是否已成功移除。

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

4. 可选：如果您遇到 **cephosd:osd.0 is not ok to destroy to destroy to destroy** from the **ocs-osd-removal-job** pod in OpenShift Container Platform 的错误，请参阅 对 **cephosd:osd.0** 错误进行故障排除，同时删除失败或不需要的 Ceph OSD。

5. 删除与故障 OSD 关联的持久性卷声明(PVC)资源。

   a. 获取与故障 OSD 关联的 **PVC**。

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

   b. 获取与 **PVC 关联的持久性卷** (PV)。

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

c. 获取失败的设备名称。

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

d. 获取与故障 OSD 关联的 **prepare-pod**。

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

e. 在删除关联的 PVC 前，删除 **osd-prepare pod**。

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

f. 删除与故障 OSD 关联的 **PVC**。

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

6. 从 **LocalVolume 自定义资源** (CR)中删除失败的设备条目。

a. 使用失败设备登录到节点。

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

b. 为失败的设备名称记录 /dev/disk/by-id/<id>。

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

7. 可选：如果是，Local Storage Operator 用于置备 OSD，使用 {osd-id} 登录机器并删除设备符号链接。

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

a. 获取故障设备名称的 OSD 符号链接。

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

b. 删除 符号链接。

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

8. 删除与 OSD 关联的 PV。

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

验证步骤

- 要检查 OSD 是否已成功删除，请运行：

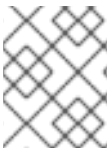
```
#oc get pod -n openshift-storage ocs-osd-removal-$<failed_osd_id>-<pod-suffix>
```

此命令必须将状态返回为 **Completed**。

8.4. 对 CEPHOSD:OSD.0 错误进行故障排除，在删除失败或不需要的 CEPH OSD 时无法销毁

如果您收到 **cephosd:osd.0 is not ok to destroy to destroy** from the **ocs-osd-removal-job** pod in OpenShift Container Platform 的错误，请使用 **FORCE_OSD_REMOVAL** 选项运行 Object Storage Device (OSD)删除作业，将 OSD 移到 destroyed 状态。

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



注意

只有在所有 PG 都处于 active 状态时，才必须使用 **FORCE_OSD_REMOVAL** 选项。如果没有，PG 必须完成回填或进一步调查，以确保它们处于活动状态。

第 9 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。如果没有这样做，资源仍然处于 **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 -n <namespace>
```

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的对象类型。查看以上输出中的消息。
示例：

message: Some content in the namespace has finalizers remaining: cephobjectstoreuser.ceph.rook.io

此处 **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}}'
```

Example:

```
$ 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 10 章 对外部模式的 CEPHFS PVC 创建进行故障排除

如果您已将 Red Hat Ceph Storage 集群从低于 4.1.1 的版本更新为最新版本，且不是全新的集群，则必须在 Red Hat Ceph Storage 集群上为 CephFS 池手动设置应用程序类型，以便在外部模式中启用 CephFS 持久性卷声明(PVC)创建。

1. 检查 CephFS pvc 处于 **Pending** 状态。

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

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 **oc describe** 命令的输出以查看相应 pvc 的事件。
预期的错误消息为 **cephfs_metadata/csi.volumes.default/csi.volume.pvc-xxxxxx-xxxx-xxxx-xxxx-xxxxxxx:(1)Operation not permitted）**

```
# 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-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. 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 -n <namespace>
```

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 11 章 在 OPENSHIFT DATA FOUNDATION 中恢复 MONITOR POD

如果所有三个 Pod 都停机，并且 OpenShift Data Foundation 无法自动恢复 monitor pod，则恢复 monitor pod。

注意

这是一个灾难恢复过程，必须在红帽支持团队的指导下执行。请联系红帽支持团队。

流程

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. 在 **openshift-storage** 命名空间中创建所有部署的备份。

```
# 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. 修补对象存储设备(OSD)部署以删除 **livenessProbe** 参数，并使用命令参数作为 **sleep** 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/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. 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/pod\///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** 之前复制密钥环模板文件。

```
# 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. 从对应的机密中识别所有其他 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 "
[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"
```

重要

- 对于 **client.csi** 相关密钥环，请参阅前面的密钥环文件输出，并在从相应的 OpenShift Data Foundation 机密获取密钥后添加默认 **caps**。
- OSD 密钥环会在恢复后自动添加。

10. 导航到 **mon-a** 容器集，再验证 **monstore** 是否具有 **monmap**。

    a. 进入到 **mon-a** 容器集。

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

    b. 验证 **monstore** 是否具有 **monmap**。

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

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

11. 可选：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>

    mon-a pod 的 ID。

    <mon-a-ip>

    mon-a pod 的 IP 地址。

    <mon-b-id>

    mon-b pod 的 ID。

    <mon-b-ip>

    mon-b pod 的 IP 地址。

    <mon-c-id>

    mon-c pod 的 ID。

    <mon-c-ip>

    mon-c pod 的 IP 地址。

    <fsid>

文件系统 ID。

12. 验证 **monmap**。

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

13. 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
```

14. 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
```

15. 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
```

16. 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

17. 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

18. 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。

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

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

验证步骤

1. 检查 Ceph 状态，以确认 CephFS 正在运行。

```
# ceph -s
```

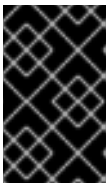
输出示例：

```
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
```

2. 检查 Multicloud 对象网关 (MCG) 状态。它应该处于活动状态，后备存储和存储桶类应处于 **Ready** 状态。

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



重要

如果 MCG 没有处于 active 状态，且后备存储和存储桶类没有处于 **Ready** 状态，则需要重启所有与 MCG 相关的 pod。如需更多信息，请参阅 第 11.1 节 “恢复 Multicloud 对象网关”。

11.1. 恢复 MULTICLOUD 对象网关

如果 Multicloud 对象网关(MCG)没有处于 active 状态，且后备存储和存储桶类没有处于 **Ready** 状态，您需要重启所有与 MCG 相关的 pod，并检查 MCG 状态以确认 MCG 是否已备份并在运行。

流程

1. 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

2. 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

注意

在 OpenShift Container Platform 4.11 中，在恢复后 RBD PVC 无法挂载到应用程序 pod 上。因此，您需要重启托管应用容器集的节点。要获取托管应用程序 pod 的节点名称，请运行以下命令：

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

第 12 章 在 OPENSHIFT DATA FOUNDATION 中恢复 CEPH-MONITOR 仲裁

在某些情况下，**ceph-mons** 可能会丢失仲裁。如果 **mons** 无法再次形成仲裁，则需要一个手动过程来再次进入仲裁。唯一的要求是至少一个 **mon** 必须健康。以下步骤从仲裁中删除不健康状态的 **mons**，并可让您使用单个 **mon** 重新组成仲裁，然后将仲裁回到原始大小。

例如，如果您有三个 **mons** 并失去了仲裁，您需要从仲裁中删除两个有问题的 **mons**，通知可以正常工作的 **mon** 它是仲裁中唯一的 **mon**，然后重启这个可以正常工作的 **mon**。

流程

1. 停止 **rook-ceph-operator**，以便在修改 **monmap** 时不通过 **mons** 失败。

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

2. 注入一个新的 **monmap**。

警告

您必须非常仔细注入 **monmap**。如果运行不正确，您的集群可以被永久销毁。Ceph **monmap** 来跟踪 **mon** 仲裁。**monmap** 被更新为仅包含健康的 mon。在本例中，健康的 mon 是 **rook-ceph-mon-b**，而不健康的 **mons** 为 **rook-ceph-mon-a** 和 **rook-ceph-mon-c**。

a. 备份当前的 **rook-ceph-mon-b** 部署：

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

b. 打开 YAML 文件，并从 **mon** 容器复制**命令**和**参数**（请参见以下示例中的容器列表）。这是 **monmap** 更改所需要的。

```
[...]
  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. 清理复制的 **command** 和 **args** 字段以形成过去的命令，如下所示：

```
# 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
```

注意

确保删除括起了 **--log-stderr-prefix** 标记的单引号，以及包括 **ROOK_CEPH_MON_MON_MON_MON_HOST**、**ROOK_CEPH_MON_MON_CEPH_MON_INITIAL_MEMBERS** 和 **ROOK_POD_IP** 变量的括号。

d. 修补 **rook-ceph-mon-b** 部署，在不删除 **mon** 的情况下停止这个 **mon** 工作。

```
# 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. 在 **mon-b** pod 上执行以下步骤：

i. 连接到健康 **mon** 的 pod 并运行以下命令：

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

ii. 设置变量。

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

iii. 将 **monmap** 提取到一个文件，从健康的 **mon** 部署中粘贴 ceph **mon** 命令并添加 **--extract-monmap=${monmap_path}** 标记。

```
# 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. 检查 **monmap** 的内容。

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

v. 从 **monmap** 中删除错误的 **mons**。

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

在本例中，我们移除了 **mon0** 和 **mon2** ：

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

vi. 把修改过的 **monmap** 注入到健康的 **mon**中，粘贴 ceph **mon** 命令并添加 **--inject-monmap=${monmap_path}** 标记：

```
# 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. 退出 shell 以继续。

3. 编辑 Rook **configmaps**。

a. 编辑 operator 用来跟踪 **mons** 的 **configmap**。

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

b. 验证在数据元素中，您可以看到如下三个 **mon** （具体取决于您的 **moncount**，可能会更多）：

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

c. 从列表中删除有问题的 **mons**，以使用一个好的 **mon** 结束。例如：

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

d. 保存文件并退出。

e. 现在，您需要使用用于 **mons** 和其他组件的 **Secret**。

i. 为变量 **good_mon_id** 设置一个值。
例如：

```
# good_mon_id=b
```

ii. 您可以使用 **oc patch** 命令来修补 **rook-ceph-config** secret，并更新两个键/值对 **mon_host** 和 **mon_initial_members**。

```
# 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}"""}}'
```

注意

如果使用 **hostNetwork: true**，则需要将 **mon_host** 变量替换为代表 **mon** 固定到的节点 IP(**nodeSelector**)。这是因为在那个 "mode" 中创建了 **rook-ceph-mon-*** 服务。

4. 重新启动 **mon**。
您需要使用原始 **ceph-mon** 命令重启好的 **mon** pod，以获取这些更改。

a. 在 **mon** 部署 YAML 文件的备份中使用 **oc replace** 命令：

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

注意

选项 **--force** 删除部署并创建新部署。

b. 验证集群的状态。
状态应该在仲裁中显示 **mon**。如果状态正常，您的集群应该再次处于健康状态。

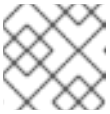
5. 删除不再预期在仲裁中的两个 mon 部署。
例如：

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

在本例中，要删除的部署有 **rook-ceph-mon-a** 和 **rook-ceph-mon-c**。

6. 重启 Operator。

a. 再次启动 rook 运算符，以恢复监控集群的健康状况。



注意

忽略多个资源已存在的错误是安全的。

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

根据 **mon** 数量，Operator 会自动添加更多 **mons** 来再次增加仲裁大小。

第 13 章 启用 RED HAT OPENSHIFT DATA FOUNDATION 控制台插件

Data Foundation 控制台插件默认启用。如果在 OpenShift Data Foundation Operator 安装过程中取消选择这个选项，请按照以下说明从图形用户界面(GUI)或命令行界面启用控制台插件。

先决条件

- 您有管理访问权限来访问 OpenShift Web 控制台。
- OpenShift Data Foundation Operator 在 **openshift-storage** 命名空间上安装并运行。

流程

从用户界面

1. 在 OpenShift Web 控制台中，点 **Operators → Installed Operators** 查看所有已安装的 Operator。
2. 确保所选 **项目** 为 **openshift-storage**。
3. 单击 **OpenShift Data Foundation** operator。
4. 启用 console 插件选项。
   a. 在 **Details** 选项卡中，单击 **Console 插件** 下 **的铅笔图标。**
   b. 选择 **Enable**，然后单击 **Save**。

使用命令行界面

- 执行以下命令启用 console 插件选项：

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

验证步骤

- 启用 console 插件选项后，显示一条带有消息的弹出窗口，**Web 控制台更新**会出现在 GUI 中。点这个弹出窗口中的 **Refresh web console** 来反映控制台的更改。
  - 在 Web 控制台中，导航到 **Storage** 并验证 **Data Foundation** 是否可用。

第 14 章 更改 OPENSHIFT DATA FOUNDATION 组件的资源

安装 OpenShift Data Foundation 时，它附带了 OpenShift Data Foundation Pod 可消耗的预定义资源。在某些情况下，可能需要提高 I/O 负载。

- 要更改 rook-ceph pod 上的 CPU 和内存资源，请参阅 第 14.1 节 “更改 rook-ceph pod 上的 CPU 和内存资源”。
- 要调整 Multicloud 对象网关(MCG)的资源，请参阅 第 14.2 节 “为 MCG 调整资源”。

14.1. 更改 ROOK-CEPH POD 上的 CPU 和内存资源

安装 OpenShift Data Foundation 时，它附带了 rook-ceph Pod 的预定义 CPU 和内存资源。您可以根据要求手动增加这些值。

您可以更改以下 pod 中的 CPU 和内存资源：

- **mgr**
- **mds**
- **rgw**

以下示例演示了如何更改 rook-ceph Pod 上的 CPU 和内存资源。在本例中，**cpu** 和 **memory** 的现有 MDS pod 值会分别从 **1** 和 **4Gi** 增加到 **2** 和 **8Gi**。

1. 编辑存储集群：

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

<storagecluster_name>
指定存储集群的名称。
例如：

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

2. 将下面几行添加到存储集群自定义资源(CR)中：

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

3. 保存更改并退出编辑器。
4. 或者，运行 **oc patch** 命令更改 **mds** pod 的 CPU 和内存值：

```
# 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>

指定存储集群的名称。
例如：

```
# 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. 为 MCG 调整资源

Multicloud 对象网关(MCG)的默认配置针对低资源消耗和不性能进行了优化。有关如何调整 MCG 资源的更多信息，请参阅用于多云对象网关(NooBaa)的红帽知识库解决方案性能调整指南。

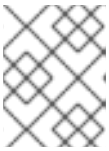
第 15 章 部署 OPENSHIFT DATA FOUNDATION 后禁用多云对象网关外部服务

部署 OpenShift Data Foundation 时，即使 OpenShift 作为私有集群安装，也会创建公共 IP。但是，您可以使用 storagecluster CRD 中的 **disableLoadBalancerService** 变量禁用多云对象网关(MCG)负载均衡器的使用。这限制 MCG 为私有集群创建任何公共资源，并有助于禁用 NooBaa 服务 **EXTERNAL-IP**。

流程

- 运行以下命令，并在 storagecluster YAML 中添加 **disableLoadBalancerService** 变量，将服务设置为 ClusterIP ：

```
$ oc edit storagecluster -n openshift-storage <storagecluster_name>
[...]
spec:
  arbiter: {}
  encryption:
    kms: {}
  externalStorage: {}
  managedResources:
    cephBlockPools: {}
    cephCluster: {}
    cephConfig: {}
    cephDashboard: {}
    cephFilesystems: {}
    cephNonResilientPools: {}
    cephObjectStoreUsers: {}
    cephObjectStores: {}
    cephRBDMirror: {}
    cephToolbox: {}
  mirroring: {}
  multiCloudGateway:
    disableLoadBalancerService: true       <---------- Add this
    endpoints:
[...]
```



注意

要撤销更改并将服务设置为 LoadBalancer，请将 **disableLoadBalancerService** 变量设置为 **false** 或完全删除该行。

第 16 章 使用 OVS-MULTITENANT 插件访问 ODF-CONSOLE，方法是手动启用全局 POD 网络

在 OpenShift Container Platform 中，当 **ovs-multitenant** 插件用于软件定义型网络 (SDN) 时，来自不同项目的 pod 无法将数据包发送到不同项目的 pod 和服务的数据包。默认情况下，pod 无法在命名空间或项目之间进行通信，因为项目的 pod 网络不是全局的。

要访问 odf-console，**openshift-console** 命名空间中的 OpenShift 控制台 pod 需要与 **openshift-storage** 命名空间中的 OpenShift Data Foundation odf-console 连接。这只有在手动启用全局 pod 网络时才可能。

问题

- 当 OpenShift Container Platform 中使用 `ovs-multitenant` 插件时，odf-console 插件会失败，并显示以下信息：

  > 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)

解决方案

- 使 OpenShift Data Foundation 项目的 pod 网络全局化：

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

第 17 章 注解加密的 RBD 存储类

从 OpenShift Data Foundation 4.14 开始，当 OpenShift 控制台创建启用加密的 RADOS 块设备(RBD)存储类时，会自动设置注解。但是，您需要在升级到 OpenShift Data Foundation 版本 4.14 之前为之前创建的任何加密的 RBD 存储类添加注解 **cdi.kubevirt.io/clone-strategy=copy**。这可让客户数据集成 (CDI) 使用主机辅助克隆，而不是默认的智能克隆。

用于访问加密卷的密钥与创建卷的命名空间相关联。当将加密卷克隆到新命名空间时，如置备新的 OpenShift Virtualization 虚拟机时，必须创建一个新卷，然后源卷的内容必须复制到新卷中。如果正确注解存储类，则会自动触发此行为。