Red Hat Ceph Storage 7

故障排除指南

Red Hat Ceph Storage 故障排除

Last Updated: 2024-06-14

Red Hat Ceph Storage 7 故障排除指南

Red Hat Ceph Storage 故障排除

法律通告

摘要

本文档论述了如何解决 Red Hat Ceph Storage 中的常见问题。 红帽承诺替换我们的代码、文档和网页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 信息。

目录

第 1 章 初始故障排除

作为存储管理员，您可以在联系红帽支持前对 Red Hat Ceph Storage 集群进行初始故障排除。本章包括以下信息：

- *识别问题。*
- *诊断存储集群的健康状况.*
- *了解 Ceph 健康.*
- *静默 Ceph 集群的健康警报。*
- *了解 Ceph 日志.*
- *生成 'sos report'。*

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

1.1. 识别问题

要确定 Red Hat Ceph Storage 集群的错误可能的原因，请回答流程部分中的问题。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

流程

1. 使用不支持的配置时可能会出现某些问题。确保您的配置被支持。
2. 您知道哪个 Ceph 组件导致了此问题吗？
   a. No. 参照 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Diagnosing the health of a Ceph storage cluster* 过程。
   b. Ceph 监控器。参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Troubleshooting Ceph Monitors* 部分。
   c. Ceph OSD。请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Troubleshooting Ceph OSDs* 部分。
   d. Ceph 放置组。请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Troubleshooting Ceph placement groups* 部分。
   e. 多站点 Ceph 对象网关.请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Troubleshooting a multi-site Ceph Object Gateway* 部分。

其它资源

- 详情请参阅 Red Hat Ceph Storage: 支持的配置 文章。

1.2. 诊断存储集群的健康状况

此流程列出了诊断 Red Hat Ceph Storage 集群健康状况的基本步骤。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

流程

1. 登录到 Cephadm shell：

   示例

   ```
   [root@host01 ~]# cephadm shell
   ```

2. 检查存储集群的整体状态：

   示例

   ```
   [ceph: root@host01 /]# ceph health detail
   ```

   如果命令返回 **HEALTH_WARN** 或 **HEALTH_ERR**，请参阅 *了解 Ceph 健康状况* 以了解详细信息。

3. 监控存储集群的日志：

   示例

   ```
   [ceph: root@host01 /]# ceph -W cephadm
   ```

4. 要将集群日志捕获到文件中，请运行以下命令：

   示例

   ```
   [ceph: root@host01 /]# ceph config set global log_to_file true
   [ceph: root@host01 /]# ceph config set global mon_cluster_log_to_file true
   ```

   日志默认位于 **/var/log/ceph/*CLUSTER_FSID*/** 目录中。检查 Ceph 日志中的 *了解 Ceph 日志* 中列出的任何错误消息。

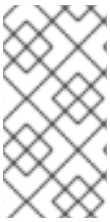
5. 如果日志没有包括足够数量的信息，请提高调试级别，并尝试重现失败的操作。详情请参阅 *配置日志记录*。

1.3. 了解 CEPH 的健康状态

ceph health 命令返回有关 Red Hat Ceph Storage 集群状态的信息：

- **HEALTH_OK** 表示集群处于健康状态。
- **HEALTH_WARN** 表示警告。在某些情况下，Ceph 状态会自动返回到 **HEALTH_OK**。例如，当 Red Hat Ceph Storage 集群完成重新平衡过程时。但是，如果集群处于 **HEALTH_WARN** 状态的时间较长，请考虑进一步的故障排除。
- **HEALTH_ERR** 表示更严重的问题，需要您立即关注的问题。

使用 **ceph health detail** 和 **ceph -s** 命令获取更详细的输出。



注意

如果没有 **mgr** 守护进程正在运行，则会显示运行状况警告。如果删除了 Red Hat Storage 集群的最后一个 **mgr** 守护进程，您可以在 Red Hat Ceph Storage 集群随机主机上手动部署 **mgr** 守护进程。请参阅 *Red Hat Ceph Storage 7 管理指南中的 手动部署 mgr 守护进程*。

其它资源

- 请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Ceph Monitor error messages* 表。
- 请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Ceph OSD error messages* 表。
- 请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Placement group error messages* 表。

1.4. 静默 CEPH 集群的健康警报

在某些情况下，用户可能希望临时静默一些警告，因为它们已经了解警告且无法立即操作。您可以静默健康检查，以便它们不会影响 Ceph 集群的总体报告状态。

使用健康检查代码指定警报。例如，当 OSD 停机进行维护时，**OSD_DOWN** 警告是正常的。您可以选择在维护结束前静默警告，因为在维护期间，这些警告会使集群处于 **HEALTH_WARN** 状态而不是 **HEALTH_OK** 状态。

如果一个警报的状态变差，则大多数健康状况也会消失。例如，如果有一个 OSD down，并且警报被静默，如果一个或多个额外的 OSD 停机，则静默会消失。对于任何涉及指示触发警告或错误的对象数量或数量的健康警报，这都为 true。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 对节点的根级别的访问权限。
- 健康警告信息。

流程

1. 登录到 Cephadm shell：

   示例

   ```
   [root@host01 ~]# cephadm shell
   ```

2. 运行 **ceph health detail** 命令检查 Red Hat Ceph Storage 集群的健康状态：

   示例

   ```
   [ceph: root@host01 /]# ceph health detail

   HEALTH_WARN 1 osds down; 1 OSDs or CRUSH {nodes, device-classes} have
   ```

```
{NOUP,NODOWN,NOIN,NOOUT} flags set
[WRN] OSD_DOWN: 1 osds down
    osd.1 (root=default,host=host01) is down
[WRN] OSD_FLAGS: 1 OSDs or CRUSH {nodes, device-classes} have
{NOUP,NODOWN,NOIN,NOOUT} flags set
    osd.1 has flags noup
```

您可以看到存储集群处于 **HEALTH_WARN** 状态，因为其中一个 OSD 为 down。

3. 静默警报：

语法

```
ceph health mute HEALTH_MESSAGE
```

示例

```
[ceph: root@host01 /]# ceph health mute OSD_DOWN
```

4. 可选：健康检查可以有一个与它关联的生存时间(TTL)，以便在指定时间段内自动过期。在命令中使用可选的 TTL 作为指定持续时间的参数：

语法

```
ceph health mute HEALTH_MESSAGE DURATION
```

DURATION 可以在 **s**、**sec**、**m**、**min**、**h** 或 **hour** 中指定。

示例

```
[ceph: root@host01 /]# ceph health mute OSD_DOWN 10m
```

在本例中，警报 **OSD_DOWN** 为 muted 达到 10 分钟。

5. 验证 Red Hat Ceph Storage 集群状态是否已更改为 **HEALTH_OK** ：

示例

```
[ceph: root@host01 /]# ceph -s
  cluster:
    id:     81a4597a-b711-11eb-8cb8-001a4a000740
    health: HEALTH_OK
            (muted: OSD_DOWN(9m) OSD_FLAGS(9m))

  services:
    mon: 3 daemons, quorum host01,host02,host03 (age 33h)
    mgr: host01.pzhfuh(active, since 33h), standbys: host02.wsnngf, host03.xwzphg
    osd: 11 osds: 10 up (since 4m), 11 in (since 5d)

  data:
    pools:   1 pools, 1 pgs
```

```
    objects: 13 objects, 0 B
    usage:   85 MiB used, 165 GiB / 165 GiB avail
    pgs:     1 active+clean
```

在本例中，您可以看到警报 *OSD_DOWN* 和 *OSD_FLAG* 处于静默状态，静默会持续 9 分钟。

6. 可选：您可以通过使其 **粘滞** 在清除警报后保留静默。

   语法

   ```
   ceph health mute HEALTH_MESSAGE DURATION --sticky
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph health mute OSD_DOWN 1h --sticky
   ```

7. 您可以运行以下命令来删除静默：

   语法

   ```
   ceph health unmute HEALTH_MESSAGE
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph health unmute OSD_DOWN
   ```

其它资源

- 详情请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Health messages of a Ceph cluster* 部分。

1.5. 了解 CEPH 日志

在启用了日志记录到文件后，Ceph 将其日志存储在 **/var/log/ceph/*CLUSTER_FSID*/** 目录中。

***CLUSTER_NAME*.log** 是包含全局事件的主存储集群日志文件。默认情况下，日志文件名称为 **ceph.log**。只有 Ceph Monitor 节点会包含主要的存储集群日志。

每个 Ceph OSD 和 monitor 都有自己的日志文件，名为 ***CLUSTER_NAME*-osd.*NUMBER*.log** 和 ***CLUSTER_NAME*-mon.*HOSTNAME*.log**。

当您提高 Ceph 子系统的调试级别时，Ceph 也为这些子系统生成新的日志文件。

其它资源

- 有关日志记录的详细信息，请参阅 *Red Hat Ceph Storage 故障排除指南* 中的*配置日志记录*。
- 请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Common Ceph Monitor error messages in the Ceph logs* 表中。
- 请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Common Ceph OSD error messages in the Ceph logs* 表中。

- 请参阅 *Ceph 守护进程日志*，以启用日志记录到文件。

1.6. 生成 SOS 报告

您可以运行 **sos report** 命令，从 Red Hat Enterprise Linux 收集 Red Hat Ceph Storage 集群的配置详情、系统信息和诊断信息。红帽支持团队使用此信息进一步排除存储集群的问题。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 对节点的根级别访问权限。

流程

1. 安装 **sos** 软件包：

   示例

   ```
   [root@host01 ~]# dnf install sos
   ```

2. 运行 **sos 报告** 以获取存储集群的系统信息：

   示例

   ```
   [root@host01 ~]# sosreport -a --all-logs
   ```

   该报告保存在 **/var/tmp** 文件中。

   对于特定的 Ceph 守护进程信息，运行以下命令：

   示例

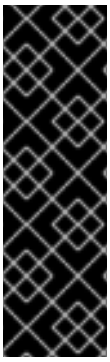
   ```
   [root@host01 ~]# sos report --all-logs -e
   ceph_mgr,ceph_common,ceph_mon,ceph_osd,ceph_ansible,ceph_mds,ceph_rgw
   ```

其它资源

- 请参阅 *sosreport 是什么以及如何在 Red Hat Enterprise Linux 中创建？* 如需更多信息，知识库文章。

第 2 章 配置日志记录

本章论述了如何为各种 Ceph 子系统配置日志记录。



重要

日志记录是资源密集型的。另外，详细日志记录可以在相对短时间内生成大量数据。如果您在集群的特定子系统中遇到问题，请只启用该子系统的日志。请参阅 第 2.1 节 “Ceph 子系统” 了解更多信息。

另外，请考虑设置日志文件轮转。详情请查看 第 2.4 节 “加快日志轮转”。

在遇到的所有问题都被修复后，将子系统日志和内存级别更改为默认值。有关所有 Ceph 子系统及其默认值的列表，请参阅 附录 A, *Ceph 子系统默认日志记录级别值*。

您可以通过以下方法配置 Ceph 日志记录：

- 在运行时使用 **ceph** 命令。这是最常见的方法。详情请查看 第 2.2 节 “在运行时配置日志记录”。
- 更新 Ceph 配置文件。如果您在启动集群时遇到问题，请使用这种方法。详情请查看 第 2.3 节 “在配置文件中配置日志”。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

2.1. CEPH 子系统

本节包含有关 Ceph 子系统及其日志级别的信息。

了解 Ceph 子系统及其日志记录级别

Ceph 由多个子系统组成。

每个子系统都有一个日志级别：

- 输出日志默认存储在 **/var/log/ceph/*CLUSTER_FSID*/** 目录（日志级别）
- 存储在内存缓存中的日志（内存级别）

通常，Ceph 不会将内存中存储的日志发送到输出日志，除非：

- 引发致命信号
- 触发源代码中的 assert
- 您请求它

您可以为每个子系统设置不同的值。Ceph 日志级别在 **1** 到 **20** 范围内运行，其中 **1** 最简洁，**20** 最详细。

对日志级别和内存级别使用单个值，以将它们都设置为相同的值。例如，**debug_osd = 5** 将 **ceph-osd** 守护进程的 debug 级别设置为 **5**。

要将不同的值用于输出日志级别和内存级别，请使用正斜杠(/)分隔它们的值。例如，**debug_mon = 1/5** 将 **ceph-mon** 守护进程的 debug 日志级别设置为 **1**，并将其内存日志级别设置为 **5**。

表 2.1. Ceph 子系统和日志记录默认值

子系统	日志级别	内存级别	描述
asok	1	5	管理套接字
auth	1	5	身份验证
client	0	5	任何使用 **librados** 连接到集群的应用或库
bluestore	1	5	BlueStore OSD 后端
journal	1	5	OSD 日志
mds	1	5	元数据服务器
monc	0	5	monitor 客户端处理大多数 Ceph 守护进程和 monitor 之间的通信
mon	1	5	Monitors
ms	0	5	Ceph 组件之间的消息传递系统
osd	0	5	OSD 守护进程
paxos	0	5	监控用来建立共识的算法
rados	0	5	可靠的自主分布式对象存储，这是 Ceph 的核心组件
rbd	0	5	Ceph 块设备
rgw	1	5	Ceph 对象网关

日志输出示例

下例演示了当您提高 monitor 和 OSD 的详细程度时，日志中的消息类型。

监控调试设置

```
debug_ms = 5
debug_mon = 20
debug_paxos = 20
debug_auth = 20
```

Monitor Debug Settings 的日志输出示例

```
2022-05-12 12:37:04.278761 7f45a9afc700 10 mon.cephn2@0(leader).osd e322 e322: 2 osds: 2 up,
```

```
2 in
2022-05-12 12:37:04.278792 7f45a9afc700 10 mon.cephn2@0(leader).osd e322
min_last_epoch_clean 322
2022-05-12 12:37:04.278795 7f45a9afc700 10 mon.cephn2@0(leader).log v1010106 log
2022-05-12 12:37:04.278799 7f45a9afc700 10 mon.cephn2@0(leader).auth v2877 auth
2022-05-12 12:37:04.278811 7f45a9afc700 20 mon.cephn2@0(leader) e1 sync_trim_providers
2022-05-12 12:37:09.278914 7f45a9afc700 11 mon.cephn2@0(leader) e1 tick
2022-05-12 12:37:09.278949 7f45a9afc700 10 mon.cephn2@0(leader).pg v8126 v8126: 64 pgs: 64
active+clean; 60168 kB data, 172 MB used, 20285 MB / 20457 MB avail
2022-05-12 12:37:09.278975 7f45a9afc700 10 mon.cephn2@0(leader).paxosservice(pgmap
7511..8126) maybe_trim trim_to 7626 would only trim 115 < paxos_service_trim_min 250
2022-05-12 12:37:09.278982 7f45a9afc700 10 mon.cephn2@0(leader).osd e322 e322: 2 osds: 2 up,
2 in
2022-05-12 12:37:09.278989 7f45a9afc700  5 mon.cephn2@0(leader).paxos(paxos active c
1028850..1029466) is_readable = 1 - now=2021-08-12 12:37:09.278990 lease_expire=0.000000 has
v0 lc 1029466
....
2022-05-12 12:59:18.769963 7f45a92fb700  1 -- 192.168.0.112:6789/0 <== osd.1
192.168.0.114:6800/2801 5724 ==== pg_stats(0 pgs tid 3045 v 0) v1 ==== 124+0+0 (2380105412 0
0) 0x5d96300 con 0x4d5bf40
2022-05-12 12:59:18.770053 7f45a92fb700  1 -- 192.168.0.112:6789/0 --> 192.168.0.114:6800/2801
-- pg_stats_ack(0 pgs tid 3045) v1 -- ?+0 0x550ae00 con 0x4d5bf40
2022-05-12 12:59:32.916397 7f45a9afc700  0 mon.cephn2@0(leader).data_health(1) update_stats
avail 53% total 1951 MB, used 780 MB, avail 1053 MB
....
2022-05-12 13:01:05.256263 7f45a92fb700  1 -- 192.168.0.112:6789/0 --> 192.168.0.113:6800/2410
-- mon_subscribe_ack(300s) v1 -- ?+0 0x4f283c0 con 0x4d5b440
```

OSD 调试设置

```
debug_ms = 5
debug_osd = 20
```

OSD 调试设置的日志输出示例

```
2022-05-12 11:27:53.869151 7f5d55d84700  1 -- 192.168.17.3:0/2410 --> 192.168.17.4:6801/2801 --
osd_ping(ping e322 stamp 2021-08-12 11:27:53.869147) v2 -- ?+0 0x63baa00 con 0x578dee0
2022-05-12 11:27:53.869214 7f5d55d84700  1 -- 192.168.17.3:0/2410 --> 192.168.0.114:6801/2801
-- osd_ping(ping e322 stamp 2021-08-12 11:27:53.869147) v2 -- ?+0 0x638f200 con 0x578e040
2022-05-12 11:27:53.870215 7f5d6359f700  1 -- 192.168.17.3:0/2410 <== osd.1
192.168.0.114:6801/2801 109210 ==== osd_ping(ping_reply e322 stamp 2021-08-12
11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x63c1a00 con 0x578e040
2022-05-12 11:27:53.870698 7f5d6359f700  1 -- 192.168.17.3:0/2410 <== osd.1
192.168.17.4:6801/2801 109210 ==== osd_ping(ping_reply e322 stamp 2021-08-12
11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x6313200 con 0x578dee0
....
2022-05-12 11:28:10.432313 7f5d6e71f700  5 osd.0 322 tick
2022-05-12 11:28:10.432375 7f5d6e71f700 20 osd.0 322 scrub_random_backoff lost coin flip,
randomly backing off
2022-05-12 11:28:10.432381 7f5d6e71f700 10 osd.0 322 do_waiters -- start
2022-05-12 11:28:10.432383 7f5d6e71f700 10 osd.0 322 do_waiters -- finish
```

其它资源

- *在运行时配置日志记录*
- *在配置文件中配置日志*

2.2. 在运行时配置日志记录

您可以在系统运行时配置 Ceph 子系统日志记录，以帮助对可能出现的任何问题进行故障排除。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 可以访问 Ceph 调试器。

流程

1. 要在运行时激活 Ceph 调试输出 **dout ()** ：

```
ceph tell TYPE.ID injectargs --debug-SUBSYSTEM VALUE [--NAME VALUE]
```

2. 替换：
   - ***TYPE***，类型为 Ceph 守护进程(**osd**、**mon** 或 **mds**)
   - 使用 Ceph 守护进程的特定 ***ID*** 的 ID。或者，使用 * 将运行时设置应用到特定类型的所有守护进程。
   - ***SUBSYSTEM*** 带有一个特定的子系统。
   - ***VALUE*** 为介于 **1** 到 **20** 之间的一个值，其中 **1** 为 terse，**20** 为详细。
     例如，要将名为 **osd.0** 的 OSD 上的 OSD 子系统的日志级别设为 0，内存级别设为 5：

```
# ceph tell osd.0 injectargs --debug-osd 0/5
```

在运行时查看配置设置：

1. 使用正在运行的 Ceph 守护进程登录主机，如 **ceph-osd** 或 **ceph-mon**。
2. 显示配置：

   语法

```
ceph daemon NAME config show | less
```

   示例

```
[ceph: root@host01 /]# ceph daemon osd.0 config show | less
```

其它资源

- 详情请参阅 *Ceph 子系统*。
- *详情请参阅配置文件中的配置日志记录*。

- Red Hat Ceph Storage 7 的 *Configuration Guide* 中的 *Ceph Debugging and Logging Configuration Reference* 章节

2.3. 在配置文件中配置日志

配置 Ceph 子系统，以将信息、警告和错误消息记录到日志文件。您可以在 Ceph 配置文件中指定调试级别，默认为 **/etc/ceph/ceph.conf**。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

流程

1. 要在启动时激活 Ceph 调试输出，**dout()** 在启动时添加调试设置到 Ceph 配置文件。
   a. 对于各个守护进程通用的子系统，请在 **[global]** 部分下添加设置。
   b. 对于特定守护进程的子系统，在守护进程部分中添加设置，如 **[mon]**、**[osd]** 或 **[mds]**。

   示例

```
[global]
        debug_ms = 1/5

[mon]
        debug_mon = 20
        debug_paxos = 1/5
        debug_auth = 2

[osd]
        debug_osd = 1/5
        debug_monc = 5/20

[mds]
        debug_mds = 1
```

其它资源

- *Ceph 子系统*
- *在运行时配置日志记录*
- Red Hat Ceph Storage 7 的 *Configuration Guide* 中的 *Ceph Debugging and Logging Configuration Reference* 章节

2.4. 加快日志轮转

为 Ceph 组件增加调试级别可能会导致大量数据。如果您几乎已满磁盘，可以通过修改 **/etc/logrotate.d/ceph-<fsid> 中的 Ceph 日志轮转文件** 来加快日志轮转。Cron Job 调度程序使用此文件来调度日志轮转。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 节点的根级别访问权限。

流程

1. 在轮转频率后向日志轮转文件中添加大小设置：

```
rotate 7
weekly
size SIZE
compress
sharedscripts
```

例如，在日志文件达到 500 MB 时轮转日志文件：

```
rotate 7
weekly
size 500 MB
compress
sharedscripts
size 500M
```

注意

大小值可以表示为 '500 MB' 或 '500M'。

2. 打开 **crontab** 编辑器：

```
[root@mon ~]# crontab -e
```

3. 添加一个条目来检查 **/etc/logrotate.d/ceph-<fsid>** 文件。例如，指示 Cron 每 30 分钟检查 **/etc/logrotate.d/ceph-<fsid** > ：

```
30 * * * * /usr/sbin/logrotate /etc/logrotate.d/ceph-d3bb5396-c404-11ee-9e65-002590fc2a2e
>/dev/null 2>&1
```

2.5. 为 CEPH 对象网关创建和收集操作日志

用户身份信息添加到操作日志输出中。这用于使用户访问此信息以审核 S3 访问。跟踪 S3 请求在所有 Ceph 对象网关操作日志中可靠的用户身份。

流程

1. 查找日志所在的位置：

语法

```
logrotate -f
```

示例

```
[root@host01 ~]# logrotate -f
/etc/logrotate.d/ceph-12ab345c-1a2b-11ed-b736-fa163e4f6220
```

2. 列出指定位置的日志：

语法

```
ll LOG_LOCATION
```

示例

```
[root@host01 ~]# ll /var/log/ceph/12ab345c-1a2b-11ed-b736-fa163e4f6220
 -rw-r--r--. 1 ceph ceph    412 Sep 28 09:26 opslog.log.1.gz
```

3. 列出当前的存储桶：

示例

```
[root@host01 ~]# /usr/local/bin/s3cmd ls
```

4. 创建存储桶：

语法

```
/usr/local/bin/s3cmd mb s3://NEW_BUCKET_NAME
```

示例

```
[root@host01 ~]# /usr/local/bin/s3cmd mb s3://bucket1
Bucket `s3://bucket1` created
```

5. 列出当前的日志：

语法

```
ll LOG_LOCATION
```

示例

```
[root@host01 ~]# ll /var/log/ceph/12ab345c-1a2b-11ed-b736-fa163e4f6220
 total 852
 ...
 -rw-r--r--. 1 ceph ceph    920 Jun 29 02:17 opslog.log
 -rw-r--r--. 1 ceph ceph    412 Jun 28 09:26 opslog.log.1.gz
```

6. 收集日志：

语法

```
tail -f LOG_LOCATION/opslog.log
```

示例

```
[root@host01 ~]# tail -f /var/log/ceph/12ab345c-1a2b-11ed-b736-fa163e4f6220/opslog.log

{"bucket":"","time":"2022-09-29T06:17:03.133488Z","time_local":"2022-09-
29T06:17:03.133488+0000","remote_addr":"10.0.211.66","user":"test1",
"operation":"list_buckets","uri":"GET /
HTTP/1.1","http_status":"200","error_code":"","bytes_sent":232,
"bytes_received":0,"object_size":0,"total_time":9,"user_agent":"","referrer":
"","trans_id":"tx00000c80881a9acd2952a-006335385f-175e5-primary",
"authentication_type":"Local","access_key_id":"1234","temp_url":false}

{"bucket":"cn1","time":"2022-09-29T06:17:10.521156Z","time_local":"2022-09-
29T06:17:10.521156+0000","remote_addr":"10.0.211.66","user":"test1",
"operation":"create_bucket","uri":"PUT /cn1/
HTTP/1.1","http_status":"200","error_code":"","bytes_sent":0,
"bytes_received":0,"object_size":0,"total_time":106,"user_agent":"",
"referrer":"","trans_id":"tx0000058d60c593632c017-0063353866-175e5-primary",
"authentication_type":"Local","access_key_id":"1234","temp_url":false}
```

第 3 章 网络问题故障排除

本章列出了与网络以及网络时间协议(NTP)连接的基本故障排除步骤。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

3.1. 基本网络故障排除

Red Hat Ceph Storage 依赖于可靠的网络连接。Red Hat Ceph Storage 节点使用网络相互通信。网络问题可能会导致 Ceph OSD 的许多问题，如它们流化，或者被错误地报告为 **down**。网络问题也可以导致 Ceph Monitor 的时钟偏移错误。另外，数据包丢失、高延迟或有限的带宽可能会影响集群性能和稳定性。

先决条件

- 节点的根级别访问权限。

流程

1. 安装 **net-tools** 和 **telnet** 软件包有助于对 Ceph 存储集群中可能出现的网络问题进行故障排除：

   示例

   ```
   [root@host01 ~]# dnf install net-tools
   [root@host01 ~]# dnf install telnet
   ```

2. 登录 **cephadm** shell，再验证 Ceph 配置文件中的 **public_network** 参数是否包含正确的值：

   示例

   ```
   [ceph: root@host01 /]# cat /etc/ceph/ceph.conf
   # minimal ceph.conf for 57bddb48-ee04-11eb-9962-001a4a000672
   [global]
    fsid = 57bddb48-ee04-11eb-9962-001a4a000672
    mon_host = [v2:10.74.249.26:3300/0,v1:10.74.249.26:6789/0]
   [v2:10.74.249.163:3300/0,v1:10.74.249.163:6789/0]
   [v2:10.74.254.129:3300/0,v1:10.74.254.129:6789/0]
   [mon.host01]
   public network = 10.74.248.0/21
   ```

3. 退出 shell 并验证网络接口是否已启动：

   示例

   ```
   [root@host01 ~]# ip link list
   1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
   DEFAULT group default qlen 1000
       link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
   DEFAULT group default qlen 1000
       link/ether 00:1a:4a:00:06:72 brd ff:ff:ff:ff:ff:ff
   ```

4. 使用它们的短主机名，验证 Ceph 节点是否能够互相访问。在存储集群的每个节点上验证它：

   语法

   ```
   ping SHORT_HOST_NAME
   ```

   示例

   ```
   [root@host01 ~]# ping host02
   ```

5. 如果使用防火墙，请确保 Ceph 节点能够在适当的端口上相互访问。**firewall-cmd** 和 **telnet** 工具可验证端口状态，以及端口是否分别打开：

   语法

   ```
   firewall-cmd --info-zone=ZONE
   telnet IP_ADDRESS PORT
   ```

   示例

   ```
   [root@host01 ~]# firewall-cmd --info-zone=public
   public (active)
     target: default
     icmp-block-inversion: no
     interfaces: ens3
     sources:
     services: ceph ceph-mon cockpit dhcpv6-client ssh
     ports: 9283/tcp 8443/tcp 9093/tcp 9094/tcp 3000/tcp 9100/tcp 9095/tcp
     protocols:
     masquerade: no
     forward-ports:
     source-ports:
     icmp-blocks:
     rich rules:

   [root@host01 ~]# telnet 192.168.0.22 9100
   ```

6. 验证接口计数器上没有错误。验证节点之间的网络连接是否有预期的延迟，并且没有数据包丢失。

   a. 使用 **ethtool** 命令：

      语法

      ```
      ethtool -S INTERFACE
      ```

      示例

      ```
      [root@host01 ~]# ethtool -S ens3 | grep errors
      NIC statistics:
          rx_fcs_errors: 0
          rx_align_errors: 0
          rx_frame_too_long_errors: 0
      ```

```
    rx_in_length_errors: 0
    rx_out_length_errors: 0
    tx_mac_errors: 0
    tx_carrier_sense_errors: 0
    tx_errors: 0
    rx_errors: 0
```

b. 使用 **ifconfig** 命令：

示例

```
[root@host01 ~]# ifconfig
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.74.249.26  netmask 255.255.248.0  broadcast 10.74.255.255
        inet6 fe80::21a:4aff:fe00:672  prefixlen 64  scopeid 0x20<link>
        inet6 2620:52:0:4af8:21a:4aff:fe00:672  prefixlen 64  scopeid 0x0<global>
        ether 00:1a:4a:00:06:72  txqueuelen 1000  (Ethernet)
        RX packets 150549316  bytes 56759897541 (52.8 GiB)
        RX errors 0  dropped 176924  overruns 0  frame 0
        TX packets 55584046  bytes 62111365424 (57.8 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9373290  bytes 16044697815 (14.9 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9373290  bytes 16044697815 (14.9 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

c. 使用 **netstat** 命令：

示例

```
[root@host01 ~]# netstat -ai
Kernel Interface table
Iface             MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-
OVR Flg
ens3             1500 311847720      0 364903 0      114341918      0      0      0 BMRU
lo              65536 19577001       0      0 0       19577001       0      0      0 LRU
```

7. 对于性能问题，除了延迟检查和验证存储集群所有节点之间的网络带宽外，使用 **iperf3** 工具。**iperf3** 工具在服务器和客户端之间执行一个简单的点对点网络带宽测试。

a. 在您要检查带宽的 Red Hat Ceph Storage 节点上安装 **iperf3** 软件包：

示例

```
[root@host01 ~]# dnf install iperf3
```

b. 在 Red Hat Ceph Storage 节点上，启动 **iperf3** 服务器：

示例

```
[root@host01 ~]# iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

注意

默认端口为 5201，但可使用 **-P** 命令参数进行设置。

c. 在不同的 Red Hat Ceph Storage 节点上，启动 **iperf3** 客户端：

示例

```
[root@host02 ~]# iperf3 -c mon
Connecting to host mon, port 5201
[  4] local xx.x.xxx.xx port 52270 connected to xx.x.xxx.xx port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec   114 MBytes   954 Mbits/sec    0    409 KBytes
[  4]   1.00-2.00   sec   113 MBytes   945 Mbits/sec    0    409 KBytes
[  4]   2.00-3.00   sec   112 MBytes   943 Mbits/sec    0    454 KBytes
[  4]   3.00-4.00   sec   112 MBytes   941 Mbits/sec    0    471 KBytes
[  4]   4.00-5.00   sec   112 MBytes   940 Mbits/sec    0    471 KBytes
[  4]   5.00-6.00   sec   113 MBytes   945 Mbits/sec    0    471 KBytes
[  4]   6.00-7.00   sec   112 MBytes   937 Mbits/sec    0    488 KBytes
[  4]   7.00-8.00   sec   113 MBytes   947 Mbits/sec    0    520 KBytes
[  4]   8.00-9.00   sec   112 MBytes   939 Mbits/sec    0    520 KBytes
[  4]   9.00-10.00  sec   112 MBytes   939 Mbits/sec    0    520 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  1.10 GBytes   943 Mbits/sec    0             sender
[  4]   0.00-10.00  sec  1.10 GBytes   941 Mbits/sec                  receiver

iperf Done.
```

此输出显示 Red Hat Ceph Storage 节点之间 1.1 Gbits/秒的网络带宽，在测试过程中不会重新传输(**Retr**)。

红帽建议您验证存储集群中所有节点之间的网络带宽。

8. 确保所有节点具有相同的网络互连速度。连接较慢的节点可能会减慢连接速度更快的节点。另外，确保交换机链接可以处理附加节点的聚合带宽：

语法

```
ethtool INTERFACE
```

示例

```
[root@host01 ~]# ethtool ens3
Settings for ens3:
Supported ports: [ TP ]
Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
```

```
                    1000baseT/Half 1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Supported FEC modes: Not reported
Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full
Advertised pause frame use: Symmetric
Advertised auto-negotiation: Yes
Advertised FEC modes: Not reported
Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                     100baseT/Half 100baseT/Full
                                     1000baseT/Full
Link partner advertised pause frame use: Symmetric
Link partner advertised auto-negotiation: Yes
Link partner advertised FEC modes: Not reported
Speed: 1000Mb/s (1)
Duplex: Full (2)
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: off
Supports Wake-on: g
Wake-on: d
Current message level: 0x000000ff (255)
       drv probe link timer ifdown ifup rx_err tx_err
Link detected: yes (3)
```

其它资源

- 详情请查看客户门户网站中的*基本网络故障排除*解决方案。
- 详情请查看 *什么是"ethtool"命令以及如何使用它来获取有关我的网络设备和接口的信息*。
- 详情请查看 *RHEL 网络接口在客户门户网站中丢弃数据包*解决方案。
- 详情请参阅客户门户网站上的 *Red Hat Ceph Storage 可用的性能基准工具是什么* 。
- 如需更多信息，请参阅客户门户网站中的与网络问题故障排除相关的 *知识库文章和解决方案*。

3.2. 基本 CHRONY NTP 故障排除

本节包括基本的 chrony NTP 故障排除步骤。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- Ceph 监控节点的根级别访问权限.

流程

1. 验证 **chronyd** 守护进程是否在 Ceph Monitor 主机上运行：

示例

```
[root@mon ~]# systemctl status chronyd
```

2. 如果 **chronyd** 没有运行，请启用并启动它：

示例

```
[root@mon ~]# systemctl enable chronyd
[root@mon ~]# systemctl start chronyd
```

3. 确定 **chronyd** 正确同步了时钟：

示例

```
[root@mon ~]# chronyc sources
[root@mon ~]# chronyc sourcestats
[root@mon ~]# chronyc tracking
```

其它资源

- 有关高级 *chrony NTP 故障排除步骤，请参阅红帽客户门户网站中如何排除 chrony 问题*。
- 如需了解更多详细信息，请参阅 *Red Hat Ceph Storage 故障排除指南* 中的*时钟偏移*一节。
- 详情请查看 *Checking if chrony is synchronized* 部分。

第 4 章 CEPH 监控器故障排除

本章介绍了如何修复与 Ceph 监控器相关的最常见的错误。

先决条件

- 验证网络连接。

4.1. 最常见的 CEPH MONITOR 错误

下表列出了 **ceph health detail** 命令返回的最常见错误消息，或者包含在 Ceph 日志中。这些表中提供了相应部分的链接，这些部分解释了错误并指向修复问题的特定程序。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

4.1.1. Ceph 监控错误消息

常见 Ceph Monitor 错误消息表，以及潜在的修复。

错误消息	查看
HEALTH_WARN	
mon.X 已关闭（仲裁外）	Ceph 监控器没有仲裁
clock skew	Clock skew
存储会变得太大！	Ceph 监控器存储太大

4.1.2. Ceph 日志中常见的 Ceph Monitor 错误消息

Ceph 日志中找到的常见 Ceph Monitor 错误消息表，以及到潜在修复的链接。

错误消息	日志文件	查看
clock skew	主集群日志	Clock skew
clocks not synchronized	主集群日志	Clock skew
损坏：记录中出现的错误	监控日志	Ceph 监控器没有仲裁 恢复 Ceph Monitor 存储
损坏：1 缺少文件	监控日志	Ceph 监控器没有仲裁 恢复 Ceph Monitor 存储

错误消息	日志文件	查看
Caught signal (Bus error)	监控日志	Ceph 监控器没有仲裁

4.1.3. Ceph 监控器没有仲裁

一个或多个 Ceph 监控器标记为 **down**，但其他 Ceph 监控器仍然可以组成仲裁。此外， **ceph health detail** 命令返回类似如下的错误消息：

```
HEALTH_WARN 1 mons down, quorum 1,2 mon.b,mon.c
mon.a (rank 0) addr 127.0.0.1:6789/0 is down (out of quorum)
```

这意味着

Ceph 由于各种原因，Ceph 会将 Ceph 标记为 **down**。

如果 **ceph-mon** 守护进程未在运行，它可能具有损坏的存储或者其他一些错误阻止守护进程启动。另外，**/var/** 分区也可能已满。因此，**ceph-mon** 无法对默认位于 **/var/lib/ceph/mon-*SHORT_HOST_NAME*/store.db** 和终止的存储执行任何操作。

如果 **ceph-mon** 守护进程正在运行，但 Ceph 监控器没有仲裁并标记为 **down**，则问题的原因取决于 Ceph Monitor 状态：

- 如果 Ceph 监控器处于 *probing* 状态的时间超过预期，则代表它无法找到其他 Ceph 监控器。此问题可能是由网络问题造成的，或者 Ceph monitor 可能有过时的 Ceph monitor map(**monmap**)，并尝试访问错误的 IP 地址上的其他 Ceph 监控器。或者，如果 **monmap** 是最新的，Ceph Monitor 的时钟可能无法同步。
- 如果 Ceph monitor 处于 *electing* 状态的时间超过预期，Ceph 监控器的时钟可能没有同步。
- 如果 Ceph Monitor 将其状态从 *synchronizing* 变为 *electing* 并返回，则代表集群状态有进展。这意味着，生成新映射的速度比同步过程可以处理的速度快。
- 如果 Ceph 监控器将自身标记为 *leader* 或 *peon*，那么它认为自己处于仲裁状态，而剩余的集群则确定它没有处于这个状态。此问题可能是由时钟同步失败造成的。

要排除此问题，请执行以下操作

1. 验证 **ceph-mon** 守护进程是否正在运行。如果没有，启动它：

   语法

   ```
   systemctl status ceph-FSID@DAEMON_NAME
   systemctl start ceph-FSID@DAEMON_NAME
   ```

   示例

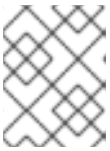
   ```
   [root@mon ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@mon.host01.service
   [root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@mon.host01.service
   ```

2. 如果无法启动 **ceph-mon** 守护进程，请按照 *The **ceph-mon** daemon cannot start* 步骤进行操作。

3. 如果您能够启动 **ceph-mon** 守护进程但标记为 **down**，请按照 *The **ceph-mon** 守护进程正在运行，但标记为"down "*。

ceph-mon 守护进程无法启动

1. 检查位于 **/var/log/ceph/*CLUSTER_FSID*/ceph-mon.*HOST_NAME*.log** 的对应 Ceph Monitor 日志。



注意

默认情况下，日志文件夹中不存在 monitor 日志。您需要启用记录到文件，以便日志出现在 文件夹中。请参阅 *Ceph 守护进程日志*，以启用日志记录到文件。

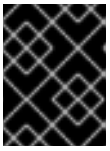
2. 如果日志包含与以下类似的错误消息，Ceph 监控器可能具有损坏的存储：

```
Corruption: error in middle of record
Corruption: 1 missing files; example: /var/lib/ceph/mon/mon.0/store.db/1234567.ldb
```

若要修复此问题，可替换 Ceph Monitor。请参阅 *替换失败的监控器*。

3. 如果日志包含与以下类似的错误消息，**/var/** 分区可能已满。从 **/var/** 中删除任何不必要的数据。

```
Caught signal (Bus error)
```



重要

不要手动删除 Monitor 目录中的任何数据。反之，使用 **ceph-monstore-tool** 来压缩它。详情请参阅 *Compacting the Ceph Monitor 存储*。

4. 如果您看到任何其他错误消息，请创建一个支持问题单。*详情请参阅联系红帽支持以获取服务*。

ceph-mon 守护进程正在运行，但 Still 标记为down

1. 从没有仲裁的 Ceph Monitor 主机中，使用 **mon_status** 命令检查其状态：

```
[root@mon ~]# ceph daemon ID mon_status
```

使用 Ceph Monitor 的 ***ID*** 替换 ID，例如：

```
[ceph: root@host01 /]# ceph daemon mon.host01 mon_status
```

2. 如果状态是 *probing*，请验证 **mon_status** 输出中其他 Ceph monitor 的位置。

   a. 如果地址不正确，Ceph Monitor 具有不正确的 Ceph monitor map (**monmap**)。要解决这个问题，请参阅 *注入 Ceph monitor 映射*。

   b. 如果地址正确，请验证 Ceph Monitor 时钟是否已同步。详情请查看 *Clock skew*。

3. 如果状态选择 ， 请验证 Ceph Monitor 时钟是否已同步。详情请查看 *Clock skew*。

4. 如果状态从 *electing* 变为 *synchronizing*，请创建一个支持问题单。*详情请参阅联系红帽支持以获取服务*。

5. 如果 Ceph Monitor 是 *领导机* 或 *peon*，请验证 Ceph Monitor 时钟是否已同步。详情请查看 *Clock skew*。如果同步时钟无法解决问题，请创建一个支持问题单。*详情请参阅联系红帽支持以获取服务*。

其它资源

- *请参阅了解 Ceph Monitor 状态*
- *Red Hat Ceph Storage Administration Guide* 中的 *Starting, Stopping, Restarting the Ceph daemons* 部分。
- *Red Hat Ceph Storage Administration Guide* 中的 *Using the Ceph Administration Socket* 部分。

4.1.4. Clock skew

Ceph 监控器没有仲裁，**ceph health detail** 命令输出包含类似如下的错误消息：

```
mon.a (rank 0) addr 127.0.0.1:6789/0 is down (out of quorum)
mon.a addr 127.0.0.1:6789/0 clock skew 0.08235s > max 0.05s (latency 0.0045s)
```

另外，Ceph 日志包含类似如下的错误消息：

```
2022-05-04 07:28:32.035795 7f806062e700 0 log [WRN] : mon.a 127.0.0.1:6789/0 clock skew 0.14s
> max 0.05s
2022-05-04 04:31:25.773235 7f4997663700 0 log [WRN] : message from mon.1 was stamped
0.186257s in the future, clocks not synchronized
```

这意味着

clock skew 错误消息表示 Ceph Monitor 的时钟没有同步。时钟同步非常重要，因为 Ceph 监控器依赖于时间精度，并在其时钟未同步时行为不可预测。

mon_clock_drift_allowed 参数决定时钟容许在哪些位置。默认情况下，此参数设置为 0.05 秒。

重要

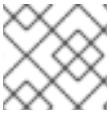
不要在没有之前的测试的情况下更改 **mon_clock_drift_allowed** 的默认值。更改此值可能会影响 Ceph 监控和 Ceph 存储集群的稳定性。

clock skew 错误可能的原因包括 chrony 网络时间协议(NTP)同步的网络问题或问题（如果已配置）。另外，时间同步无法在虚拟机上部署的 Ceph monitor 上正常工作。

要排除此问题，请执行以下操作

1. 验证您的网络是否正常工作。

2. 如果您使用远程 NTP 服务器，请考虑在网络上部署自己的 chrony NTP 服务器。详情请查看红帽客户门户网站中的相关 OS 版本的 Product Documentation for {os-product 中的*配置基本系统设置*指南中的 *使用 Chrony 套件配置 NTP* 章节。



注意

Ceph 仅每五分钟评估时间同步，使得修复问题和清除 **时钟偏移** 消息之间会有一个延迟。

其它资源

- *了解 Ceph Monitor 状态*
- *Ceph 监控器没有仲裁*

4.1.5. Ceph 监控器存储太大

ceph health 命令返回类似如下的错误消息：

```
mon.ceph1 store is getting too big! 48031 MB >= 15360 MB -- 62% avail
```

这意味着

Ceph 监控存储实际上是一个 RocksDB 数据库，它将条目存储为键值对。数据库包含一个集群映射，默认位于 **/var/lib/ceph/*CLUSTER_FSID*/mon.*HOST_NAME*/store.db**。

查询大型 monitor 存储可能需要一些时间。因此，Ceph Monitor 可能会延迟响应客户端查询。

此外，如果 **/var/** 分区已满，Ceph Monitor 无法对存储和终止执行任何写入操作。有关此问题故障排除的详细信息，请参阅 *Ceph Monitor 不在仲裁之外*。

要排除此问题，请执行以下操作

1. 检查数据库的大小：

   语法

   ```
   du -sch /var/lib/ceph/CLUSTER_FSID/mon.HOST_NAME/store.db/
   ```

   指定集群的名称以及运行 **ceph-mon** 的主机的短主机名。

   示例

   ```
   [root@mon ~]# du -sh  /var/lib/ceph/b341e254-b165-11ed-a564-ac1f6bb26e8c/mon.host01/
   109M /var/lib/ceph/b341e254-b165-11ed-a564-ac1f6bb26e8c/mon.host01/
   47G     /var/lib/ceph/mon/ceph-ceph1/store.db/
   47G     total
   ```

2. 紧凑 Ceph Monitor 存储。详情请参阅 编译 *Ceph Monitor 存储*。

其它资源

- Ceph 监控器没有仲裁

4.1.6. 了解 Ceph Monitor 状态

mon_status 命令返回 Ceph Monitor 的信息，例如：

- 状态

- 等级
- 选举时期
- Monitor map (**monmap**)

如果 Ceph 监控器能够形成仲裁，请将 **mon_status** 与 **ceph** 命令行实用程序一起使用。

如果 Ceph monitor 无法形成仲裁，但 **ceph-mon** 守护进程正在运行，请使用管理套接字来执行 **mon_status**。

mon_status 输出示例

```
{
    "name": "mon.3",
    "rank": 2,
    "state": "peon",
    "election_epoch": 96,
    "quorum": [
        1,
        2
    ],
    "outside_quorum": [],
    "extra_probe_peers": [],
    "sync_provider": [],
    "monmap": {
        "epoch": 1,
        "fsid": "d5552d32-9d1d-436c-8db1-ab5fc2c63cd0",
        "modified": "0.000000",
        "created": "0.000000",
        "mons": [
            {
                "rank": 0,
                "name": "mon.1",
                "addr": "172.25.1.10:6789\/0"
            },
            {
                "rank": 1,
                "name": "mon.2",
                "addr": "172.25.1.12:6789\/0"
            },
            {
                "rank": 2,
                "name": "mon.3",
                "addr": "172.25.1.13:6789\/0"
            }
        ]
    }
}
```

Ceph monitor 状态

leader

在选择阶段，Ceph 监控器正在选举领导机。领导机是排名最高的 Ceph 监控器，其排名最低。在上例中，领导机是 **mon.1**。

peon

Ppeons 是仲裁中的 Ceph monitor，而不是领导。如果领导失败，则排名最高的 peon 将成为新的领导。

探测

如果 Ceph monitor 正在查找其他 Ceph 监控器，则 Ceph Monitor 处于 probing 状态。例如，在启动 Ceph monitor 后，它们会进行*探测*，直到找到在 Ceph Monitor map（**monmap**）中指定的足够的 Ceph monitor 来形成仲裁。

选择

如果 Ceph 监控器处于选举状态，则 Ceph Monitor 处于 electing 状态。通常，此状态会快速变化。

同步

如果 Ceph 监控器与其他 Ceph 监控器同步，则 Ceph 监控器处于同步状态，以加入仲裁。Ceph 监控器存储容量越小，同步过程越快。因此，如果您有一个大型存储，同步会需要更长的时间。

其它资源

- 详情请参阅 Red Hat *Ceph* Storage 7 *管理指南中的使用* Ceph 管理套接字一节。
- 请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 第 4.1.1 节 “Ceph 监控错误消息”。
- 请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 第 4.1.2 节 “Ceph 日志中常见的 Ceph Monitor 错误消息”。

4.2. 注入 MONMAP

如果 Ceph monitor 具有过时或损坏的 Ceph monitor map(**monmap**)，它就无法加入仲裁，因为它正在尝试访问错误的 IP 地址上的其他 Ceph monitor。

修复此问题的最安全的方法是从其他 Ceph 监控器获取和注入实际的 Ceph monitor map。

注意

此操作将覆盖 Ceph monitor 保存的现有 Ceph monitor map。

此步骤显示，当其他 Ceph 监控器能够形成仲裁时，或者至少一个 Ceph monitor 具有正确的 Ceph monitor map 时，如何注入 Ceph Monitor map。如果所有 Ceph 监控器都有损坏的存储，因此也具有 Ceph Monitor 映射，*请参阅恢复 Ceph Monitor 存储*。

先决条件

- 访问 Ceph Monitor map。
- Ceph 监控节点的根级别访问权限.

流程

1. 如果剩余的 Ceph 监控器能够形成仲裁，请使用 **ceph mon getmap** 命令获取 Ceph Monitor map：

   示例

   ```
   [ceph: root@host01 /]# ceph mon getmap -o /tmp/monmap
   ```

2. 如果剩余的 Ceph 监控器无法形成仲裁，并且至少有一个 Ceph monitor 带有正确的 Ceph monitor map，请从该 Ceph 监控器复制它：

   a. 停止您要从中复制 Ceph Monitor 映射的 Ceph Monitor：

      语法

      ```
      systemctl stop ceph-FSID@DAEMON_NAME
      ```

      示例

      ```
      [root@mon ~]# systemctl stop ceph-b404c440-9e4c-11ec-a28a-
      001a4a0001df@mon.host01.service
      ```

   b. 复制 Ceph Monitor 映射：

      语法

      ```
      ceph-mon -i ID --extract-monmap /tmp/monmap
      ```

      使用您要从中复制 Ceph Monitor 映射的 Ceph Monitor ***ID*** 替换 ID：

      示例

      ```
      [ceph: root@host01 /]#  ceph-mon -i mon.a  --extract-monmap /tmp/monmap
      ```

3. 停止具有损坏或过时的 Ceph Monitor 映射的 Ceph Monitor：

   语法

   ```
   systemctl stop ceph-FSID@DAEMON_NAME
   ```

   示例

   ```
   [root@mon ~]# systemctl stop ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@mon.host01.service
   ```

4. 注入 Ceph Monitor 映射：

   语法

   ```
   ceph-mon -i ID --inject-monmap /tmp/monmap
   ```

   将 ***ID*** 替换为 Ceph Monitor 的 ID，并带有损坏的或过时的 Ceph Monitor 映射：

   示例

   ```
   [root@mon ~]# ceph-mon -i mon.host01 --inject-monmap /tmp/monmap
   ```

5. 启动 Ceph Monitor：

   语法

```
systemctl start ceph-FSID@DAEMON_NAME
```

示例

```
[root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

如果您从另一个 Ceph 监控器复制了 Ceph monitor map，则也启动该 Ceph monitor：

语法

```
systemctl start ceph-FSID@DAEMON_NAME
```

示例

```
[root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

其它资源

- 请参阅 *Ceph 监控器没有仲裁*
- *请参阅恢复 Ceph Monitor 存储*

4.3. 替换失败的 MONITOR

当 Ceph Monitor 具有损坏的存储时，您可以替换存储集群中的监控器。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 能够形成仲裁。
- Ceph 监控节点的根级别访问权限。

流程

1. 从 Monitor 主机，删除 Monitor 存储（默认位于 **/var/lib/ceph/mon/*CLUSTER_NAME-SHORT_HOST_NAME***）：

   ```
   rm -rf /var/lib/ceph/mon/CLUSTER_NAME-SHORT_HOST_NAME
   ```

   指定 monitor 主机和集群名称的短主机名。例如，要从名为 **remote** 的集群中删除 **host1** 上运行的 monitor 存储：

   ```
   [root@mon ~]# rm -rf /var/lib/ceph/mon/remote-host1
   ```

2. 从 monitor 映射(**monmap**)中删除 monitor：

   ```
   ceph mon remove SHORT_HOST_NAME --cluster CLUSTER_NAME
   ```

指定 monitor 主机和集群名称的短主机名。例如，要从名为 **remote** 的集群中删除 **host1** 上运行的 monitor：

```
[ceph: root@host01 /]# ceph mon remove host01 --cluster remote
```

3. 排除故障并修复与 monitor 主机底层文件系统或硬件相关的问题。

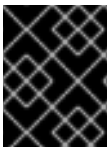
其它资源

- 详情请参阅 *Ceph 监控器没有仲裁*。

4.4. 紧凑 MONITOR 存储

当 monitor 存储大小增大时，您可以压缩它：

- 使用 **ceph tell** 命令动态使用。
- 在 **ceph-mon** 守护进程的开头。
- 在 **ceph-mon** 守护进程没有运行时，使用 **ceph-monstore-tool**。当前面提到的方法无法压缩 monitor 存储或者 monitor 超出仲裁并且其日志包含 **Caught signal (Bus error)** 错误信息时，可使用此方法。



重要

当集群没有处于 **active+clean** 状态或重新平衡过程中，监控存储大小会改变。因此，在重新平衡完成后，压缩 monitor 存储。另外，确保放置组处于 **active+clean** 状态。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- Ceph 监控节点的根级别访问权限.

流程

1. 在 **ceph-mon** 守护进程运行时压缩 monitor 存储：

   语法

   ```
   ceph tell mon.HOST_NAME compact
   ```

2. 将 ***HOST_NAME*** 替换为运行 **ceph-mon** 主机的短主机名。在不确定时使用 **hostname -s** 命令。

   示例

   ```
   [ceph: root@host01 /]# ceph tell mon.host01 compact
   ```

3. 在 **[mon]** 部分下的 Ceph 配置中添加以下参数：

   ```
   [mon]
   mon_compact_on_start = true
   ```

4. 重启 **ceph-mon** 守护进程：

   语法

   ```
   systemctl restart ceph-FSID@DAEMON_NAME
   ```

   示例

   ```
   [root@mon ~]# systemctl restart ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@mon.host01.service
   ```

5. 确保 monitor 已形成仲裁：

   ```
   [ceph: root@host01 /]# ceph mon stat
   ```

6. 如果需要，在其他 monitor 上重复这些步骤。

   注意

   开始之前，请确保已安装了 **ceph-test** 软件包。

7. 验证具有大存储的 **ceph-mon** 守护进程是否没有运行。如果需要，停止守护进程。

   语法

   ```
   systemctl status ceph-FSID@DAEMON_NAME
   systemctl stop ceph-FSID@DAEMON_NAME
   ```

   示例

   ```
   [root@mon ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@mon.host01.service
   [root@mon ~]# systemctl stop ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@mon.host01.service
   ```

8. 紧凑 monitor 存储：

   语法

   ```
   ceph-monstore-tool /var/lib/ceph/CLUSTER_FSID/mon.HOST_NAME compact
   ```

   使用 monitor 主机的短主机名替换 ***HOST_NAME***。

   示例

   ```
   [ceph: root@host01 /]# ceph-monstore-tool /var/lib/ceph/b404c440-9e4c-11ec-a28a-
   001a4a0001df/mon.host01 compact
   ```

9. 再次启动 **ceph-mon** ：

   语法

```
systemctl start ceph-FSID@DAEMON_NAME
```

示例

```
[root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

其它资源

- 请参阅 *Ceph Monitor 存储太大*
- 请参阅 *Ceph 监控器没有仲裁*

4.5. 打开 CEPH 管理器的端口

ceph-mgr 守护进程从与 **ceph-osd** 守护进程相同的端口上的 OSD 接收放置组信息。如果没有打开这些端口，集群将从 **HEALTH_OK** 状态变为 **HEALTH_WARN**，并且指出 PG 为 **unknown** 并显示为 unknown 状态的 PG 数量。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- Ceph 管理器的根级别访问权限。

流程

1. 要解决这种情况，对于运行 **ceph-mgr** 守护进程的每个主机，请打开端口 **6800-7300**。

   示例

   ```
   [root@ceph-mgr] # firewall-cmd --add-port 6800-7300/tcp
   [root@ceph-mgr] # firewall-cmd --add-port 6800-7300/tcp --permanent
   ```

2. 重启 **ceph-mgr** 守护进程。

4.6. 恢复 CEPH MONITOR 存储

Ceph 监控器将 cluster map 存储在键值存储中，如 RocksDB。如果存储在 monitor 上损坏，则 monitor 会意外终止，且无法再次启动。Ceph 日志可能包括以下错误：

```
Corruption: error in middle of record
Corruption: 1 missing files; e.g.: /var/lib/ceph/mon/mon.0/store.db/1234567.ldb
```

Red Hat Ceph Storage 集群使用至少三个 Ceph 监控器，以便在一个失败时，它可以被替换为另一个。然而，在某些情况下，所有 Ceph 监控器可能会损坏存储。例如，当 Ceph 监控节点配置错误的磁盘或文件系统设置时，断电可能会破坏底层文件系统。

如果所有 Ceph 监控上都存在损坏，则可以使用名为 **ceph-monstore-tool** 和 **ceph-objectstore-tool** 的实用程序，通过 OSD 节点上存储的信息来恢复它。

重要

这些步骤无法恢复以下信息：

- 元数据服务器(MDS)密钥环和映射
- 放置组设置：
  - 使用 **ceph pg set_full_ratio** 命令设置 **full ratio**
  - 使用 **ceph pg set_nearfull_ratio** 命令设置 **nearfull ratio**

重要

从旧备份中恢复 Ceph Monitor 存储。使用以下步骤从当前集群状态重建 Ceph Monitor 存储，并从中恢复。

4.6.1. 使用 BlueStore 时恢复 Ceph Monitor 存储

如果 Ceph Monitor 存储在所有 Ceph 监控器上损坏，且您使用 BlueStore 后端，请按照以下步骤操作。

在容器化环境中，此方法需要附加 Ceph 存储库并首先恢复到非容器化 Ceph monitor。

警告

这个流程可能会导致数据丢失。如果您不确定这个过程中的任何步骤，请联系红帽技术支持以获取恢复过程的帮助。

先决条件

- 所有 OSD 容器都已停止。
- 根据 Ceph 节点上的角色，启用 Ceph 存储库。
- **ceph-test** 和 **rsync** 软件包安装在 OSD 和 monitor 节点上。
- **ceph-mon** 软件包安装在 monitor 节点上。
- **ceph-osd** 软件包安装在 OSD 节点上。

流程

1. 将所有带有 Ceph 数据的磁盘挂载到临时位置。对所有 OSD 节点重复此步骤。
   a. 使用 **ceph-volume** 命令列出数据分区：

   示例

   ```
   [ceph: root@host01 /]# ceph-volume lvm list
   ```

   b. 将数据分区挂载到临时位置：

语法

```
mount -t tmpfs tmpfs /var/lib/ceph/osd/ceph-$i
```

c. 恢复 SELinux 上下文：

语法

```
for i in {OSD_ID}; do restorecon /var/lib/ceph/osd/ceph-$i; done
```

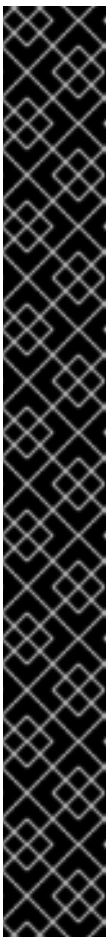
将 *OSD_ID* 替换为 OSD 节点上的 Ceph OSD ID 数字、空格分隔的列表。

d. 将所有者和组更改为 **ceph:ceph** ：

语法

```
for i in {OSD_ID}; do chown -R ceph:ceph /var/lib/ceph/osd/ceph-$i; done
```

将 *OSD_ID* 替换为 OSD 节点上的 Ceph OSD ID 数字、空格分隔的列表。



重要

由于一个程序错误会导致 **update-mon-db** 命令为 monitor 数据库使用额外的 **db** 和 **db.slow** 目录，因此您还必须复制这些目录。要做到这一点：

1. 准备容器外部的临时位置，以挂载和访问 OSD 数据库，并提取恢复 Ceph Monitor 所需的 OSD 映射：

   语法

   ```
   ceph-bluestore-tool --cluster=ceph prime-osd-dir --dev OSD-DATA --path /var/lib/ceph/osd/ceph-OSD-ID
   ```

   将 *OSD-DATA* 替换为卷组(VG)或逻辑卷(LV)路径，将 OSD 数据和 *OSD 的 ID* 替换为 OSD 的 ID。

2. 在 BlueStore 数据库和 **block.db** 之间创建符号链接：

   语法

   ```
   ln -snf BLUESTORE DATABASE /var/lib/ceph/osd/ceph-OSD-ID/block.db
   ```

   将 *BLUESTORE-DATABASE* 替换为卷组(VG)或逻辑卷(LV)路径，将 *OSD-ID* 替换为 OSD 的 ID。

2. 从具有损坏存储的 Ceph 监控节点使用以下命令：为所有节点上的所有 OSD 重复它们。

   a. 从所有 OSD 节点收集 cluster map：

   示例

   ```
   [root@host01 ~]# cd /root/
   ```

```
[root@host01 ~]# ms=/tmp/monstore/
[root@host01 ~]# db=/root/db/
[root@host01 ~]# db_slow=/root/db.slow/

[root@host01 ~]# mkdir $ms
[root@host01 ~]# for host in $osd_nodes; do
                echo "$host"
                rsync -avz $ms $host:$ms
                rsync -avz $db $host:$db
                rsync -avz $db_slow $host:$db_slow

                rm -rf $ms
                rm -rf $db
                rm -rf $db_slow

                sh -t $host <<EOF
                  for osd in /var/lib/ceph/osd/ceph-*; do
                    ceph-objectstore-tool --type bluestore --data-path \$osd --op update-mon-db
--mon-store-path $ms

                  done
                EOF

                     rsync -avz $host:$ms $ms
                     rsync -avz $host:$db $db
                     rsync -avz $host:$db_slow $db_slow
                done
```

b. 设置适当的功能：

示例

```
[ceph: root@host01 /]# ceph-authtool /etc/ceph/ceph.client.admin.keyring -n mon. --cap
mon 'allow *' --gen-key
[ceph: root@host01 /]# cat /etc/ceph/ceph.client.admin.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 *"
```

c. 将所有 **sst** 文件从 **db** 和 **db.slow** 目录移到临时位置：

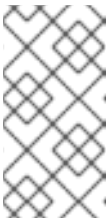
示例

```
[ceph: root@host01 /]# mv /root/db/*.sst /root/db.slow/*.sst /tmp/monstore/store.db
```

d. 从收集的 map 重建 monitor 存储：

示例

```
[ceph: root@host01 /]# ceph-monstore-tool /tmp/monstore rebuild -- --keyring
/etc/ceph/ceph.client.admin
```



注意

使用此命令后，Ceph 身份验证数据库中仅存在从 OSD 提取的密钥环和 **ceph-monstore-tool** 命令行中指定的密钥环。您必须重新创建或导入所有其他密钥环，如客户端、Ceph 管理器和 Ceph 对象网关等，以便这些客户端可以访问集群。

e. 备份损坏的存储。对所有 Ceph 监控节点重复此步骤：

语法

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

将 *HOSTNAME* 替换为 Ceph 监控节点的主机名。

f. 替换损坏的存储。对所有 Ceph 监控节点重复此步骤：

语法

```
scp -r /tmp/monstore/store.db HOSTNAME:/var/lib/ceph/mon/ceph-HOSTNAME/
```

将 *HOSTNAME* 替换为 monitor 节点的主机名。

g. 更改新存储的所有者。对所有 Ceph 监控节点重复此步骤：

语法

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

将 *HOSTNAME* 替换为 Ceph 监控节点的主机名。

3. 卸载所有节点上所有临时挂载的 OSD：

示例

```
[root@host01 ~]# umount /var/lib/ceph/osd/ceph-*
```

4. 启动所有 Ceph Monitor 守护进程：

语法

```
systemctl start ceph-FSID@DAEMON_NAME
```

示例

```
[root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

5. 确保 monitor 能够形成仲裁：

   语法

   ```
   ceph -s
   ```

   将 *HOSTNAME* 替换为 Ceph 监控节点的主机名。

6. 导入 Ceph Manager 密钥环并启动所有 Ceph Manager 进程：

   语法

   ```
   ceph auth import -i /etc/ceph/ceph.mgr.HOSTNAME.keyring
   systemctl start ceph-FSID@DAEMON_NAME
   ```

   示例

   ```
   [root@mon ~]# systemctl start ceph-b341e254-b165-11ed-a564-
   ac1f6bb26e8c@mgr.extensa003.exrqql.service
   ```

   将 *HOSTNAME* 替换为 Ceph Manager 节点的主机名。

7. 启动所有 OSD 节点中的所有 OSD 进程。对集群中的所有 OSD 重复此操作：

   语法

   ```
   systemctl start ceph-FSID@osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

8. 确保 OSD 返回到服务：

   示例

   ```
   [ceph: root@host01 /]# ceph -s
   ```

其它资源

- 有关将 Ceph 节点注册到内容交付网络(CDN)的详细信息，请参阅 *Red Hat Ceph Storage 安装指南中的将 Red Hat Ceph Storage 节点注册到 CDN 并附加订阅* 部分。
- 如需与网络相关的问题，请参阅 *Red Hat Ceph Storage 故障排除指南* 中的*对网络进行故障排除*。

第 5 章 CEPH OSD 故障排除

本章介绍了如何修复与 Ceph OSD 相关的最常见的错误。

先决条件

- 验证您的网络连接。*详情请参阅对网络问题进行故障排除*。
- 使用 **ceph health** 命令验证 monitor 具有仲裁。如果命令返回健康状态 (**HEALTH_OK**、**HEALTH_WARN** 或 **HEALTH_ERR**)，则 monitor 能够形成仲裁。如果没有，请首先解决任何 monitor 问题。详情请参阅 *Ceph Monitor 故障排除*。如需有关 **ceph health** 的详细信息，*请参阅了解 Ceph 健康状况*。
- （可选）停止重新平衡过程，以节省时间和资源。*详情请参阅停止和启动重新平衡*。

5.1. 最常见的 CEPH OSD 错误

下表列出了 **ceph health detail** 命令返回的最常见错误消息，或者包含在 Ceph 日志中。这些表中提供了相应部分的链接，这些部分解释了错误并指向修复问题的特定程序。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。

5.1.1. Ceph OSD 错误消息

常见 Ceph OSD 错误消息表，以及潜在的修复。

错误消息	查看
HEALTH_ERR	
full osds	OSD 已满
HEALTH_WARN	
backfillfull osds	backfillfull OSDS
nearfull osds	nearfull OSDs
OSD 为 down	OSD 下线 Fflapping OSD
requests are blocked	请求慢或请求被阻止
slow requests	请求慢或请求被阻止

5.1.2. Ceph 日志中常见的 Ceph OSD 错误消息

Ceph 日志中找到的常见 Ceph OSD 错误消息表，以及到潜在修复的链接。

错误消息	日志文件	查看
heartbeat_check ：没有来自 osd.X 的回复	主集群日志	Fflapping OSD
错误地标记为 down	主集群日志	Fflapping OSD
OSD 的请求速度较慢	主集群日志	请求慢或请求被阻止
FAILED assert(0 == "hit suicide timeout")	OSD 日志	OSD 下线

5.1.3. OSD 已满

ceph health detail 命令返回类似如下的错误消息：

```
HEALTH_ERR 1 full osds
osd.3 is full at 95%
```

这意味着

Ceph 可防止客户端对完整 OSD 节点执行 I/O 操作，以避免丢失数据。当集群达到由 **mon_osd_full_ratio** 参数设定的容量时，它会返回 **HEALTH_ERR full osds** 消息。默认情况下，此参数被设置为 **0.95**，即 集群容量的 95%。

要排除此问题，请执行以下操作

确定使用了多少原始存储(**%RAW USED**)：

```
ceph df
```

如果 **%RAW USED** 超过 70-75%，您可以：

- 删除不必要的数据.这是一个短期解决方案，可以避免生产停机时间。
- 通过添加新 OSD 节点来扩展集群。这是红帽推荐的长期解决方案。

其它资源

- *Red Hat Ceph Storage 故障排除指南中的 Nearfull OSDs*。
- 详情请参阅*从完整存储集群中删除数据*。

5.1.4. backfillfull OSDs

ceph health detail 命令返回类似如下的错误消息：

```
health: HEALTH_WARN
3 backfillfull osd(s)
Low space hindering backfill (add storage if this doesn't resolve itself): 32 pgs backfill_toofull
```

这意味着

当一个或多个 OSD 超过 backfillfull 阈值时，Ceph 会阻止数据重新平衡到这个设备。这是一个早期警告，重新平衡可能无法完成，且集群已接近满。默认的 backfullfull 阈值为 90%。

要排除此问题

检查池的利用率：

```
ceph df
```

如果 **%RAW USED** 超过 70-75%，您可以执行以下操作之一：

- 删除不必要的数据.这是一个短期解决方案，可以避免生产停机时间。
- 通过添加新 OSD 节点来扩展集群。这是红帽推荐的长期解决方案。
- 为包含 PG 处于 **backfull_toofull** 的 OSD 增加 **backfillfull** 比率，以允许恢复过程继续。尽快向集群添加新存储，或移除数据以防止填满 OSD。

  语法

  ```
  ceph osd set-backfillfull-ratio VALUE
  ```

  VALUE 的范围为 0.0 到 1.0。

  示例

  ```
  [ceph: root@host01/]# ceph osd set-backfillfull-ratio 0.92
  ```

其它资源

- *Red Hat Ceph Storage 故障排除指南中的 Nearfull OSDS*。
- 详情请参阅*从完整存储集群中删除数据*。

5.1.5. nearfull OSDs

ceph health detail 命令返回类似如下的错误消息：

```
HEALTH_WARN 1 nearfull osds
osd.2 is near full at 85%
```

这意味着

当集群达到由 **mon osd nearfull ratio defaults** 指定的容量时，Ceph 会返回 **nearfull osds** 信息。默认情况下，此参数设置为 **0.85**，这意味着集群容量的 85%。

Ceph 以最佳方式基于 CRUSH 层次结构分发数据，但它不能保证相等的分布。数据分布和 **nearfull osds** 信息的主要原因包括：

- OSD 在集群中的 OSD 节点之间没有平衡。也就是说，一些 OSD 节点托管的 OSD 比其他 OSD 高得多，或者 CRUSH map 中部分 OSD 的权重不足以满足其容量要求。
- PG 计数与 OSD 数量、用例、每个 OSD 目标 PG 和 OSD 利用率不同。

- 集群使用不正确的 CRUSH 可调项。
- OSD 的后端存储几乎已满。

排除此问题，请执行以下操作：

1. 验证 PG 数是否足够，并在需要时增加。
2. 验证您使用对集群版本的最佳 CRUSH 可调项，如果不是，则调整它们。
3. 根据利用率更改 OSD 的权重。
4. 确定 OSD 使用的磁盘上保留多少空间。
   a. 查看 OSD 一般使用的空间量：

   ```
   [ceph: root@host01 /]# ceph osd df
   ```

   b. 查看 OSD 在特定节点上使用的空间量。从包含 **nearfull** OSD 的节点使用以下命令：

   ```
   df
   ```

   c. 如果需要，添加新 OSD 节点。

其它资源

- *OSD 已满*
- 请参阅 Red Hat Ceph *Storage 7 的存储策略指南中的 Set a OSD 的 Weight by Utilization* 部分。
- 详情请参阅 Red Hat Ceph *Storage 7 的存储策略指南中的 CRUSH Tunables* 部分，以及如何 *测试 CRUSH map 可调整的影响 CRUSH map 可跨 OSD 在 Red Hat Ceph Storage 的 Red Hat Ceph Storage? 解决方案上进行 PG 分发* 部分。
- 详情请参阅 *增加放置组*。

5.1.6. OSD 下线

ceph health detail 命令返回类似如下的错误：

```
HEALTH_WARN 1/3 in osds are down
```

这意味着

由于服务故障或与其他 OSD 通信存在问题，其中一个 **ceph-osd** 进程不可用。因此，存活 **ceph-osd** 守护进程会向 monitor 报告这个失败。

如果 **ceph-osd** 守护进程未在运行，则代表底层 OSD 驱动器或文件系统已损坏，或者存在一些其他错误（如缺少密钥环）阻止守护进程启动。

在大多数情况下，网络问题会导致在 **ceph-osd** 守护进程运行时，仍标记为 **down** 的情况。

要排除此问题，请执行以下操作

1. 确定哪个 OSD 为 **down** ：

```
[ceph: root@host01 /]# ceph health detail
HEALTH_WARN 1/3 in osds are down
osd.0 is down since epoch 23, last address 192.168.106.220:6800/11080
```

2. 尝试重启 **ceph-osd** 守护进程。将 *OSD_ID* 替换为 down 的 OSD 的 ID：

   语法

   ```
   systemctl restart ceph-FSID@osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# systemctl restart ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

   a. 如果您无法启动 **ceph-osd**，请按照 ***ceph-osd** 守护进程中的步骤启动*。

   b. 如果您能够启动 **ceph-osd** 守护进程，但它标记为 **down**，请按照 ***ceph-osd** 守护进程运行中的步骤，但仍然标记为"down "*。

ceph-osd 守护进程无法启动

1. 如果您的节点包含多个 OSD（通常超过 12 倍），请验证默认最多线程数（PID 数）是否足够。详情请参阅 *增加 PID 计数*。

2. 验证 OSD 数据和日志分区是否已正确挂载。您可以使用 **ceph-volume lvm list** 命令列出与 Ceph Storage 集群关联的所有设备和卷，然后手动检查是否正确挂载它们。详情请查看 **mount (8)** 手册页。

3. 如果您遇到 **ERROR: missing keyring，则无法将 cephx 用于身份验证** 错误消息，则 OSD 是缺少的密钥环。

4. 如果您遇到 **ERROR: unable to open OSD superblock on /var/lib/ceph/osd/ceph-1** 错误信息，**ceph-osd** 守护进程不能读底层的文件系统。有关如何排除故障并修复此错误的说明，请参阅以下步骤。

   a. 检查对应的日志文件，以确定故障的原因。默认情况下，Ceph 在启用了日志记录到文件后将日志文件存储在 **/var/log/ceph/*CLUSTER_FSID*/** 目录中。

   b. **EIO** 错误消息表示底层磁盘失败。若要修复此问题，可替换底层 OSD 磁盘。详情请参阅 *替换 OSD 驱动器*。

   c. 如果日志包含任何其他 **FAILED assert** 错误，如以下错误，请创建一个支持问题单。*详情请参阅联系红帽支持以获取服务*。

      ```
      FAILED assert(0 == "hit suicide timeout")
      ```

5. 检查 **dmesg** 输出是否有底层文件系统或磁盘的错误：

   ```
   dmesg
   ```

   a. **类似于以下内容的错误 -5** 错误消息表示底层 XFS 文件系统崩溃。如需解决这个问题的信息，请参阅红帽客户门户网站中的 *What is the meaning of "xfs_log_force: error -5 returned"?* 。

```
xfs_log_force: error -5 returned
```

b. 如果 **dmesg** 输出包含任何 **SCSI error** 错误信息，请参阅红帽客户门户网站中的 *SCSI Error Codes Solution Finder* 解决方案，以确定解决问题的最佳方法。

c. 或者，如果您无法修复底层文件系统，请替换 OSD 驱动器。详情请参阅 *替换 OSD 驱动器*。

6. 如果 OSD 因分段错误而出现故障，如以下 OSD，请收集必要的信息并创建一个支持问题单。*详情请参阅联系红帽支持以获取服务*。

```
Caught signal (Segmentation fault)
```

ceph-osd 正在运行，但仍标记为 down

1. 检查对应的日志文件，以确定故障的原因。默认情况下，Ceph 在启用了日志记录到文件后将日志文件存储在 **/var/log/ceph/*CLUSTER_FSID*/** 目录中。

a. 如果日志包含与以下类似的错误消息，请参阅 *Flapping OSD*。

```
wrongly marked me down
heartbeat_check: no reply from osd.2 since back
```

b. 如果您看到任何其他错误，请创建一个支持问题单。*详情请参阅联系红帽支持以获取服务*。

其它资源

- *Fflapping OSD*
- *Stale 放置组*
- 请参阅 *Ceph 守护进程日志*，以启用日志记录到文件。

5.1.7. Fflapping OSD

ceph -w | grep osds 命令重复显示 OSD 为 **down**，然后在短时间内再次显示为 **up** ：

```
ceph -w | grep osds
2022-05-05 06:27:20.810535 mon.0 [INF] osdmap e609: 9 osds: 8 up, 9 in
2022-05-05 06:27:24.120611 mon.0 [INF] osdmap e611: 9 osds: 7 up, 9 in
2022-05-05 06:27:25.975622 mon.0 [INF] HEALTH_WARN; 118 pgs stale; 2/9 in osds are down
2022-05-05 06:27:27.489790 mon.0 [INF] osdmap e614: 9 osds: 6 up, 9 in
2022-05-05 06:27:36.540000 mon.0 [INF] osdmap e616: 9 osds: 7 up, 9 in
2022-05-05 06:27:39.681913 mon.0 [INF] osdmap e618: 9 osds: 8 up, 9 in
2022-05-05 06:27:43.269401 mon.0 [INF] osdmap e620: 9 osds: 9 up, 9 in
2022-05-05 06:27:54.884426 mon.0 [INF] osdmap e622: 9 osds: 8 up, 9 in
2022-05-05 06:27:57.398706 mon.0 [INF] osdmap e624: 9 osds: 7 up, 9 in
2022-05-05 06:27:59.669841 mon.0 [INF] osdmap e625: 9 osds: 6 up, 9 in
2022-05-05 06:28:07.043677 mon.0 [INF] osdmap e628: 9 osds: 7 up, 9 in
2022-05-05 06:28:10.512331 mon.0 [INF] osdmap e630: 9 osds: 8 up, 9 in
2022-05-05 06:28:12.670923 mon.0 [INF] osdmap e631: 9 osds: 9 up, 9 in
```

此外，Ceph 日志包含类似以下的错误消息：

> 2022-05-25 03:44:06.510583 osd.50 127.0.0.1:6801/149046 18992 : cluster [WRN] map e600547 wrongly marked me down

> 2022-05-25 19:00:08.906864 7fa2a0033700 -1 osd.254 609110 heartbeat_check: no reply from osd.2 since back 2021-07-25 19:00:07.444113 front 2021-07-25 18:59:48.311935 (cutoff 2021-07-25 18:59:48.906862)

这意味着

flapping OSD 的主要原因包括：

- 某些存储集群操作（如清理或恢复）在具有大型索引或大型放置组的对象上执行这些操作时（例如，需要一定时间）。通常，这些操作完成后，flapping OSD 问题会解决。
- 底层物理硬件的问题。在本例中，**ceph health detail** 命令也会返回 **slow requests** 错误信息。
- 与网络相关的问题。

Ceph OSD 无法管理存储集群的专用网络出现故障的情况，或者显著延迟位于面向公共客户端的网络上。

Ceph OSD 使用专用网络在彼此间发送心跳数据包，以代表它们处于 **up** 和 **in** 状态。如果私有存储集群网络无法正常工作，OSD 无法发送和接收 heartbeat 数据包。因此，它们会将彼此报告为 **down** 到 Ceph 监控器，同时将自身标记为 **up**。

Ceph 配置文件中的以下参数会影响此行为：

参数	描述	默认值
osd_heartbeat_grace_time	在将 OSD 报告为 **down** 到 Ceph 监控器之前，OSD 等待心跳数据包返回的时间。	20 秒
mon_osd_min_down_reporters	在 Ceph 监控将该 OSD 标记为 **down** 之前，需要多少 OSD 报告了另一个 OSD 为 **down**	2

下表显示了在默认配置中，Ceph 监控器在只有一个 OSD 三次报告了第一个 OSD 为 **down** 时，才会将 OSD 标记为 **down**。在某些情况下，如果一个主机遇到网络问题，整个集群可能会遇到流化 OSD。这是因为主机上的 OSD 将报告集群中的其他 OSD 为 **down**。

注意

flapping OSD 方案不包括在 OSD 进程启动时的情况，然后立即终止。

要排除此问题，请执行以下操作

1. 再次检查 **ceph health detail** 命令的输出。如果其中包含 **slow requests** 错误信息，请参阅如何排除此问题。

```
ceph health detail
HEALTH_WARN 30 requests are blocked > 32 sec; 3 osds have slow requests
30 ops are blocked > 268435 sec
1 ops are blocked > 268435 sec on osd.11
```

```
1 ops are blocked > 268435 sec on osd.18
28 ops are blocked > 268435 sec on osd.39
3 osds have slow requests
```

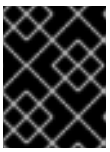
2. 确定哪些 OSD 标记为 **down**，以及它们所在的节点上：

```
ceph osd tree | grep down
```

3. 在包含 flapping OSD 的节点上，对任何网络问题进行故障排除并修复。

4. 或者，您可以通过设置 **noup** 和 **nodown** 标记，临时强制 monitor 停止将 OSD 标记为 **down** 和 **up**：

```
ceph osd set noup
ceph osd set nodown
```



重要

使用 **noup** 和 **nodown** 标志不会修复问题的根本原因，而是只防止 OSD 出现问题。要创建一个支持问题单，请参阅 *联系红帽支持部分*。

重要

扁平化 OSD 可能是由 Ceph OSD 节点上的 MTU 错误配置、网络交换机级别或两者导致的。要解决这个问题，请在所有存储集群节点上将 MTU 设置为统一大小，包括在核心上和通过计划停机访问网络交换机。不要调整 **osd heartbeat min size**，因为更改此设置可能会隐藏网络中的问题，且无法解决实际网络不一致的问题。

其它资源

- 请参阅 *Red Hat Ceph Storage Architecture Guide* 中的 *Ceph heartbeat* 部分。
- 请参阅 *Red Hat Ceph Storage 故障排除指南* 中的 *请求较慢或请求被阻塞*部分。

5.1.8. 请求慢或请求被阻塞

ceph-osd 守护进程响应请求的速度非常慢，**ceph health detail** 命令返回类似如下的错误消息：

```
HEALTH_WARN 30 requests are blocked > 32 sec; 3 osds have slow requests
30 ops are blocked > 268435 sec
1 ops are blocked > 268435 sec on osd.11
1 ops are blocked > 268435 sec on osd.18
28 ops are blocked > 268435 sec on osd.39
3 osds have slow requests
```

此外，Ceph 日志包含类似于以下的错误消息：

```
2022-05-24 13:18:10.024659 osd.1 127.0.0.1:6812/3032 9 : cluster [WRN] 6 slow requests, 6
included below; oldest blocked for > 61.758455 secs
```

```
2022-05-25 03:44:06.510583 osd.50 [WRN] slow request 30.005692 seconds old, received at {date-
time}: osd_op(client.4240.0:8 benchmark_data_ceph-1_39426_object7 [write 0~4194304]
0.69848840) v4 currently waiting for subops from [610]
```

-

这意味着

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。

造成 OSD 请求缓慢的主要原因包括：

- 底层硬件的问题，如磁盘驱动器、主机、机架或网络交换机
- 与网络相关的问题。这些问题通常与 flapping OSD 相关。详情请参阅 *Flapping OSD*。
- 系统负载

下表显示了较慢请求的类型。使用 **dump_historic_ops** 管理套接字命令确定较慢请求的类型。有关管理套接字的详细信息，请参阅 Red Hat *Ceph* Storage 7 *管理指南中的使用* Ceph 管理套接字一节。

慢速请求类型	描述
等待 rw 锁定	OSD 正在等待在 PG 上获取操作的锁定。
waiting for subops	OSD 正在等待副本 OSD 将操作应用到日志。
no flag points reached	OSD 无法到达任何主要的操作里程碑。
等待降级对象	OSD 尚未复制一个对象的指定次数。

要排除此问题，请执行以下操作

1. 确定具有缓慢或块请求的 OSD 是否共享一个通用的硬件，如磁盘驱动器、主机、机架或网络交换机。
2. 如果 OSD 共享一个磁盘：
   a. 使用 **smartmontools** 工具检查磁盘的健康状态或日志来确定磁盘上的任何错误。

注意

smartmontools 工具包含在 **smartmontools** 软件包中。

   b. 使用 **iostat** 实用程序获取 OSD 磁盘上的 I/O 等待报告 (**%iowai**)，以确定磁盘是否负载过重。

注意

iostat 实用程序包含在 **sysstat** 软件包中。

3. 如果 OSD 与另一个服务共享节点：
   a. 检查 RAM 和 CPU 使用率
   b. 使用 **netstat** 工具查看网络接口控制器(NIC) 上的网络统计信息，并对任何网络问题进行故障排除。

4. 如果 OSD 共享一个机架，请检查机架的网络交换机。例如，如果您使用巨型帧，请验证路径中的 NIC 是否已设置了巨型帧。

5. 如果您无法确定请求速度较慢的 OSD 共享的硬件部分，或者无法对硬件和网络问题进行故障排除和修复，请创建一个支持问题单。*详情请参阅联系红帽支持以获取服务*。

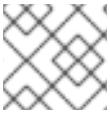
其它资源

- 详情请参阅 *Red Hat Ceph Storage Administration Guide* 中的 *Using the Ceph Administration Socket* 部分。

5.2. 停止并启动重新平衡

当 OSD 出现故障或您停止了它时，CRUSH 算法会自动启动重新平衡过程，以在剩余的 OSD 之间重新分发数据。

重新平衡可能需要时间和资源，因此请考虑在故障排除或维护 OSD 期间停止重新平衡。



注意

在故障排除和维护过程中，已停止的 OSD 中的放置组会变为 **degraded**。

先决条件

- Ceph 监控节点的根级别访问权限.

流程

1. 登录到 Cephadm shell：

   示例

   ```
   [root@host01 ~]# cephadm shell
   ```

2. 在停止 OSD 前设置 **noout** 标志：

   示例

   ```
   [ceph: root@host01 /]# ceph osd set noout
   ```

3. 完成故障排除或维护后，取消设置 **noout** 标志以启动重新平衡：

   示例

   ```
   [ceph: root@host01 /]# ceph osd unset noout
   ```

其它资源

- *Red Hat Ceph Storage 架构指南中的 重新平衡和恢复* 部分。

5.3. 替换 OSD 驱动器

Ceph 是为容错而设计的，这意味着可以在不丢失数据的情况下以 **degraded** 状态运行。因此，即使数据存储驱动器失败，Ceph 也能运行。在故障驱动器的上下文中，**degraded** 状态意味着其他 OSD 上存储的数据的额外副本将自动回填到集群中的其他 OSD。不过，如果发生这种情况，请替换失败的 OSD 驱动器，并手动重新创建 OSD。

当驱动器出现故障时，Ceph 将会报告 OSD 为 **down** ：

```
HEALTH_WARN 1/3 in osds are down
osd.0 is down since epoch 23, last address 192.168.106.220:6800/11080
```

注意

因为网络或权限问题的出现，Ceph 也可以将 OSD 标记为 **down**。详情请参阅*关闭 OSD*。

现代服务器通常使用热插拔驱动器进行部署，以便您可以将失败的驱动器替换为新的驱动器，而无需关闭节点。整个流程包括这些步骤：

1. 从 Ceph 集群移除 OSD。详情请参阅*从 Ceph 集群中删除 OSD*。
2. 替换驱动器。详情请查看*替换物理驱动器部分*。
3. 将 OSD 添加到集群中。详情请参阅*将 OSD 添加到 Ceph 集群*的步骤。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- Ceph 监控节点的根级别访问权限.
- 至少一个 OSD 为 **down**。

从 Ceph 集群中删除 OSD

1. 登录到 Cephadm shell：

   示例

   ```
   [root@host01 ~]# cephadm shell
   ```

2. 确定哪个 OSD 为 **down**。

   示例

   ```
   [ceph: root@host01 /]# ceph osd tree | grep -i down
   ID  CLASS  WEIGHT   TYPE NAME          STATUS  REWEIGHT  PRI-AFF
    0   hdd 0.00999        osd.0      down  1.00000          1.00000
   ```

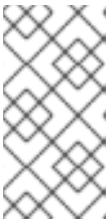
3. 将 OSD 标记为 **out**，以便集群重新平衡并将其数据复制到其他 OSD。

   语法

   ```
   ceph osd out OSD_ID.
   ```

示例

```
[ceph: root@host01 /]# ceph osd out osd.0
marked out osd.0.
```



注意

如果 OSD 为 **down**，Ceph 会在 600 秒后自动将其标记为 **out**，如果它没有根据 **mon_osd_down_out_interval** 参数从 OSD 接收任何 heartbeat 数据包。发生这种情况时，具有故障 OSD 数据副本的其他 OSD 开始回填，以确保集群中存在所需的副本数。在集群回填时，集群将处于 **degraded** 状态。

4. 确保失败的 OSD 正在回填。

示例

```
[ceph: root@host01 /]# ceph -w | grep backfill
2022-05-02 04:48:03.403872 mon.0 [INF] pgmap v10293282: 431 pgs: 1
active+undersized+degraded+remapped+backfilling, 28 active+undersized+degraded, 49
active+undersized+degraded+remapped+wait_backfill, 59 stale+active+clean, 294
active+clean; 72347 MB data, 101302 MB used, 1624 GB / 1722 GB avail; 227 kB/s rd, 1358
B/s wr, 12 op/s; 10626/35917 objects degraded (29.585%); 6757/35917 objects misplaced
(18.813%); 63500 kB/s, 15 objects/s recovering
2022-05-02 04:48:04.414397 mon.0 [INF] pgmap v10293283: 431 pgs: 2
active+undersized+degraded+remapped+backfilling, 75
active+undersized+degraded+remapped+wait_backfill, 59 stale+active+clean, 295
active+clean; 72347 MB data, 101398 MB used, 1623 GB / 1722 GB avail; 969 kB/s rd, 6778
B/s wr, 32 op/s; 10626/35917 objects degraded (29.585%); 10580/35917 objects misplaced
(29.457%); 125 MB/s, 31 objects/s recovering
2022-05-02 04:48:00.380063 osd.1 [INF] 0.6f starting backfill to osd.0 from (0'0,0'0] MAX to
2521'166639
2022-05-02 04:48:00.380139 osd.1 [INF] 0.48 starting backfill to osd.0 from (0'0,0'0] MAX to
2513'43079
2022-05-02 04:48:00.380260 osd.1 [INF] 0.d starting backfill to osd.0 from (0'0,0'0] MAX to
2513'136847
2022-05-02 04:48:00.380849 osd.1 [INF] 0.71 starting backfill to osd.0 from (0'0,0'0] MAX to
2331'28496
2022-05-02 04:48:00.381027 osd.1 [INF] 0.51 starting backfill to osd.0 from (0'0,0'0] MAX to
2513'87544
```

在迁移完成后，您应该可以看到放置组状态从 **active+clean** 变为 **active, some degraded objects**，最终变为 **active+clean**。

5. 停止 OSD：

语法

```
ceph orch daemon stop OSD_ID
```

示例

```
[ceph: root@host01 /]# ceph orch daemon stop osd.0
```

6. 从存储集群中移除 OSD：

   语法

   ```
   ceph orch osd rm OSD_ID --replace
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph orch osd rm 0 --replace
   ```

   OSD_ID 被保留。

替换物理驱动器

有关替换物理驱动器的详情，请查看硬件节点的文档。

1. 如果驱动器支持热插拔，请将失败的驱动器替换为新驱动器。
2. 如果驱动器不可热插拔并且节点包含多个 OSD，您可能需要关闭整个节点并替换物理驱动器。考虑防止集群回填。详情请参阅 *Red Hat Ceph Storage 故障排除指南中的停止和启动 重新平衡* 章节。
3. 当驱动器出现在 **/dev/** 目录下时，请注意驱动器路径。
4. 如果要手动添加 OSD，找到 OSD 驱动器并格式化磁盘。

将 OSD 添加到 Ceph 集群

1. 插入新驱动器后，您可以使用以下选项来部署 OSD：
   - 如果未设置 **--unmanaged** 参数，OSD 由 Ceph 编排器自动部署。

     示例

     ```
     [ceph: root@host01 /]# ceph orch apply osd --all-available-devices
     ```

   - 将 OSD 部署到所有可用的设备上，并将 **unmanaged** 参数设置为 **true**。

     示例

     ```
     [ceph: root@host01 /]# ceph orch apply osd --all-available-devices --unmanaged=true
     ```

   - 将 OSD 部署到特定的设备和主机上。

     示例

     ```
     [ceph: root@host01 /]# ceph orch daemon add osd host02:/dev/sdb
     ```

2. 确保 CRUSH 层次结构准确：

   示例

   ```
   [ceph: root@host01 /]# ceph osd tree
   ```

其它资源

- 请参阅 *Red Hat Ceph Storage Operations Guide* 中的 *Deploying Ceph OSDs on all available devices* 部分。
- 请参阅 *Red Hat Ceph Storage Operations Guide* 中的 *Deploying Ceph OSDs on specific devices and hosts* 部分。
- 请参阅 *Red Hat Ceph Storage Troubleshooting Guide* 中的 *Down OSDs* 部分。
- 请参阅 *Red Hat Ceph Storage 安装指南*。

5.4. 增加 PID 数量

如果您有一个包含了超过 12 个 Ceph OSD 的节点，则默认线程数 (PID 数) 可能不足，特别是在恢复期间。因此，一些 **ceph-osd** 守护进程可能会终止且无法再次启动。如果发生这种情况，请增加允许的最大线程数量。

流程

临时增加这个数字：

```
[root@mon ~]# sysctl -w kernel.pid.max=4194303
```

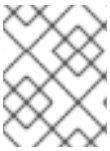
要永久增加数量，请更新 **/etc/sysctl.conf** 文件，如下所示：

```
kernel.pid.max = 4194303
```

5.5. 从完整存储集群中删除数据

Ceph 自动阻止 OSD 上任何达到 **mon_osd_full_ratio** 参数指定的容量的 I/O 操作，并返回 **full osds** 错误消息。

这个步骤演示了如何删除不必要的数据来修复这个错误。



注意

在创建集群时，**mon_osd_full_ratio** 参数会设置 **full_ratio** 参数的值。以后您将无法更改 **mon_osd_full_ratio** 的值。要临时增加 **full_ratio** 的值，请增加 **set-full-ratio** 的值。

先决条件

- Ceph 监控节点的根级别访问权限.

流程

1. 登录到 Cephadm shell：

   示例

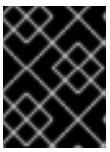
   ```
   [root@host01 ~]# cephadm shell
   ```

2. 确定 **full_ratio** 的当前值，其默认设置为 **0.95** ：

```
[ceph: root@host01 /]# ceph osd dump | grep -i full
full_ratio 0.95
```

3. 临时将 **set-full-ratio** 的值增加到 **0.97** ：

```
[ceph: root@host01 /]# ceph osd set-full-ratio 0.97
```



重要

红帽强烈建议不要将 **set-full-ratio** 设置为大于 0.97 的值。将此参数设置为更高的值会使恢复过程变得更加困难。因此，您可能无法完全恢复完整的 OSD。

4. 验证您是否成功将该参数设置为 **0.97** ：

```
[ceph: root@host01 /]# ceph osd dump | grep -i full
full_ratio 0.97
```

5. 监控集群状态：

```
[ceph: root@host01 /]# ceph -w
```

当集群的状态从 **full** 改为 **nearfull** 时，请删除任何不必要的数据。

6. 将 **full_ratio** 的值设置为 **0.95** ：

```
[ceph: root@host01 /]# ceph osd set-full-ratio 0.95
```

7. 验证您是否成功将该参数设置为 **0.95**:

```
[ceph: root@host01 /]# ceph osd dump | grep -i full
full_ratio 0.95
```

其它资源

- *Red Hat Ceph Storage 故障排除指南* 中的 *Full OSDs* 部分。
- *Red Hat Ceph Storage 故障排除指南中的 Nearfull OSDs* 部分。

第 6 章 对多站点 CEPH 对象网关进行故障排除

本章介绍了如何修复与多站点 Ceph 对象网关配置和操作条件相关的最常见的错误。

注意

当 **radosgw-admin bucket sync status** 命令报告存储桶位于分片（即使数据在多站点之间保持一致）时，对存储桶运行额外的写入。它同步状态报告，并显示存储桶从源中发现的消息。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 正在运行的 Ceph 对象网关.

6.1. CEPH 对象网关的错误代码定义

Ceph 对象网关日志包含错误和警告消息，以协助对环境中条件进行故障排除。下面列出了一些推荐的解析信息。

常见错误消息

data_sync: ERROR: a sync operation returned error

这是提示较低级别 bucket 同步进程返回错误的高级别数据同步过程。此消息冗余；存储桶同步错误出现在日志中。

data sync: ERROR: failed to sync object: *BUCKET_NAME*:_OBJECT_NAME_

进程无法通过 HTTP 从远程网关获取所需的对象，或者进程无法将该对象写入 RADOS，还会重试。

data sync: ERROR: failure in sync, backing out (sync_status=2)

一个低级别的信息反映了上述的一个条件，特别是数据在可以同步前会被删除，从而显示一个 **-2 ENOENT** 状态。

data sync: ERROR: failure in sync, backing out (sync_status=-5)

低级消息反映了上述其中一个条件，特别是我们无法将该对象写入 RADOS，从而显示 **-5 EIO**。

ERROR: failed to fetch remote data log information: ret=11

这是 **libcurl** 中的 **EAGAIN** 通用错误代码，反映了来自另一个网关的错误条件。默认情况下，它将重试。

meta sync: ERROR: failed to read mdlog info with (2) No such file or directory

mdlog 的分片从没有创建，因此不会同步。

同步错误消息

同步对象失败

进程无法通过 HTTP 从远程网关获取此对象，或者未能将该对象写入 RADOS，还会重试。

同步存储桶实例失败：(11)资源临时不可用

主区域和次要区域之间的连接问题。

同步存储桶实例失败：(125) Operation canceled

在写入同一 RADOS 对象之间存在追踪条件。

ERROR: request failed: (13) Permission denied 如果 master zone 上更改了域，则可能需要重启 master 区域的网关来识别此用户

在配置二级站点时，有时一个 **rgw realm pull --url http://primary_endpoint --access-key <> --secret <&** gt; 命令会失败，并显示 permission denied 错误。
在这种情况下，在主站点中运行以下命令，以确保系统用户凭证相同：

```
radosgw-admin user info --uid SYNCHRONIZATION_USER, and
radosgw-admin zone get
```

其它资源

- 联系红帽支持 以获取任何其他帮助。

6.2. 同步多站点 CEPH 对象网关

多站点同步从其他区域读取更改日志。要从元数据和数据日志中获取同步进度的高级视图，您可以使用以下命令：

示例

```
[ceph: root@host01 /]# radosgw-admin sync status
```

此命令列出源区域后面的日志分片（若有）。

注意

有时，在运行 **radosgw-admin sync status** 命令时可能会观察恢复分片。对于数据同步，每个复制日志都有 128 个分片。如果这些复制日志事件触发的任何操作导致网络、存储或其他位置的任何错误，则这些错误会被跟踪，以便操作可以稍后重试。虽然给定分片具有需要重试的错误，但 **radosgw-admin sync status** 命令将该分片报告为 **recovering**。此恢复会自动进行，因此操作器不需要干预来解决问题。

如果以上运行的同步状态的结果返回日志分片，运行以下命令，将 *X* 替换为实际的 shard-id。

多站点对象内的 bucket 也可以在 Ceph 仪表板上监控。有关更多信息，请参阅 *Red Hat Ceph Storage Dashboard 指南中的 监控多站点对象的存储桶*。

语法

```
radosgw-admin data sync status --shard-id=X --source-zone=ZONE_NAME
```

示例

```
[ceph: root@host01 /]# radosgw-admin data sync status --shard-id=27 --source-zone=us-east
{
  "shard_id": 27,
  "marker": {
        "status": "incremental-sync",
        "marker": "1_1534494893.816775_131867195.1",
        "next_step_marker": "",
        "total_entries": 1,
```

```
        "pos": 0,
        "timestamp": "0.000000"
    },
    "pending_buckets": [],
    "recovering_buckets": [
        "pro-registry:4ed07bb2-a80b-4c69-aa15-fdc17ae6f5f2.314303.1:26"
    ]
}
```

输出列出了同步旁边的存储桶，以及会因为前面的错误而重试哪些存储桶（若有）。

通过以下命令检查各个 bucket 的状态，使用 bucket ID 替换 *X*。

语法

```
radosgw-admin bucket sync status --bucket=X.
```

将 *X* 替换为存储桶的 ID 号。

结果显示哪些存储桶索引日志分片位于其源区后面。

同步中的一个常见错误是 **EBUSY**，这意味着同步已在进行中，通常在另一个网关上。读取写入到同步错误日志的错误，可以使用以下命令进行读取：

```
radosgw-admin sync error list
```

同步过程将重试，直到成功为止。仍可能会出现可能需要干预的错误。

6.3. 执行多站点 CEPH 对象网关的数据同步的计数器

以下性能计数器可用于 Ceph 对象网关的多站点配置来测量数据同步：

- **poll_latency** 测量远程复制日志的请求延迟。
- **fetch_bytes** 测量数据同步获取的对象和字节数。

使用 **ceph --admin-daemon** 命令查看性能计数器的当前指标数据：

语法

```
ceph --admin-daemon /var/run/ceph/ceph-client.rgw.RGW_ID.asok perf dump data-sync-from-ZONE_NAME
```

示例

```
[ceph: root@host01 /]# ceph --admin-daemon /var/run/ceph/ceph-client.rgw.host02-rgw0.103.94309060818504.asok perf dump data-sync-from-us-west

{
    "data-sync-from-us-west": {
        "fetch bytes": {
            "avgcount": 54,
            "sum": 54526039885
        },
```

```
        "fetch not modified": 7,
        "fetch errors": 0,
        "poll latency": {
            "avgcount": 41,
            "sum": 2.533653367,
            "avgtime": 0.061796423
        },
        "poll errors": 0
    }
}
```

注意

您必须从运行守护进程的节点运行 **ceph --admin-daemon** 命令。

其它资源

- 有关 *性能计数器* 的更多信息，请参阅 *Red Hat Ceph Storage Administration Guide* 中的 Ceph 性能计数器章节。

6.4. 在多站点 CEPH 对象网关配置中同步数据

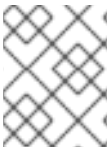
在存储集群的多站点 Ceph 对象网关配置中，故障转移和故障恢复会导致数据同步停止。**radosgw-admin sync status** 命令报告数据同步在延长时间内已落后。

您可以运行 **radosgw-admin data sync init** 命令，来同步站点之间的数据，然后重启 Ceph 对象网关。此命令不涉及任何实际对象数据，并启动指定源区的数据同步。这会导致区从源区重启完全同步。

重要

在运行 **data sync init** 命令前，请联系红帽支持。
https://access.redhat.com/support/contact/technicalSupport

如果您要完全重启同步，并且源区上需要同步很多数据，则必须相应地规划带宽消耗。



注意

如果用户意外删除次要站点上的存储桶，您可以使用站点上的 ***metadata sync init*** *命令同步数据。*

先决条件

- *一个正在运行的 Red Hat Ceph Storage 集群。*
- *Ceph 对象网关至少配置两个站点。*

流程

1. *检查站点之间的同步状态：*

   示例

   ```
   [ceph: host04 /]# radosgw-admin sync status
   ```

```
          realm d713eec8-6ec4-4f71-9eaf-379be18e551b (india)
      zonegroup ccf9e0b2-df95-4e0a-8933-3b17b64c52b7 (shared)
           zone 04daab24-5bbd-4c17-9cf5-b1981fd7ff79 (primary)
   current time 2022-09-15T06:53:52Z
zonegroup features enabled: resharding
  metadata sync no sync (zone is master)
      data sync source: 596319d2-4ffe-4977-ace1-8dd1790db9fb (secondary)
                        syncing
                        full sync: 0/128 shards
                        incremental sync: 128/128 shards
                        data is caught up with source
```

2. *同步来自 second zone 的数据：*

示例

```
[ceph: root@host04 /]# radosgw-admin data sync init --source-zone primary
```

3. *在站点重启所有 Ceph 对象网关守护进程：*

示例

```
[ceph: root@host04 /]# ceph orch restart rgw.myrgw
```

6.5. 升级集群后对 RADOSGW-ADMIN 命令进行故障排除

在升级集群后，使用 cephadm shell 中的 ***radosgw-admin*** *命令进行故障排除。*

以下是在升级集群后尝试在 cephadm shell 中运行 ***radosgw-admin*** *命令后可能发出的错误示例。*

```
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to decode obj from
.rgw.root:periods.91d2a42c-735b-492a-bcf3-05235ce888aa.3
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 failed reading current period info:  (5) Input/output
error
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to start notify service ((5)
Input/output error
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to init services (ret=(5) Input/output
error)
couldn't init storage provider
```

示例

```
[ceph: root@host01 /]# date;radosgw-admin bucket list
Mon May 13 09:05:30 UTC 2024
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to decode obj from
.rgw.root:periods.91d2a42c-735b-492a-bcf3-05235ce888aa.3
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 failed reading current period info:  (5) Input/output
error
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to start notify service ((5)
Input/output error
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to init services (ret=(5) Input/output
error)
couldn't init storage provider
```

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 对节点的根级别访问权限。

流程

- 使用 **radosgw-admin** 语法再次运行该命令，以修复其预期。

  语法

  ```
  cephadm shell --radosgw-admin COMMAND
  ```

  示例

  ```
  [root@host01 /]# cephadm shell -- radosgw-admin bucket list
  ```

第 7 章 CEPH 放置组故障排除

本节介绍修复与 Ceph 放置组(PG)相关的最常见错误。

先决条件

- 验证您的网络连接。
- 确保 monitor 能够形成仲裁。
- 确保所有健康的 OSD 为 **up** 和 **in**，回填和恢复过程已完成。

7.1. 最常见的 CEPH 放置组错误

下表列出了 **ceph health detail** 命令返回的最常见的错误消息。这些表中提供了相应部分的链接，这些部分解释了错误并指向修复问题的特定程序。

另外，您可以列出处于非最佳状态的放置组。详情请查看 第 7.2 节 "列出放置组处于 **过时**、**不活动** 或未 **清除** 状态"。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 正在运行的 Ceph 对象网关.

7.1.1. 放置组错误消息

常见放置组错误消息表以及潜在的修复。

错误消息	查看
HEALTH_ERR	
PG 停机	放置组处于 **down**状态
pgs inconsistent	Inconsistent placement groups
scrub errors	Inconsistent placement groups
HEALTH_WARN	
pgs stale	Stale placement groups
unfound	Unfound objects

7.1.2. Stale 放置组

ceph health 命令将一些放置组 (PG) 列为 **stale** ：

```
HEALTH_WARN 24 pgs stale; 3/300 in osds are down
```

这意味着

当放置组没有从 PG 的操作集合的 Primary OSD 接收任何状态更新，或者当其他 OSD 报告 Primary OSD 为 **down** 时，Monitor 会将该放置组标记为 **stale**。

通常，PG 在启动存储集群后进入 **stale** 状态，直到对等过程完成为止。但是，如果 PG 处于 **stale** 状态的时间超过预期，这可能表示这些 PG 的 Primary OSD 为 **down** 或未向 monitor 报告 PG 统计信息。当存储过时 PG 的 Primary OSD **备份** 时，Ceph 会开始恢复 PG。

mon_osd_report_timeout 设置决定了 OSD 向 monitor 报告 PG 统计的频率。默认情况下，此参数设置为 **0.5**，这意味着 OSD 每半年报告统计数据。

要排除此问题，请执行以下操作

1. 识别哪些 PG 处于 **stale** 状态，以及它们存储在哪些 OSD 上。错误消息包括类似以下示例的信息：

   示例

   ```
   [ceph: root@host01 /]# ceph health detail
   HEALTH_WARN 24 pgs stale; 3/300 in osds are down
   ...
   pg 2.5 is stuck stale+active+remapped, last acting [2,0]
   ...
   osd.10 is down since epoch 23, last address 192.168.106.220:6800/11080
   osd.11 is down since epoch 13, last address 192.168.106.220:6803/11539
   osd.12 is down since epoch 24, last address 192.168.106.220:6806/11861
   ```

2. 对标记为 **down** 的 OSD 的任何问题进行故障排除。详情请参阅 关闭 OSD。

其它资源

- Red Hat Ceph Storage 7 管理指南中的 监控放置组设置 部分

7.1.3. Inconsistent placement groups

有些放置组被标记为 **active + clean + inconsistent**，**ceph health detail** 会返回类似如下的错误消息：

```
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

这意味着

当 Ceph 检测到 PG 中一个或多个对象副本中的不一致时，它会将该 PG 标记为 **inconsistent**。最常见的不一致是：

- 对象的大小不正确。
- 恢复完成后，一个副本中的对象会丢失。

在大多数情况下，清理过程中的错误会在放置组内造成不一致。

要排除此问题，请执行以下操作

1. 登录到 Cephadm shell：

   示例

   ```
   [root@host01 ~]# cephadm shell
   ```

2. 确定哪个放置组处于**不一致的状态**：

   ```
   [ceph: root@host01 /]# ceph health detail
   HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
   pg 0.6 is active+clean+inconsistent, acting [0,1,2]
   2 scrub errors
   ```

3. 确定 PG **不一致**的原因。

   a. 在 PG 中启动深度清理过程：

      语法

      ```
      ceph pg deep-scrub ID
      ```

      使用 **inconsistent** 的 PG 的 **ID** 替换 ID，例如：

      ```
      [ceph: root@host01 /]# ceph pg deep-scrub 0.6
      instructing pg 0.6 on osd.0 to deep-scrub
      ```

   b. 搜索 **ceph -w** 的输出，以查找与该放置组相关的任何消息：

      语法

      ```
      ceph -w | grep ID
      ```

      使用 **inconsistent** 的 PG 的 **ID** 替换 ID，例如：

      ```
      [ceph: root@host01 /]# ceph -w | grep 0.6
      2022-05-26 01:35:36.778215 osd.106 [ERR] 0.6 deep-scrub stat mismatch, got 636/635
      objects, 0/0 clones, 0/0 dirty, 0/0 omap, 0/0 hit_set_archive, 0/0 whiteouts,
      1855455/1854371 bytes.
      2022-05-26 01:35:36.788334 osd.106 [ERR] 0.6 deep-scrub 1 errors
      ```

4. 如果输出包含与以下类似的错误消息，您可以修复**不一致的**放置组。详情请参阅 修复不一致的放置组。

   语法

   ```
   PG.ID shard OSD: soid OBJECT missing attr , missing attr _ATTRIBUTE_TYPE
   PG.ID shard OSD: soid OBJECT digest 0 != known digest DIGEST, size 0 != known size
   SIZE
   PG.ID shard OSD: soid OBJECT size 0 != known size SIZE
   ```

```
PG.ID deep-scrub stat mismatch, got MISMATCH
PG.ID shard OSD: soid OBJECT candidate had a read error, digest 0 != known digest
DIGEST
```

5. 如果输出包含与以下类似的错误消息，则无法安全地修复**不一致的** PG，因为您可以丢失数据。在这种情况下，创建一个支持问题单。详情请参阅联系红帽支持。

```
PG.ID shard OSD: soid OBJECT digest DIGEST != known digest DIGEST
PG.ID shard OSD: soid OBJECT omap_digest DIGEST != known omap_digest DIGEST
```

其它资源

- 请参阅 Red Hat Ceph Storage 故障排除指南中的 列出放置组不一致的情况。
- 请参阅 Red Hat Ceph Storage Architecture Guide 中的 Ceph data integrity 部分。
- 详情请参阅 Red Hat Ceph Storage 配置指南 中的清理 OSD 部分。

7.1.4. unclean PG

ceph health 命令返回类似如下的错误消息：

```
HEALTH_WARN 197 pgs stuck unclean
```

这意味着

如果 PG 未在 Ceph 配置文件中的 **mon_pg_stuck_threshold** 参数中指定的秒数达到 **active+clean** 状态，则 Ceph 会将其标记为 **unclean**。**mon_pg_stuck_threshold** 的默认值为 **300** 秒。

如果放置组为 **unclean**，它包含没有复制 **osd_pool_default_size** 参数中指定的次数的对象。**osd_pool_default_size** 的默认值为 **3**，这意味着 Ceph 创建三个副本。

通常，**unclean** 放置组代表某些 OSD 可能处于 **down**。

要排除此问题，请执行以下操作

1. 确定哪些 OSD 为 **down** ：

```
[ceph: root@host01 /]# ceph osd tree
```

2. 故障排除和修复与 OSD 相关的问题。详情请参阅关闭 OSD。

其它资源

- 列出放置组处于 stale inactive 或 unclean 状态。

7.1.5. 不活跃的放置组

ceph health 命令返回类似如下的错误消息：

```
HEALTH_WARN 197 pgs stuck inactive
```

这意味着

如果 PG 在 Ceph ***配置文件中的 mon_pg_stuck_threshold*** *参数中指定的秒数中未激活，Ceph 会将它标记为不活动。* ***mon_pg_stuck_threshold*** *的默认值为* ***300*** *秒。*

通常， ***inactive*** *放置组代表某些 OSD 可能处于* ***down***。

要排除此问题，请执行以下操作

1. *确定哪些 OSD 为* ***down*** *：*

```
# ceph osd tree
```

2. *故障排除和修复与 OSD 相关的问题。*

其它资源

- *列出放置组处于 stale inactive 或 unclean 状态*
- *详情请参阅关闭 OSD。*

7.1.6. 放置组处于 down 状态

ceph health detail *命令报告某些放置组* ***已停机*** *：*

```
HEALTH_ERR 7 pgs degraded; 12 pgs down; 12 pgs peering; 1 pgs recovering; 6 pgs stuck
unclean; 114/3300 degraded (3.455%); 1/3 in osds are down
...
pg 0.5 is down+peering
pg 1.4 is down+peering
...
osd.1 is down since epoch 69, last address 192.168.106.220:6801/8651
```

这意味着

在某些情况下，对等过程可能会被阻断，这会阻止放置组处于活跃状态并可用。通常，OSD 故障会导致对等失败。

要排除此问题，请执行以下操作

确定什么阻塞了对等进程：

语法

```
ceph pg ID query
```

将 ***ID*** *替换为* ***down*** *的 PG 的 ID：*

示例

```
[ceph: root@host01 /]#  ceph pg 0.5 query

{ "state": "down+peering",
  ...
  "recovery_state": [
      { "name": "Started\/Primary\/Peering\/GetInfo",
```

```
        "enter_time": "2021-08-06 14:40:16.169679",
        "requested_info_from": []},
      { "name": "Started\/Primary\/Peering",
        "enter_time": "2021-08-06 14:40:16.169659",
        "probing_osds": [
            0,
            1],
        "blocked": "peering is blocked due to down osds",
        "down_osds_we_would_probe": [
            1],
        "peering_blocked_by": [
            { "osd": 1,
              "current_lost_at": 0,
              "comment": "starting or marking this osd lost may let us proceed"}]},
      { "name": "Started",
        "enter_time": "2021-08-06 14:40:16.169513"}
    ]
}
```

recovery_state 部分包含有关为什么对等进程被阻止的信息。

- 如果输出包含 **peering is blocked due due due dueing osds** 错误消息，请参阅 Down OSDs。
- 如果您看到任何其他错误消息，请创建一个支持问题单。详情请参阅联系红帽支持服务。

其它资源

- Red Hat Ceph Storage Administration Guide 中的 Ceph OSD peering 部分。

7.1.7. Unfound objects

ceph health 命令返回一个类似于以下的错误消息，其中包含 **unfound** 关键字：

```
HEALTH_WARN 1 pgs degraded; 78/3778 unfound (2.065%)
```

这意味着

当知道这些对象或它们的较新副本存在但无法找到它们时，Ceph 会将其标记为 **unfound**。因此，Ceph 无法恢复这样的对象并继续恢复过程。

Situation 示例

放置组在 **osd.1** 和 **osd.2** 上存储数据。

1. **OSD.1 停机**。
2. **osd.2** 处理一些写入操作。
3. **OSD.1** 启动。
4. **osd.1** 和 **osd.2** 启动之间的对等进程，并且 **osd.1** 上缺少的对象被排队以进行恢复。
5. 在 Ceph 复制新对象之前，**osd.2** 会 **停机**。

因此，**osd.1** 知道这些对象存在，但没有 OSD 具有对象的副本。

*在这种情况下，Ceph 正在等待故障节点再次访问，而 **unfound** 对象会阻止恢复过程。*

要排除此问题，请执行以下操作

1. *登录到 Cephadm shell：*

   示例

   ```
   [root@host01 ~]# cephadm shell
   ```

2. *确定哪个放置组包含 **unfound** 对象：*

   ```
   [ceph: root@host01 /]# ceph health detail
   HEALTH_WARN 1 pgs recovering; 1 pgs stuck unclean; recovery 5/937611 objects
   degraded (0.001%); 1/312537 unfound (0.000%)
   pg 3.8a5 is stuck unclean for 803946.712780, current state active+recovering, last acting
   [320,248,0]
   pg 3.8a5 is active+recovering, acting [320,248,0], 1 unfound
   recovery 5/937611 objects degraded (0.001%); **1/312537 unfound (0.000%)**
   ```

3. *列出放置组的更多信息：*

   语法

   ```
   ceph pg ID query
   ```

   *使用包含 **unfound** 对象的放置组 **ID** 替换 ID：*

   示例

   ```
   [ceph: root@host01 /]# ceph pg 3.8a5 query
   { "state": "active+recovering",
     "epoch": 10741,
     "up": [
           320,
           248,
           0],
     "acting": [
           320,
           248,
           0],
   <snip>
     "recovery_state": [
           { "name": "Started\/Primary\/Active",
             "enter_time": "2021-08-28 19:30:12.058136",
             "might_have_unfound": [
                 { "osd": "0",
                   "status": "already probed"},
                 { "osd": "248",
                   "status": "already probed"},
                 { "osd": "301",
                   "status": "already probed"},
                 { "osd": "362",
                   "status": "already probed"},
   ```

```
            { "osd": "395",
              "status": "already probed"},
            { "osd": "429",
              "status": "osd is down"}],
        "recovery_progress": { "backfill_targets": [],
            "waiting_on_backfill": [],
            "last_backfill_started": "0\/\/0\/\/-1",
            "backfill_info": { "begin": "0\/\/0\/\/-1",
                "end": "0\/\/0\/\/-1",
                "objects": []},
            "peer_backfill_info": [],
            "backfills_in_flight": [],
            "recovering": [],
            "pg_backend": { "pull_from_peer": [],
                "pushing": []}},
        "scrub": { "scrubber.epoch_start": "0",
            "scrubber.active": 0,
            "scrubber.block_writes": 0,
            "scrubber.finalizing": 0,
            "scrubber.waiting_on": 0,
            "scrubber.waiting_on_whom": []}},
        { "name": "Started",
          "enter_time": "2021-08-28 19:30:11.044020"}],
```

may **_have_unfound** 部分包括 Ceph 试图定位 **unfound** 对象的 OSD：

- **已探测的** 状态表示 Ceph 无法找到该 OSD 中的 **unfound** 对象。
- **osd** 为 **down** 状态表示 Ceph 无法与该 OSD 联系。

4. 对标记为 **down** 的 OSD 进行故障排除。详情请参阅关闭 OSD。
5. 如果您无法修复导致 OSD **停机** 的问题，请创建一个支持问题单。详情请参阅联系红帽支持以获取服务。

7.2. 列出放置组处于 过时、不活动 或未 清除 状态

失败后，放置组进入状态，如 **degraded** 或 **peering**。这个状态表示故障恢复过程的正常进度。

但是，如果 PG 处于这些状态之一的时间比预期长，则可能代表更大的问题。监控器报告当放置组处于不最佳状态。

Ceph 配置文件中的 **mon_pg_stuck_threshold** 选项决定了放置组被视为 **不活动**、**unclean** 或 **stale** 的秒数。

下表列出了这些状态及简短的说明：

状态	它代表什么	最常见的原因	查看
inactive	PG 无法服务读/写请求。	• 对等问题	不活跃的放置组

状态	它代表什么	最常见的原因	查看
unclean	PG 包含没有复制所需次数的对象。某种阻止 PG 恢复。	• **unfound** 对象 • OSD 为 **down** • 配置不正确	unclean PG
stale	PG 的状态尚未由 **ceph-osd** 守护进程更新。	• OSD 为 **down**	Stale 放置组

先决条件

- *一个正在运行的 Red Hat Ceph Storage 集群。*
- *节点的根级别访问权限。*

流程

1. *登录到 Cephadm shell：*

   示例

   ```
   [root@host01 ~]# cephadm shell
   ```

2. *列出卡住 PG：*

   示例

   ```
   [ceph: root@host01 /]# ceph pg dump_stuck inactive
   [ceph: root@host01 /]# ceph pg dump_stuck unclean
   [ceph: root@host01 /]# ceph pg dump_stuck stale
   ```

其它资源

- *请参阅 Red Hat Ceph Storage Administration Guide 中的 Placement Group States 部分。*

7.3. 列出放置组不一致

*使用 **rados** 实用程序列出不同对象副本中的不一致。使用 **--format=json-pretty** 选项列出更详细的输出。*

本节涵盖以下列表：

- *池中的放置组不一致*
- *放置组中的对象不一致*
- *放置组中的快照集不一致*

先决条件

- 正在运行的 Red Hat Ceph Storage 集群处于健康状态。
- 节点的根级别访问权限。

流程

- 列出池中所有不一致的放置组：

  语法

  ```
  rados list-inconsistent-pg POOL --format=json-pretty
  ```

  示例

  ```
  [ceph: root@host01 /]# rados list-inconsistent-pg data --format=json-pretty
  [0.6]
  ```

- 使用 ID 列出 PG 中不一致的对象：

  语法

  ```
  rados list-inconsistent-obj PLACEMENT_GROUP_ID
  ```

  示例

  ```
  [ceph: root@host01 /]# rados list-inconsistent-obj 0.6
  {
      "epoch": 14,
      "inconsistents": [
          {
              "object": {
                  "name": "image1",
                  "nspace": "",
                  "locator": "",
                  "snap": "head",
                  "version": 1
              },
              "errors": [
                  "data_digest_mismatch",
                  "size_mismatch"
              ],
              "union_shard_errors": [
                  "data_digest_mismatch_oi",
                  "size_mismatch_oi"
              ],
              "selected_object_info": "0:602f83fe:::foo:head(16'1 client.4110.0:1
  dirty|data_digest|omap_digest s 968 uv 1 dd e978e67f od ffffffff alloc_hint [0 0 0])",
              "shards": [
                  {
                      "osd": 0,
                      "errors": [],
                      "size": 968,
  ```

```
            "omap_digest": "0xffffffff",
            "data_digest": "0xe978e67f"
        },
        {
            "osd": 1,
            "errors": [],
            "size": 968,
            "omap_digest": "0xffffffff",
            "data_digest": "0xe978e67f"
        },
        {
            "osd": 2,
            "errors": [
                "data_digest_mismatch_oi",
                "size_mismatch_oi"
            ],
            "size": 0,
            "omap_digest": "0xffffffff",
            "data_digest": "0xffffffff"
        }
    ]
  }
 ]
}
```

以下字段非常重要，以确定导致不一致的原因：

- ***name***：*副本不一致的对象名称。*
- ***nspace***：*池逻辑分割的命名空间。默认为空。*
- ***locator*** ：*这个键用于放置的替代对象名称。*
- ***snap*** ：*对象的快照 ID。对象的唯一可写版本被称为* ***head***。*如果对象是克隆，此字段包含其顺序 ID。*
- ***version*** ：*副本不一致的对象版本 ID。每个写入操作都会递增对象。*
- ***错误*** ：*指定分片或分片之间不一致的错误列表，而不确定哪个分片或分片不正确。请参阅* ***shard*** *数组以进一步调查错误。*
  - ***data_digest_mismatch*** ：*从一个 OSD 读取的副本摘要与其他 OSD 不同。*
  - ***size_mismatch*** ：*克隆的大小或* ***head*** *对象与预期不匹配。*
  - ***read_error*** ：*这个错误表示磁盘错误最有可能导致的不一致。*
- ***union_shard_error*** ：*所有特定于分片的错误的 union。这些错误连接到故障分片。以* ***oi*** *结尾的错误表示您必须将故障对象中的信息与选定对象的信息进行比较。请参阅* ***shard*** *数组以进一步调查错误。*
  在上例中，存储在 ***osd.2*** *上的对象副本的摘要与* ***osd.0*** *和* ***osd.1*** *上存储的副本不同。具体来说，副本摘要不是从* ***osd.2*** *读取的分片计算的* ***0xffffffff***，*但* ***0xe978e67f***。*此外，从* ***osd.2*** *读取的副本大小为 0，由* ***osd.0*** *和* ***osd.1*** *报告的大小为 968。*

- *列出快照集不一致：*

  语法

```
rados list-inconsistent-snapset PLACEMENT_GROUP_ID
```

示例

```
[ceph: root@host01 /]# rados list-inconsistent-snapset 0.23 --format=json-pretty
{
    "epoch": 64,
    "inconsistents": [
        {
            "name": "obj5",
            "nspace": "",
            "locator": "",
            "snap": "0x00000001",
            "headless": true
        },
        {
            "name": "obj5",
            "nspace": "",
            "locator": "",
            "snap": "0x00000002",
            "headless": true
        },
        {
            "name": "obj5",
            "nspace": "",
            "locator": "",
            "snap": "head",
            "ss_attr_missing": true,
            "extra_clones": true,
            "extra clones": [
                2,
                1
            ]
        }
    ]
```

该命令返回以下错误：

- **ss_attr_missing** ：缺少一个或多个属性。属性是快照集中编码为键值对列表的信息。
- **ss_attr_corrupted**: 一个或多个属性无法解码。
- **clone_missing** ：缺少克隆。
- **snapset_mismatch** ：快照集本身不一致。
- **head_mismatch** ：快照集表示 头 存在或不存在，但清理结果报告其他。
- **无头** ：缺少快照集的 头。
- **size_mismatch** ：克隆的大小或 **head** 对象与预期不匹配。

其它资源

- Red Hat Ceph Storage Troubleshooting Guide 中的 inconsistent placement groups 部分。

- Red Hat Ceph Storage Troubleshooting Guide 中的 repair ing inconsistent placement groups 部分。

7.4. 修复不一致的放置组

由于深度清理过程中出现错误，一些放置组可以包含不一致。Ceph 报告此类放置组 **不一致** ：

```
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

警告

您只能修复某些不一致。

如果 Ceph 日志包含以下错误，请不要修复放置组：

```
_PG_._ID_ shard _OSD_: soid _OBJECT_ digest _DIGEST_ != known digest _DIGEST_
_PG_._ID_ shard _OSD_: soid _OBJECT_ omap_digest _DIGEST_ != known omap_digest
_DIGEST_
```

创建一个支持问题单。详情请参阅联系红帽支持以获取服务。

先决条件

- Ceph 监控节点的根级别访问权限.

流程

- 修复 **不一致的** 放置组：

  语法

  ```
  ceph pg repair ID
  ```

  使用 **inconsistent** PG 的 **ID** 替换 ID。

其它资源

- 请参阅 Red Hat Ceph Storage Troubleshooting Guide 中的 Inconsistent placement groups 部分。
- 请参阅 Red Hat Ceph Storage 故障排除指南 中的 列出放置组不一致的部分。

7.5. 增加放置组

放置组(PG)计数不足，会影响 Ceph 集群和数据分布的性能。它是 ***nearfull osds*** *错误消息的主要原因之一。*

推荐的比率是每个 OSD 100 到 300 个 PG。当您向集群添加更多 OSD 时，此比率可能会降低。

pg_num *和* ***pgp_num*** *参数决定了 PG 数。这些参数为每个池配置，因此您必须单独调整每个池的 PG 数较低。*

重要

增加 PG 数是您可以在 Ceph 集群上执行的最密集型进程。如果没有以缓慢和方法方式执行，这个过程可能会对性能有严重影响。增加 ***pgp_num*** *后，您将无法停止或撤销该进程，您必须完成它。考虑在业务关键处理时间分配之外增加 PG 计数，并提醒所有关于潜在性能影响的客户端。如果集群处于* ***HEALTH_ERR*** *状态，请不要更改 PG 数。*

先决条件

- *正在运行的 Red Hat Ceph Storage 集群处于健康状态。*
- *节点的根级别访问权限。*

流程

1. *减少数据重新发布和恢复在单个 OSD 和 OSD 主机上的影响：*
   a. *降低* ***osd max backfills***、***osd_recovery_max_active*** *和* ***osd_recovery_op_priority*** *参数的值：*

```
[ceph: root@host01 /]# ceph tell osd.* injectargs '--osd_max_backfills 1 --
osd_recovery_max_active 1 --osd_recovery_op_priority 1'
```

   b. *禁用浅刮除和深度刮除：*

```
[ceph: root@host01 /]# ceph osd set noscrub
[ceph: root@host01 /]# ceph osd set nodeep-scrub
```

2. *使用 Ceph Placement Groups (PGs) per Pool Calculator 来计算* ***pg_num*** *和* ***pgp_num*** *参数的最佳值。*
3. *以较小增量增加* ***pg_num*** *值，直到您达到所需的值。*
   a. *确定起始递增值。使用一个非常低的值（2 的指数），并在您确定对集群的影响时增加这个值。最佳值取决于池大小、OSD 数和客户端 I/O 负载。*
   b. *递增* ***pg_num*** *值：*

   语法

```
ceph osd pool set POOL pg_num VALUE
```

   指定池名称和新值，例如：

   示例

```
[ceph: root@host01 /]# ceph osd pool set data pg_num 4
```

-

c. *监控集群的状态：*

示例

```
[ceph: root@host01 /]# ceph -s
```

PG 状态将从 ***creating*** *变为* ***active+clean****。等待所有 PG 都处于* ***active+clean*** *状态。*

4. *以较小的增量增加* ***pgp_num*** *值，直到您达到所需的值：*

a. *确定起始递增值。使用一个非常低的值（2 的指数），并在您确定对集群的影响时增加这个值。最佳值取决于池大小、OSD 数和客户端 I/O 负载。*

b. *递增* ***pgp_num*** *值：*

语法

```
ceph osd pool set POOL pgp_num VALUE
```

指定池名称和新值，例如：

```
[ceph: root@host01 /]# ceph osd pool set data pgp_num 4
```

c. *监控集群的状态：*

```
[ceph: root@host01 /]# ceph -s
```

PG 状态将通过 ***peering****、* ***wait_backfill****、* ***回填****、* ***恢复*** *及其他状态进行更改。等待所有 PG 都处于* ***active+clean*** *状态。*

5. *对 PG 数量不足的所有池重复前面的步骤。*

6. *将* ***osd max backfills****、* ***osd_recovery_max_active*** *和* ***osd_recovery_op_priority*** *设置为其默认值：*

```
[ceph: root@host01 /]# ceph tell osd.* injectargs '--osd_max_backfills 1 --
osd_recovery_max_active 3 --osd_recovery_op_priority 3'
```

7. *启用浅刮除和深度刮除：*

```
[ceph: root@host01 /]# ceph osd unset noscrub
[ceph: root@host01 /]# ceph osd unset nodeep-scrub
```

其它资源

- *请参阅 Nearfull OSD*
- *请参阅 Red Hat Ceph Storage Administration Guide 中的 Monitoring Placement Group Sets 部分。*
- *详情请查看 第 3 章 网络问题故障排除。*

- 有关与 Ceph 监控器相关的最常见错误故障排除的详细信息，请参阅 第 4 章 Ceph 监控器故障排除。
- 有关对 Ceph OSD 相关的最常见的错误进行故障排除的详细信息，请参阅 第 5 章 Ceph OSD 故障排除。
- 如需有关 PG 自动缩放器的更多信息，请参阅 Red Hat Ceph Storage 策略指南中的 自动扩展放置组 部分。

第 8 章 CEPH 对象故障排除

作为存储管理员，您可以使用 **ceph-objectstore-tool** 程序执行高级别或低级对象操作。**ceph-objectstore-tool** 实用程序可帮助您排除与特定 OSD 或放置组中的对象相关的问题。

重要

操作对象可能会导致无法恢复的数据丢失。在使用 **ceph-objectstore-tool** 实用程序前，请联系红帽支持。

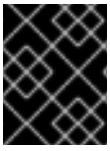
先决条件

- 验证没有与网络相关的问题。

8.1. 高级对象操作故障排除

作为存储管理员，您可以使用 **ceph-objectstore-tool** 程序执行高级别对象操作。**ceph-objectstore-tool** 实用程序支持以下高级别对象操作：

- 列出对象
- 列出丢失的对象
- 修复丢失的对象



重要

操作对象可能会导致无法恢复的数据丢失。在使用 **ceph-objectstore-tool** 实用程序前，请联系红帽支持。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。

8.1.1. 列出对象

OSD 可以包含零个到多个 PG 的 PG，对放置组(PG)中的多个对象包含零。**ceph-objectstore-tool** 实用程序允许您列出 OSD 中存储的对象。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。
- 停止 **ceph-osd** 守护进程。

流程

1. 验证适当的 OSD 是否为 down：

   语法

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

示例

```
[root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@osd.0.service
```

2. 登录到 OSD 容器：

语法

```
cephadm shell --name osd.OSD_ID
```

示例

```
[root@host01 ~]# cephadm shell --name osd.0
```

3. 识别 OSD 中的所有对象，而不考虑其放置组：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op list
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op list
```

4. 识别放置组中的所有对象：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID --op list
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c
--op list
```

5. 识别对象所属的 PG：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op list OBJECT_ID
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op list
default.region
```

8.1.2. 修复丢失的对象

*您可以使用 **ceph-objectstore-tool** 实用程序列出和修复 Ceph OSD 中存储的 丢失和未找到 的对象。此流程只适用于旧的对象。*

先决条件

- *对 Ceph OSD 节点的 root 级别访问权限。*
- *停止 **ceph-osd** 守护进程。*

流程

1. *验证适当的 OSD 是否为 down：*

   语法

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-001a4a0001df@osd.0.service
   ```

2. *登录到 OSD 容器：*

   语法

   ```
   cephadm shell --name osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. *列出所有丢失的传统对象：*

   语法

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost --dry-run
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op fix-lost --dry-run
   ```

4. *使用 **ceph-objectstore-tool** 实用程序修复 lost 和 unfound 对象。选择适当的情况：*

   a. *修复所有丢失的对象：*

      语法

      ```
      ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost
      ```

      示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op
fix-lost
```

b. *修复放置组中的所有丢失对象：*

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID --op fix-lost
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid
0.1c --op fix-lost
```

c. *按标识符修复丢失的对象：*

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost OBJECT_ID
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op
fix-lost default.region
```

8.2. 低级对象操作故障排除

作为存储管理员，您可以使用 ***ceph-objectstore-tool*** *程序执行低级对象操作。* ***ceph-objectstore-tool*** *实用程序支持以下低级别对象操作：*

- *操作对象的内容*
- *删除对象*
- *列出对象映射(OMAP)*
- *处理 OMAP 标头*
- *操作 OMAP 密钥*
- *列出对象的属性*
- *操作对象的属性键*

重要

操作对象可能会导致无法恢复的数据丢失。在使用 ***ceph-objectstore-tool*** *实用程序前，请联系红帽支持。*

先决条件

- *对 Ceph OSD 节点的 root 级别访问权限。*

8.2.1. 操作对象的内容

使用 **ceph-objectstore-tool** 实用程序，您可以在对象上获取或设置字节。

重要

在对象上设置字节可能会导致无法恢复的数据丢失。为防止数据丢失，请对对象进行备份副本。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。
- 停止 **ceph-osd** 守护进程。

流程

1. 验证适当的 OSD 是否为 down：

   语法

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

2. 通过列出 OSD 或放置组(PG)的对象来查找对象。

3. 登录到 OSD 容器：

   语法

   ```
   cephadm shell --name osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

4. 在对象中设置字节前，请进行备份和对象的工作副本：

   语法

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
   OBJECT \
   get-bytes > OBJECT_FILE_NAME
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c
   \
   ```

```
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}'  \
get-bytes > zone_info.default.backup

[ceph: root@host01 /]#  ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c
\
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}'  \
get-bytes > zone_info.default.working-copy
```

5. 编辑工作复制对象文件，并相应地修改对象内容。

6. 设置对象的字节：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
set-bytes < OBJECT_FILE_NAME
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c
\
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
set-bytes < zone_info.default.working-copy
```

8.2.2. 删除对象

使用 **ceph-objectstore-tool** 实用程序删除对象。通过移除对象，其内容和引用将从 PG 中删除。

重要

对象被删除后，您就无法重新创建对象。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。
- 停止 **ceph-osd** 守护进程。

流程

1. 登录到 OSD 容器：

语法

```
cephadm shell --name osd.OSD_ID
```

示例

```
[root@host01 ~]# cephadm shell --name osd.0
```

2. 删除对象：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
remove
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c
\
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
remove
```

8.2.3. 列出对象映射

使用 **ceph-objectstore-tool** 实用程序列出对象映射(OMAP)的内容。输出为您提供了键列表。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。
- 停止 **ceph-osd** 守护进程。

流程

1. 验证适当的 OSD 是否为 down：

语法

```
systemctl status ceph-osd@OSD_ID
```

示例

```
[root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@osd.0.service
```

2. 登录到 OSD 容器：

语法

```
cephadm shell --name osd.OSD_ID
```

示例

```
[root@host01 ~]# cephadm shell --name osd.0
```

3. 列出对象映射：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
list-omap
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c
\
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
list-omap
```

8.2.4. 操作对象映射标头

ceph-objectstore-tool 实用程序使用与对象键关联的值输出对象映射(OMAP)标头。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。
- 停止 **ceph-osd** 守护进程。

流程

1. 验证适当的 OSD 是否为 down：

   语法

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

2. 登录到 OSD 容器：

   语法

   ```
   cephadm shell --name osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. 获取对象映射标头：

   语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-omaphdr > OBJECT_MAP_FILE_NAME
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
get-omaphdr > zone_info.default.omaphdr.txt
```

4. 设置对象映射标头：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-omaphdr < OBJECT_MAP_FILE_NAME
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
set-omaphdr < zone_info.default.omaphdr.txt
```

8.2.5. 操作对象映射键

使用 **ceph-objectstore-tool** 程序更改对象映射(OMAP)密钥。您需要提供数据路径、放置组标识符(PG ID)、对象和 OMAP 中的密钥。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。
- 停止 **ceph-osd** 守护进程。

流程

1. 登录到 OSD 容器：

语法

```
cephadm shell --name osd.OSD_ID
```

示例

```
[root@host01 ~]# cephadm shell --name osd.0
```

2. 获取对象映射键：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-omap KEY > OBJECT_MAP_FILE_NAME
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
get-omap "" > zone_info.default.omap.txt
```

3. 设置对象映射键：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
set-omap KEY < OBJECT_MAP_FILE_NAME
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
set-omap "" < zone_info.default.omap.txt
```

4. 删除对象映射键：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
rm-omap KEY
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
rm-omap ""
```

8.2.6. 列出对象的属性

使用 **ceph-objectstore-tool** 实用程序列出对象的属性。输出为您提供对象的键和值。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。
- 停止 **ceph-osd** 守护进程。

流程

1. 验证适当的 OSD 是否为 down：

   语法

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

2. 登录到 OSD 容器：

   语法

   ```
   cephadm shell --name osd.OSD_ID
   ```

   示例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. 列出对象的属性：

   语法

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   list-attrs
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
   --pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
   2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
   list-attrs
   ```

8.2.7. 操作对象属性键

使用 **ceph-objectstore-tool** 程序更改对象的属性。要操作对象的属性，您需要数据路径、放置组标识符 (PG ID)、对象和对象属性中的键。

先决条件

- 对 Ceph OSD 节点的 root 级别访问权限。
- 停止 **ceph-osd** 守护进程。

流程

1. 验证适当的 OSD 是否为 down：

语法

```
systemctl status ceph-FSID@osd.OSD_ID
```

示例

```
[root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@osd.0.service
```

2. 登录到 OSD 容器：

语法

```
cephadm shell --name osd.OSD_ID
```

示例

```
[root@host01 ~]# cephadm shell --name osd.0
```

3. 获取对象的属性：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-attr KEY > OBJECT_ATTRS_FILE_NAME
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
get-attr "oid" > zone_info.default.attr.txt
```

4. 设置对象的属性：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
set-attr KEY < OBJECT_ATTRS_FILE_NAME
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
set-attr "oid"<zone_info.default.attr.txt
```

5. 删除对象的属性：

语法

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT  \
rm-attr KEY
```

示例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
rm-attr "oid"
```

其它资源

- *有关 Red Hat Ceph Storage 支持，请查看 红帽客户门户网站。*

第 9 章 在扩展模式下对集群进行故障排除

您可以替换和移除失败的 tiebreaker monitor。您还可以强制集群进入恢复或健康模式。

其它资源

有关扩展模式 集群的更多信息，请参阅 Ceph 存储的 Stretch 集群。

9.1. 使用仲裁中的 MONITOR 替换 TIEBREAKER

如果您的 tiebreaker monitor 失败，您可以将其替换为仲裁中的现有 monitor，并将它从集群中移除。

先决条件

- 正在运行的 Red Hat Ceph Storage 集群
- 在集群中启用扩展模式

流程

1. 禁用自动监控器部署：

   示例

   ```
   [ceph: root@host01 /]# ceph orch apply mon --unmanaged

   Scheduled mon update...
   ```

2. 查看仲裁中的监控器：

   示例

   ```
   [ceph: root@host01 /]# ceph -s

   mon: 5 daemons, quorum host01, host02, host04, host05 (age 30s), out of quorum: host07
   ```

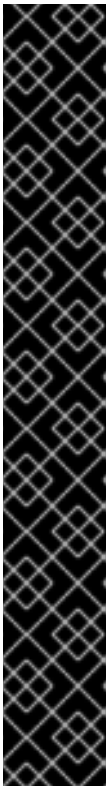
3. 将仲裁中的监控器设置为一个新的 tiebreaker：

   语法

   ```
   ceph mon set_new_tiebreaker NEW_HOST
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph mon set_new_tiebreaker host02
   ```



重要

如果 monitor 与现有非tiebreaker 监视器位于同一个位置，您会收到错误消息：

示例

```
[ceph: root@host01 /]# ceph mon set_new_tiebreaker host02

Error EINVAL: mon.host02 has location DC1, which matches mons host02 on
the datacenter dividing bucket for stretch mode.
```

如果发生这种情况，请更改 monitor 的位置：

语法

```
ceph mon set_location HOST datacenter=DATACENTER
```

示例

```
[ceph: root@host01 /]# ceph mon set_location host02 datacenter=DC3
```

4. 删除失败的 tiebreaker 监控器：

语法

```
ceph orch daemon rm FAILED_TIEBREAKER_MONITOR --force
```

示例

```
[ceph: root@host01 /]# ceph orch daemon rm mon.host07 --force

Removed mon.host07 from host 'host07'
```

5. 从主机中删除 monitor 后，重新部署 monitor：

语法

```
ceph mon add HOST IP_ADDRESS datacenter=DATACENTER
ceph orch daemon add mon HOST
```

示例

```
[ceph: root@host01 /]# ceph mon add host07 213.222.226.50 datacenter=DC1
[ceph: root@host01 /]# ceph orch daemon add mon host07
```

6. 确保仲裁中有五个监控器：

示例

```
[ceph: root@host01 /]# ceph -s

mon: 5 daemons, quorum host01, host02, host04, host05, host07 (age 15s)
```

7. 验证一切是否已正确配置：

示例

```
[ceph: root@host01 /]# ceph mon dump

epoch 19
fsid 1234ab78-1234-11ed-b1b1-de456ef0a89d
last_changed 2023-01-17T04:12:05.709475+0000
created 2023-01-16T05:47:25.631684+0000
min_mon_release 16 (pacific)
election_strategy: 3
stretch_mode_enabled 1
tiebreaker_mon host02
disallowed_leaders host02
0: [v2:132.224.169.63:3300/0,v1:132.224.169.63:6789/0] mon.host02; crush_location
{datacenter=DC3}
1: [v2:220.141.179.34:3300/0,v1:220.141.179.34:6789/0] mon.host04; crush_location
{datacenter=DC2}
2: [v2:40.90.220.224:3300/0,v1:40.90.220.224:6789/0] mon.host01; crush_location
{datacenter=DC1}
3: [v2:60.140.141.144:3300/0,v1:60.140.141.144:6789/0] mon.host07; crush_location
{datacenter=DC1}
4: [v2:186.184.61.92:3300/0,v1:186.184.61.92:6789/0] mon.host03; crush_location
{datacenter=DC2}
dumped monmap epoch 19
```

8. 重新部署 monitor：

语法

```
ceph orch apply mon --placement="HOST_1, HOST_2, HOST_3, HOST_4, HOST_5”
```

示例

```
[ceph: root@host01 /]# ceph orch apply mon --placement="host01, host02, host04, host05,
host07"

Scheduled mon update...
```

9.2. 将 TIEBREAKER 替换为新监控器

如果您的 tiebreaker monitor 失败，您可以将其替换为新的 monitor，并将它从集群中移除。

先决条件

- 正在运行的 Red Hat Ceph Storage 集群
- 在集群中启用的扩展模式

流程

1. 在集群中添加新监控器：

   a. 手动将 **crush_location** 添加到新监控器：

   语法

   ```
   ceph mon add NEW_HOST IP_ADDRESS datacenter=DATACENTER
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph mon add host06 213.222.226.50 datacenter=DC3

   adding mon.host06 at [v2:213.222.226.50:3300/0,v1:213.222.226.50:6789/0]
   ```

   注意

   新监控器必须与现有非tiebreaker 监视器不同的位置。

   b. 禁用自动监控器部署：

   示例

   ```
   [ceph: root@host01 /]# ceph orch apply mon --unmanaged

   Scheduled mon update…
   ```

   c. 部署新监控器：

   语法

   ```
   ceph orch daemon add mon NEW_HOST
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph orch daemon add mon host06
   ```

2. 确定有 6 个监控器，其中 5 个位于仲裁中：

   示例

   ```
   [ceph: root@host01 /]# ceph -s

   mon: 6 daemons, quorum host01, host02, host04, host05, host06 (age 30s), out of quorum: host07
   ```

3. 将新 monitor 设置为一个新的 tiebreaker：

   语法

   ```
   ceph mon set_new_tiebreaker NEW_HOST
   ```

示例

```
[ceph: root@host01 /]# ceph mon set_new_tiebreaker host06
```

4. *删除失败的 tiebreaker 监控器：*

语法

```
ceph orch daemon rm FAILED_TIEBREAKER_MONITOR --force
```

示例

```
[ceph: root@host01 /]# ceph orch daemon rm mon.host07 --force

Removed mon.host07 from host 'host07'
```

5. *验证一切是否已正确配置：*

示例

```
[ceph: root@host01 /]# ceph mon dump

epoch 19
fsid 1234ab78-1234-11ed-b1b1-de456ef0a89d
last_changed 2023-01-17T04:12:05.709475+0000
created 2023-01-16T05:47:25.631684+0000
min_mon_release 16 (pacific)
election_strategy: 3
stretch_mode_enabled 1
tiebreaker_mon host06
disallowed_leaders host06
0: [v2:213.222.226.50:3300/0,v1:213.222.226.50:6789/0] mon.host06; crush_location
{datacenter=DC3}
1: [v2:220.141.179.34:3300/0,v1:220.141.179.34:6789/0] mon.host04; crush_location
{datacenter=DC2}
2: [v2:40.90.220.224:3300/0,v1:40.90.220.224:6789/0] mon.host01; crush_location
{datacenter=DC1}
3: [v2:60.140.141.144:3300/0,v1:60.140.141.144:6789/0] mon.host02; crush_location
{datacenter=DC1}
4: [v2:186.184.61.92:3300/0,v1:186.184.61.92:6789/0] mon.host05; crush_location
{datacenter=DC2}
dumped monmap epoch 19
```

6. *重新部署 monitor：*

语法

```
ceph orch apply mon --placement="HOST_1, HOST_2, HOST_3, HOST_4, HOST_5”
```

示例

```
[ceph: root@host01 /]# ceph orch apply mon --placement="host01, host02, host04, host05,
host06"
```

```
Scheduled mon update…
```

9.3. 强制扩展集群恢复或健康模式

当处于扩展降级模式时，集群会在断开连接的数据中心返回后自动进入恢复模式。如果没有发生这种情况，或者您想要提前启用恢复模式，您可以强制扩展集群进入恢复模式。

先决条件

- 正在运行的 Red Hat Ceph Storage 集群
- 在集群中启用的扩展模式

流程

1. 强制扩展集群进入恢复模式：

   示例

   ```
   [ceph: root@host01 /]#  ceph osd force_recovery_stretch_mode --yes-i-really-mean-it
   ```

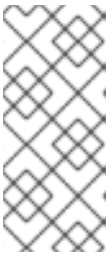
   注意

   恢复状态使集群处于 **HEALTH_WARN** 状态。

2. 在恢复模式中，集群应在放置组健康后返回到正常的扩展模式。如果没有发生这种情况，您可以将扩展集群强制变为健康模式：

   示例

   ```
   [ceph: root@host01 /]#  ceph osd force_healthy_stretch_mode --yes-i-really-mean-it
   ```

   注意

   如果您想在早期强制跨数据中心对等功能，您也可以运行此命令，且您希望风险数据停机，或者您单独验证所有放置组都可以对等，即使它们没有被完全恢复。

   您可能还希望调用健康模式来删除 **HEALTH_WARN** 状态，该状态由恢复状态生成。

   注意

   不需要 **force_recovery_stretch_mode** 和 **force_recovery_healthy_mode** 命令，因为它们包含在管理未预期的情况下的过程。

第 10 章 联系红帽支持以获取服务

如果本指南中的信息没有帮助您解决问题，本章将向您阐述如何联系 Red Hat 支持服务。

先决条件

- 红帽支持帐户。

10.1. 向红帽支持工程师提供信息

如果您无法修复与 Red Hat Ceph Storage 相关的问题，请联络红帽支持服务并提供足够数量的信息，以帮助支持工程师更快地解决遇到的问题。

先决条件

- 节点的根级别访问权限。
- 红帽支持帐户。

流程

1. 在红帽客户门户网站中创建一个支持问题单。
2. 理想情况下，将 **sosreport** 附加到票据。详情请查看 sosreport 是什么以及如何在 Red Hat Enterprise Linux 中创建？
3. 如果 Ceph 守护进程失败并显示分段错误，请考虑生成人类可读的核心转储文件。详情请参阅 生成可读内核转储文件。

10.2. 生成可读内核转储文件

当 Ceph 守护进程意外终止分段错误时，请收集关于其故障的信息，并将其提供给红帽支持工程师。

这些信息可加快初始调查速度。另外，支持工程师可将内核转储文件中的信息与 Red Hat Ceph Storage 集群进行比较。

先决条件

1. 安装 debuginfo 软件包（如果尚未安装）。
   a. 启用以下软件仓库来安装所需的 debuginfo 软件包。

   示例

   ```
   [root@host01 ~]# subscription-manager repos --enable=rhceph-6-tools-for-rhel-9-x86_64-rpms
   [root@host01 ~]# yum --enable=rhceph-6-tools-for-rhel-9-x86_64-debug-rpms
   ```

   启用软件仓库后，您可以从这个支持的软件包列表中安装您需要的 debug info 软件包：

   ```
   ceph-base-debuginfo
   ceph-common-debuginfo
   ceph-debugsource
   ```

```
ceph-fuse-debuginfo
ceph-immutable-object-cache-debuginfo
ceph-mds-debuginfo
ceph-mgr-debuginfo
ceph-mon-debuginfo
ceph-osd-debuginfo
ceph-radosgw-debuginfo
cephfs-mirror-debuginfo
```

2. 确保 **gdb** 软件包已安装，如果没有安装，请安装它：

示例

```
[root@host01 ~]# dnf install gdb
```

- 第 10.2.1 节 “在容器化部署中生成可读的内核转储文件”

10.2.1. 在容器化部署中生成可读的内核转储文件

您可以为 Red Hat Ceph Storage 生成内核转储文件，它涉及捕获内核转储文件的两个场景：

- 当 Ceph 进程因为 SIGILL、SIGTRAP、SIGABRT 或 SIGSEGV 错误而意外终止时。

或

- 手动，例如，用于调试 Ceph 进程等问题会消耗大量 CPU 周期，或者没有响应。

先决条件

- 对运行 Ceph 容器的容器节点的根级别访问权限。
- 安装适当的调试软件包。
- 安装 GNU Project Debugger (**gdb**) 软件包。
- 确保主机至少有 8 GB RAM。如果主机上有多个守护进程，红帽建议更多 RAM。

流程

1. 如果 Ceph 进程因为 SIGILL、SIGTRAP、SIGABRT 或 SIGSEGV 错误而意外终止：

   a. 在运行有故障 Ceph 进程的容器的节点上，将核心模式设置为 **systemd-coredump** 服务：

   示例

   ```
   [root@mon]# echo "| /usr/lib/systemd/systemd-coredump %P %u %g %s %t %c %h %e"
   > /proc/sys/kernel/core_pattern
   ```

   b. 观察因为 Ceph 进程导致下一个容器失败，并在 **/var/lib/systemd/coredump/** 目录中搜索核心转储文件：

   示例

   ```
   [root@mon]# ls -ltr /var/lib/systemd/coredump
   ```

```
total 8232
-rw-r-----. 1 root root 8427548 Jan 22 19:24 core.ceph-
osd.167.5ede29340b6c4fe4845147f847514c12.15622.1584573794000000.xz
```

2. 为 **Ceph Monitors** 和 **Ceph OSDs** 手动捕获内核转储文件：

   a. 获取 MONITOR_ID 或 OSD_ID 并输入容器：

   语法

   ```
   podman ps
   podman exec -it MONITOR_ID_OR_OSD_ID bash
   ```

   示例

   ```
   [root@host01 ~]# podman ps
   [root@host01 ~]# podman exec -it ceph-1ca9f6a8-d036-11ec-8263-fa163ee967ad-osd-2
   bash
   ```

   b. 在容器中安装 **procps-ng** 和 **gdb** 软件包：

   示例

   ```
   [root@host01 ~]# dnf install procps-ng gdb
   ```

   c. 查找进程 ID：

   语法

   ```
   ps -aef | grep PROCESS | grep -v run
   ```

   将 PROCESS 替换为正在运行的进程的名称，如 **ceph-mon** 或 **ceph-osd**。

   示例

   ```
   [root@host01 ~]# ps -aef | grep ceph-mon | grep -v run
   ceph       15390   15266  0 18:54 ?        00:00:29 /usr/bin/ceph-mon --cluster ceph --
   setroot ceph --setgroup ceph -d -i 5
   ceph       18110   17985  1 19:40 ?        00:00:08 /usr/bin/ceph-mon --cluster ceph --
   setroot ceph --setgroup ceph -d -i 2
   ```

   d. 生成内核转储文件：

   语法

   ```
   gcore ID
   ```

   使用您在上一步中获得的进程 ID 替换 ID，例如 **18110** ：

   示例

```
[root@host01 ~]# gcore 18110
warning: target file /proc/18110/cmdline contained unexpected null characters
Saved corefile core.18110
```

e. 验证核心转储文件是否已正确生成。

示例

```
[root@host01 ~]# ls -ltr
total 709772
-rw-r--r--. 1 root root 726799544 Mar 18 19:46 core.18110
```

f. 在 Ceph 监控容器外部复制内核转储文件：

语法

```
podman cp ceph-mon-MONITOR_ID:/tmp/mon.core.MONITOR_PID /tmp
```

将 MONITOR_ID 替换为 Ceph Monitor 的 ID 号，并将 MONITOR_PID 替换为进程 ID 号。

3. 为其他 Ceph 守护进程手动捕获内核转储文件：

a. 登录 **cephadm shell** ：

示例

```
[root@host03 ~]# cephadm shell
```

b. 为守护进程启用 **ptrace** ：

示例

```
[ceph: root@host01 /]# ceph config set mgr mgr/cephadm/allow_ptrace true
```

c. 重新部署守护进程服务：

语法

```
ceph orch redeploy SERVICE_ID
```

示例

```
[ceph: root@host01 /]# ceph orch redeploy mgr
[ceph: root@host01 /]# ceph orch redeploy rgw.rgw.1
```

d. 退出 **cephadm shell**，并登录到部署守护进程的主机：

示例

```
[ceph: root@host01 /]# exit
[root@host01 ~]# ssh root@10.0.0.11
```

e. *获取 DAEMON_ID 并输入容器：*

示例

```
[root@host04 ~]# podman ps
[root@host04 ~]# podman exec -it ceph-1ca9f6a8-d036-11ec-8263-fa163ee967ad-rgw-rgw-1-host04 bash
```

f. *安装 **procps-ng** 和 **gdb** 软件包：*

示例

```
[root@host04 /]# dnf install procps-ng gdb
```

g. *获取进程的 PID：*

示例

```
[root@host04 /]# ps aux | grep rados
ceph           6  0.3  2.8 5334140 109052 ?      Sl   May10   5:25 /usr/bin/radosgw -n client.rgw.rgw.1.host04 -f --setuser ceph --setgroup ceph --default-log-to-file=false --default-log-to-stderr=true --default-log-stderr-prefix=debug
```

h. *收集内核转储：*

语法

```
gcore PID
```

示例

```
[root@host04 /]# gcore 6
```

i. *验证核心转储文件是否已正确生成。*

示例

```
[root@host04 /]# ls -ltr
total 108798
-rw-r--r--. 1 root root 726799544 Mar 18 19:46 core.6
```

j. *在容器外复制内核转储文件：*

语法

```
podman cp ceph-mon-DAEMON_ID:/tmp/mon.core.PID /tmp
```

将 DAEMON_ID 替换为 Ceph 守护进程的 ID 号，并将 PID 替换为进程 ID 号。

4. *将核心转储文件上传至红帽支持问题单中。有关详细信息，请参阅向红帽支持工程师提供信息。*

其它资源

- *如何使用 gdb 从红帽客户门户网站上的应用程序核心解决方案生成可读的回溯追踪*
- *当应用程序崩溃或红帽客户门户网站中的分段错误解决方案时，如何启用核心文件转储*

附录 A. CEPH 子系统默认日志记录级别值

各种 Ceph 子系统的默认日志记录级别值表。

子系统	日志级别	内存级别
asok	1	5
auth	1	5
buffer	0	0
client	0	5
context	0	5
CRUSH	1	5
default	0	5
filer	0	5
bluestore	1	5
finisher	1	5
heartbeatmap	1	5
javaclient	1	5
journaler	0	5
journal	1	5
lockdep	0	5
MDS 负载均衡器	1	5
mds locker	1	5
mds log expire	1	5
MDS 日志	1	5
MDS migrator	1	5
mds	1	5

子系统	日志级别	内存级别
monc	0	5
mon	1	5
ms	0	5
objclass	0	5
objectcacher	0	5
objecter	0	0
optracker	0	5
osd	0	5
paxos	0	5
perfcounter	1	5
rados	0	5
rbd	0	5
rgw	1	5
throttle	1	5
timer	0	5
tp	0	5

附录 B. CEPH 集群的健康消息

Red Hat Ceph Storage 集群可以引发的健康信息是有限的。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。

表 B.1. Monitor

健康代码	描述
DAEMON_OLD_VERSION	如果旧版本的 Ceph 在任何守护进程上运行，则发出警告。如果检测到多个版本，它将生成一个健康错误。
MON_DOWN	一个或多个 Ceph 监控守护进程当前为 down。
MON_CLOCK_SKEW	运行 **ceph-mon** 守护进程的节点上的时钟不够好同步。使用 **ntpd** 或 **chrony** 同步时钟来解决此问题。
MON_MSGR2_NOT_ENABLED	启用 **ms_bind_msgr2** 选项，但一个或多个 Ceph Monitor 没有配置为绑定到集群的 monmap 中的 v2 端口。通过运行 **ceph mon enable-msgr2** 命令来解决这个问题。
MON_DISK_LOW	一个或多个 Ceph 监控器在磁盘空间上较低。
MON_DISK_CRIT	一个或多个 Ceph 监控器在磁盘空间上至关重要。
MON_DISK_BIG	一个或多个 Ceph 监控器的数据库大小非常大。
AUTH_INSECURE_GLOBAL_ID_RECLAIM	一个或多个客户端或守护进程连接到存储集群，在重新连接到 Ceph monitor 时，这些集群不会安全地回收其 **global_id**。
AUTH_INSECURE_GLOBAL_ID_RECLAIM_ALLOWED	Ceph 目前配置为允许客户端使用不安全的进程重新连接到监控器，以回收其之前的 **global_id**，因为设置 **auth_allow_insecure_global_id_reclaim** 已设置为 **true**。

表 B.2. Manager（管理者）

健康代码	描述
MGR_DOWN	所有 Ceph Manager 守护进程当前都处于停机状态。
MGR_MODULE_DEPENDENCY	启用的 Ceph Manager 模块失败其依赖项检查。
MGR_MODULE_ERROR	Ceph Manager 模块遇到意外错误。通常，这意味着从模块的服务函数引发了未处理的异常。

表 B.3. OSD

健康代码	描述
OSD_DOWN	一个或多个 OSD 已标记为 down。
OSD_CRUSH_TYPE_DOWN	特定 CRUSH 子树中的所有 OSD 都标记为 down，如主机上的所有 OSD。例如，OSD_HOST_DOWN 和 OSD_ROOT_DOWN
OSD_ORPHAN	OSD 在 CRUSH map 层次结构中引用，但不存在。运行 **ceph osd crush rm osd._OSD_ID** 命令来移除 OSD。
OSD_OUT_OF_ORDER_FULL	*nearfull*, *backfillfull*, *full*, 或, *failsafefull* 的利用阈值不是升序。通过运行 **ceph osd set-nearfull-ratio *RATIO***, **ceph osd set-backfillfull-ratio *RATIO***, 和 **ceph osd set-full-ratio *RATIO*** 来调整阈值
OSD_FULL	一个或多个 OSD 已超过完整阈值，导致存储集群无法提供写入服务。通过一个小的 **ceph osd set-full-ratio *RATIO*** 来增加完全阈值以恢复写入可用性。
OSD_BACKFILLFULL	一个或多个 OSD 已超过 backfillfull 阈值，这将防止允许数据重新平衡到这个设备。
OSD_NEARFULL	一个或多个 OSD 已超过 nearfull 阈值。
OSDMAP_FLAGS	设置了一个或多个感兴趣的存储集群标志。这些标志包括 *full*,*pauserd*,*pausewr*,*noup*,*nodown*,*noin*,*noout*,*nobackfill*,*norecover*,*norebalance*,*noscrub*,*nodeep_scrub*, 和 *notieragent*。除了 *full*，标记可以通过 **ceph osd set *FLAG*** 和 **ceph osd unset *FLAG*** 命令进行清除。
OSD_FLAGS	一个或多个 OSD 或 CRUSH 具有感兴趣的标志。这些标志包括 *noup*、*nodown*、*noin* 和 *noout*。
OLD_CRUSH_TUNABLES	CRUSH map 使用非常旧的设置，应该更新。
OLD_CRUSH_STRAW_CALC_VERSION	CRUSH map 使用较旧的、非优化的方法来计算 **straw** bucket 的中间权重值。

健康代码	描述
CACHE_POOL_NO_HIT_SET	一个或多个缓存池没有配置为跟踪利用率，这会阻止分层代理识别冷对象以清空并从缓存中驱除。使用 **ceph osd pool set_POOL_NAME_ hit_set_type *TYPE*, ceph osd pool set *POOL_NAME* hit_set_period *PERIOD_IN_SECONDS*, ceph osd pool set *POOL_NAME* hit_set_count *NUMBER_OF_HIT_SETS*,** 和 **ceph osd pool set *POOL_NAME* hit_set_fpp *TARGET_FALSE_POSITIVE_RATE*** 命令配置缓存池中的击中集。
OSD_NO_SORTBITWISE	未设置 **sortbitwise** 标志。使用 **ceph osd set sortbitwise** 命令设置标志。
POOL_FULL	一个或多个池已达到其配额，不再允许写入。使用 **ceph osd pool set-quota *POOL_NAME* max_objects *NUMBER_OF_OBJECTS*** 和 **ceph osd pool set-quota *POOL_NAME* max_bytes *BYTES*** 或删除一些现有数据来增加池配额，以减少使用率。
BLUEFS_SPILLOVER	使用 BlueStore 后端的一个或多个 OSD 被分配 db 分区，但空间已填满，因此元数据已"中断"到正常较慢的设备。使用 **ceph config set osd bluestore_warn_on_bluefs_spillover false** 命令禁用此功能。
BLUEFS_AVAILABLE_SPACE	此输出提供了三个值，即 *BDEV_DB free*、*BDEV_SLOW free* 和 *available_from_bluestore*。
BLUEFS_LOW_SPACE	如果 BlueStore 文件系统(BlueFS)在可用空间上运行较低，并且只有 little **available_from_bluestore**，可以考虑减少 BlueFS 分配单元大小。
BLUESTORE_FRAGMENTATION	因为 BlueStore 在底层存储上工作可用空间将变得碎片。这是正常现象，但过度的碎片将导致减慢。
BLUESTORE_LEGACY_STATFS	BlueStore 根据每个池的粒度跟踪其内部使用量统计，一个或多个 OSD 具有 BlueStore 卷。使用 **ceph config set global bluestore_warn_on_legacy_statfs false** 命令禁用警告。

健康代码	描述
BLUESTORE_NO_PER_POOL_OMAP	BlueStore 按池跟踪 omap 空间利用率。使用 **ceph config set global bluestore_warn_on_no_per_pool_omap false** 命令禁用警告。
BLUESTORE_NO_PER_PG_OMAP	BlueStore 按 PG 跟踪 omap 空间利用率。使用 **ceph config set global bluestore_warn_on_no_per_pg_omap false** 命令禁用警告。
BLUESTORE_DISK_SIZE_MISMATCH	使用 BlueStore 的一个或多个 OSD 在物理设备的大小和元数据跟踪其大小之间存在内部不一致。
BLUESTORE_NO_COMPRESSION `	一个或多个 OSD 无法加载 BlueStore 压缩插件。这可能是由安装中断造成的，其中 **ceph-osd** 二进制文件与压缩插件不匹配，或者是最近没有包括 **ceph-osd** 守护进程重启的升级。
BLUESTORE_SPURIOUS_READ_ERRORS	使用 BlueStore 的一个或多个 OSD 检测到主设备中错误的读错误。通过重试磁盘读取，BlueStore 已从这些错误中恢复。

表 B.4. 设备健康状况

健康代码	描述
DEVICE_HEALTH	一个或多个设备应该很快失败，其中 **mgr/devicehealth/warn_threshold** 配置选项控制警告阈值。将设备标记为 *out*，以迁移数据并替换硬件。
DEVICE_HEALTH_IN_USE	一个或多个设备应该很快失败，并根据 **mgr/devicehealth/mark_out_threshold** 标记为存储集群的"out"，但它仍然参与一个 PG。
DEVICE_HEALTH_TOOMANY	太多的设备应该很快失败，并且启用了 **mgr/devicehealth/self_heal** 行为，因此标记所有异常设备将超过集群 **mon_osd_min_in_ratio** 比率，防止太多 OSD 自动标记为 **out**。

表 B.5. 池和放置组

健康代码	描述
PG_AVAILABILITY	数据可用性会降低，这意味着存储集群无法为集群中的某些数据提供潜在的读写请求。

健康代码	描述
PG_DEGRADED	一些数据的数据冗余会降低，这意味着存储集群没有复制池或纠删代码片段所需的副本数。
PG_RECOVERY_FULL	由于存储集群中缺少可用空间，数据冗余可能会减少或面临风险，特别是一个或多个 PG 设置了 **recovery_toofull** 标志，这意味着集群无法迁移或恢复数据，因为一个或多个 OSD 超过 **full** 阈值。
PG_BACKFILL_FULL	由于存储集群中缺少可用空间，数据冗余可能会减少或面临风险，特别是一个或多个 PG 设置了 **backfill_toofull** 标志，这意味着集群无法迁移或恢复数据，因为一个或多个 OSD 超过 **full** 阈值。
PG_DAMAGED	数据清理在存储集群中发现了一些数据一致性问题，特别是一个或多个 PG 设置了不一致或 **snaptrim_error** 标志，表明之前的清理操作发现问题，或者设置了 **repair** 标志，这意味着当前正在进行此类不一致的修复。
OSD_SCRUB_ERRORS	最近的 OSD 清理的不一致。
OSD_TOO_MANY_REPAIRS	当出现读取错误并存在另一个副本时，可使用它立即修复错误，以便客户端可以获取对象数据。
LARGE_OMAP_OBJECTS	一个或多个池包括大量 omap 对象，由 **osd_deep_scrub_large_omap_object_key_threshold** 或 **osd_deep_scrub_large_omap_object_value_sum_threshold** 决定，或由这两者同时决定。使用 **ceph config set osd osd_deep_scrub_large_omap_object_key_threshold *KEYS*** 和 **ceph config set osd osd_deep_scrub_large_omap_object_value_sum_threshold *BYTES*命令调整阈值**。
CACHE_POOL_NEAR_FULL	缓存层池几乎已满。使用 **ceph osd pool set *CACHE_POOL_NAME* target_max_bytes *BYTES*** 和 **ceph osd pool set *CACHE_POOL_NAME* target_max_bytes *BYTES*命令调整缓存池** 目标大小。
TOO_FEW_PGS	存储集群中使用的 PG 数量低于每个 OSD 的 **mon_pg_warn_min_per_osd** PG 的可配置阈值。

健康代码	描述
POOL_PG_NUM_NOT_POWER_OF_TWO	一个或多个池带有值不是二的指数的 **pg_num** 值。使用 **ceph config set global mon_warn_on_pool_pg_num_not_power_of_two false** 命令禁用警告。
POOL_TOO_FEW_PGS	一个或多个池可能具有更多 PG，具体取决于池中当前存储的数据量。您可以使用 **ceph osd pool set *POOL_NAME* pg_autoscale_mode off** 命令禁用 PG 的自动扩展，使用 **ceph osd pool set *POOL_NAME* pg_autoscale_mode on** 命令自动调整 PG 数量，或使用 **ceph osd pool set *POOL_NAME* pg_num _NEW_PG_NUMBER** 命令手动设置 PG 数量。
TOO_MANY_PGS	存储集群中使用的 PG 数量高于每个 OSD 的可配置阈值 **mon_max_pg_per_osd** PG。通过添加更多硬件增加集群中的 OSD 数量。
POOL_TOO_MANY_PGS	一个或多个池可能具有更多 PG，具体取决于池中当前存储的数据量。您可以使用 **ceph osd pool set *POOL_NAME* pg_autoscale_mode off** 命令禁用 PG 的自动扩展，使用 **ceph osd pool set *POOL_NAME* pg_autoscale_mode on** 命令自动调整 PG 数量，或使用 **ceph osd pool set *POOL_NAME* pg_num _NEW_PG_NUMBER** 命令手动设置 PG 数量。
POOL_TARGET_SIZE_BYTES_OVERCOMMITTED	一个或多个池设置了 **target_size_bytes** 属性，用于估算池的预期大小，但这些值超过可用存储总量。使用 **ceph osd pool set *POOL_NAME* target_size_bytes 0** 命令将池的值设为零。
POOL_HAS_TARGET_SIZE_BYTES_AND_RATIO	一个或多个池同时设置了 **target_size_bytes** 和 **target_size_ratio**，以估算池的预期大小。使用 **ceph osd pool set *POOL_NAME* target_size_bytes 0** 命令将池的值设为零。
TOO_FEW_OSDS	存储集群中的 OSD 数量低于 **o'sd_pool_default_size** 的可配置阈值。
SMALLER_PGP_NUM	一个或多个池带有值小于 **pg_num** 的**pgp_num** 值。这通常表示 PG 计数已增加，且不会增加放置行为。通过设置 **pgp_num** 匹配 **pg_num** with **ceph osd pool set *POOL_NAME* pgp_num *PG_NUM_VALUE*** 来解决这个问题。
MANY_OBJECTS_PER_PG	每个 PG 一个或多个池的平均对象数量要高于整个存储集群平均值。特定阈值由 **mon_pg_warn_max_object_skew** 配置值控制。

健康代码	描述
POOL_APP_NOT_ENABLED	存在一个池，其中包含一个或多个对象，但尚未标记供特定应用使用。通过使用 **rbd pool init *POOL_NAME*** 命令标记池，从而解决此警告。
POOL_FULL	一个或多个池已达到其配额。触发此错误条件的阈值由 **mon_pool_quota_crit_threshold** 配置选项控制。
POOL_NEAR_FULL	一个或多个池正在接近配置的全度阈值。使用 **ceph osd pool set-quota *POOL_NAME* max_objects *NUMBER_OF_OBJECTS*** 和 **ceph osd pool set-quota *POOL_NAME* max_bytes *BYTES*命令调整** 池配额。
OBJECT_MISPLACED	存储群集中的一个或多个对象不存储在存储器集群希望它存储的节点上。这表明，由于最近一些存储集群更改，数据迁移尚未完成。
OBJECT_UNFOUND	存储集群中无法找到一个或多个对象，特别是 OSD 知道对象应存在新的或更新的副本，但当前在线的 OSD 上尚未找到该对象版本的副本。
SLOW_OPS	一个或多个 OSD 或 monitor 的请求需要很长时间进行处理。这可能代表了极端负载、存储设备缓慢或软件漏洞。
PG_NOT_SCRUBBED	最近没有清理一个或多个 PG。PG 通常会在全局 **osd_scrub_max_interval** 指定的每个配置间隔内清理。使用 **ceph pg scrub *PG_ID*** 命令启动刮除。
PG_NOT_DEEP_SCRUBBED	一个或多个 PG 尚未最近清理。使用 **ceph pg deep-scrub *PG_ID*命令启动** 清理。PG 通常会清理每个 **osd_deep_scrub_interval** 秒，当 **mon_warn_pg_not_deep_scrubbed_ratio** 百分比的间隔没有清理后，这个警告会触发。
PG_SLOW_SNAP_TRIMMING	一个或多个 PG 的快照修剪队列已超过配置的警告阈值。这表明最近删除了大量的快照，或者 OSD 无法足够快速地修剪快照，以跟上新快照删除的速度。

表 B.6. 其它

健康代码	描述
RECENT_CRASH	一个或多个 Ceph 守护进程最近崩溃，且管理员尚未确认崩溃。

健康代码	描述
TELEMETRY_CHANGED	遥测已经启用，但遥测报告的内容从那时起发生了变化，因此将不会发送遥测报告。
AUTH_BAD_CAPS	一个或多个 auth 用户具有无法由监控器解析的功能。使用 **ceph auth *ENTITY_NAME DAEMON_TYPE CAPS*** 命令更新用户的能力。
OSD_NO_DOWN_OUT_INTERVAL	**mon_osd_down_out_interval** 选项设为零，这意味着系统不会在 OSD 失败后自动执行任何修复或修复操作。使用 **ceph config global mon_warn_on_osd_down_out_interval_zero false 命令静默** 间隔。
DASHBOARD_DEBUG	启用 Dashboard debug 模式。这意味着，如果在处理 REST API 请求时出现错误，HTTP 错误响应包含一个 Python 回溯。使用 **ceph dashboard debug disable** 命令禁用调试模式。