Red Hat Ceph Storage 7

块设备指南

管理、创建、配置和使用 Red Hat Ceph Storage 块设备

Last Updated: 2024-06-14

Red Hat Ceph Storage 7 块设备指南

管理、创建、配置和使用 Red Hat Ceph Storage 块设备

法律通告

摘要

本文档描述了如何管理、创建、配置和使用 Red Hat Ceph Storage 块设备。 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 CEPH 块设备简介

块是具有一定长度的一组字节序列，例如 512 字节的数据块。将多个块组合到一个文件中，可用作您可以从中读取和写入的存储设备。基于块的存储接口是使用旋转介质存储数据的最常见的方式，例如：

- 硬盘驱动器
- CD/DVD 磁盘
- 软盘
- 传统的 9 轨磁带

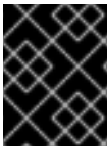
因为块设备的广泛使用，虚拟块设备成为与 Red Hat Ceph Storage 等海量数据存储系统交互的理想候选者。

Ceph 块设备是精简调配、可调整大小的，并在 Ceph 存储集群中的多个对象存储设备 (OSD) 上存储数据分条。Ceph 块设备也称为可靠的自主分布式对象存储 (RADOS) 块设备(RBD)。Ceph 块设备利用 RADOS 功能，例如：

- 快照
- 复制
- 数据一致性

Ceph 块设备利用 **librbd** 库与 OSD 交互。

Ceph 块设备为内核虚拟机 (KVM)（如快速仿真器(QEMU)）和基于云的计算系统（如 OpenStack）提供高性能，它们依赖于 **libvirt** 和 QEMU 实用程序与 Ceph 块设备集成。您可以使用同一个存储集群同时运行 Ceph 对象网关和 Ceph 块设备。



重要

若要使用 Ceph 块设备，您需要有权访问正在运行的 Ceph 存储集群。有关安装 Red Hat Ceph Storage 集群的详情，请参阅 *Red Hat Ceph Storage 安装指南*。

第 2 章 CEPH 块设备

作为存储管理员，熟悉 Ceph 的块设备命令可帮助您有效管理 Red Hat Ceph Storage 集群。您可以创建和管理块设备池和镜像，以及启用和禁用 Ceph 块设备的各种功能。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

2.1. 显示命令帮助

显示命令行界面中的命令和子命令在线帮助。

注意

-h 选项仍然显示所有可用命令的帮助信息。

先决条件

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

流程

1. 使用 **rbd help** 命令显示特定 **rbd** 命令及其子命令的帮助信息：

   语法

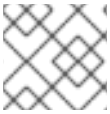
   ```
   rbd help COMMAND SUBCOMMAND
   ```

2. 显示 **snap list** 命令的帮助信息：

   ```
   [root@rbd-client ~]# rbd help snap list
   ```

2.2. 创建块设备池

在使用块设备客户端之前，请确保已启用并初始化 **rbd** 的池。



注意

您**必**须先创建一个池，然后才能将它指定为来源。

先决条件

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

流程

1. 要创建 **rbd** 池，请执行以下操作：

语法

```
ceph osd pool create POOL_NAME PG_NUM
ceph osd pool application enable POOL_NAME rbd
rbd pool init -p POOL_NAME
```

示例

```
[root@rbd-client ~]# ceph osd pool create pool1
[root@rbd-client ~]# ceph osd pool application enable pool1 rbd
[root@rbd-client ~]# rbd pool init -p pool1
```

其它资源

- 如需了解更多详细信息，请参见 *Red Hat Ceph Storage 策略指南* 中的 *池* 一章。

2.3. 创建块设备镜像

在添加块设备到节点之前，在 Ceph 存储集群中为其创建镜像。

先决条件

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

流程

1. 要创建块设备镜像，请执行以下命令：

   语法

   ```
   rbd create IMAGE_NAME --size MEGABYTES --pool POOL_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd create image1 --size 1024 --pool pool1
   ```

   本例创建一个名为 **image1** 的 1 GB 镜像，该镜像将信息存储在名为 **pool1** 的池中。

   注意

   在创建镜像之前，确保池存在。

其它资源

- 如需了解更多详细信息，请参阅 *Red Hat Ceph Storage 块设备指南* 中的 *创建块设备池* 部分。

2.4. 列出块设备镜像

列出块设备镜像。

先决条件

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

流程

1. 要列出 **rbd** 池中的块设备，请执行以下命令：

注意

RBD 是默认的池名称。

示例

```
[root@rbd-client ~]# rbd ls
```

2. 列出特定池中的块设备：

语法

```
rbd ls POOL_NAME
```

示例

```
[root@rbd-client ~]# rbd ls pool1
```

2.5. 检索块设备镜像信息

检索块设备镜像上的信息。

先决条件

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

流程

1. 要从默认 **rbd** 池中的特定镜像检索信息，请运行以下命令：

语法

```
rbd --image IMAGE_NAME info
```

示例

```
[root@rbd-client ~]# rbd --image image1 info
```

2. 从池中的镜像检索信息：

语法

```
rbd --image IMAGE_NAME -p POOL_NAME info
```

示例

```
[root@rbd-client ~]# rbd --image image1 -p pool1 info
```

2.6. 重新定义块设备镜像大小

Ceph 块设备镜像是精简配置。在开始将数据保存到其中之前，它们不会实际使用任何物理存储。但是，它们具有您通过 **--size** 选项设置的最大容量。

先决条件

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

流程

1. 为默认的 **rbd** 池增加或缩小 Ceph 块设备镜像的最大大小：

   语法

   ```
   rbd resize --image IMAGE_NAME --size SIZE
   ```

   示例

   ```
   [root@rbd-client ~]# rbd resize --image image1 --size 1024
   ```

2. 为特定池增加或缩小 Ceph 块设备镜像的最大大小：

   语法

   ```
   rbd resize --image POOL_NAME/IMAGE_NAME --size SIZE
   ```

   示例

   ```
   [root@rbd-client ~]# rbd resize --image pool1/image1 --size 1024
   ```

2.7. 删除块设备镜像

删除块设备镜像。

先决条件

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

流程

1. 从默认 **rbd** 池中删除块设备：

   语法

   ```
   rbd rm IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd rm image1
   ```

2. 从特定池中删除块设备：

   语法

   ```
   rbd rm IMAGE_NAME -p POOL_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd rm image1 -p pool1
   ```

2.8. 将块设备镜像移到回收站中

RADOS 块设备 (RBD) 镜像可以使用 **rbd trash** 命令移到回收站中。此命令提供的选项比 **rbd rm** 命令更多。

镜像移到回收站后，可以稍后将其从回收站中删除。这有助于避免意外删除。

先决条件

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

流程

1. 执行以下操作将镜像移到回收站中：

   语法

   ```
   rbd trash mv [POOL_NAME/] IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd trash mv pool1/image1
   ```

   镜像处于回收站中后，将分配一个唯一镜像 ID。

注意

如果需要使用任何回收选项，则在指定镜像时需要此镜像 ID。

2. 为一个垃圾箱中镜像 ID 的列表执行 **rbd trash list *POOL_NAME***。此命令还会返回镜像的预删除名称。此外，还有一个可选的 **--image-id** 参数，可用于 **rbd info** 和 **rbd snap** 命令。将 **--image-id** 与 **rbd info** 命令搭配使用，查看垃圾箱中的镜像属性，并使用 **rbd snap** 从回收站中删除镜像的快照。

3. 要从回收站中删除镜像，请执行以下操作：

   语法

   ```
   rbd trash rm [POOL_NAME/] IMAGE_ID
   ```

   示例

   ```
   [root@rbd-client ~]# rbd trash rm pool1/d35ed01706a0
   ```

   重要

   从回收站中删除镜像后，它将无法被恢复。

4. 执行 **rbd trash restore** 命令以恢复镜像：

   语法

   ```
   rbd trash restore [POOL_NAME/] IMAGE_ID
   ```

   示例

   ```
   [root@rbd-client ~]# rbd trash restore pool1/d35ed01706a0
   ```

5. 从回收站中删除所有已过期的镜像：

   语法

   ```
   rbd trash purge POOL_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd trash purge pool1
   Removing images: 100% complete...done.
   ```

2.9. 定义自动垃圾清除调度

您可以调度在池中定期清除操作。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

- 客户端节点的根级别访问权限。

流程

1. 要添加垃圾清除计划，请执行：

   语法

   ```
   rbd trash purge schedule add --pool POOL_NAME INTERVAL
   ```

   示例

   ```
   [ceph: root@host01 /]# rbd trash purge schedule add --pool pool1 10m
   ```

2. 要列出垃圾箱清除计划，请执行：

   语法

   ```
   rbd trash purge schedule ls --pool POOL_NAME
   ```

   示例

   ```
   [ceph: root@host01 /]# rbd trash purge schedule ls --pool pool1
   every 10m
   ```

3. 要了解回收清除计划的状态，请执行：

   示例

   ```
   [ceph: root@host01 /]# rbd trash purge schedule status
   POOL   NAMESPACE   SCHEDULE  TIME
   pool1              2021-08-02 11:50:00
   ```

4. 要删除垃圾清除计划，请执行：

   语法

   ```
   rbd trash purge schedule remove --pool POOL_NAME INTERVAL
   ```

   示例

   ```
   [ceph: root@host01 /]# rbd trash purge schedule remove --pool pool1 10m
   ```

2.10. 启用和禁用镜像功能

块设备镜像（如 **fast-diff**、**exclusive-lock**、**object-map** 或 **deep-flatten**）会被默认启用。您可以在已存在的镜像上启用或禁用这些镜像功能。

注意

deep flatten 功能只能在现有的镜像上禁用，而不能启用。要使用 **deep flatten** 功能，需要在创建镜像时启用它。

先决条件

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

流程

1. 从池中的特定镜像检索信息：

   语法

   ```
   rbd --image POOL_NAME/IMAGE_NAME info
   ```

   示例

   ```
   [ceph: root@host01 /]# rbd --image pool1/image1 info
   ```

2. 启用一个功能：

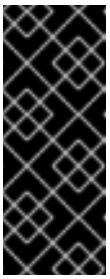
   语法

   ```
   rbd feature enable POOL_NAME/IMAGE_NAME FEATURE_NAME
   ```

   a. 要在 **pool1** 池中的 **image1** 镜像启用 **exclusive-lock** 功能：

      示例

      ```
      [ceph: root@host01 /]# rbd feature enable pool1/image1 exclusive-lock
      ```

      重要

      如果启用了 **fast-diff** 和 **object-map** 功能，则重建对象映射：

      + 语法

      ```
      rbd object-map rebuild POOL_NAME/IMAGE_NAME
      ```

3. 禁用一个功能：

   语法

   ```
   rbd feature disable POOL_NAME/IMAGE_NAME FEATURE_NAME
   ```

   a. 要在 **pool1** 池中的 **image1** 镜像中禁用 **fast-diff** 功能：

      示例

```
[ceph: root@host01 /]# rbd feature disable pool1/image1 fast-diff
```

2.11. 使用镜像元数据

Ceph 支持以键值对的形式添加自定义镜像元数据添。这些键值对没有严格的格式限制。

此外，通过使用元数据，您可以为特定镜像设置 RADOS 块设备 (RBD) 配置参数。

使用 **rbd image-meta** 命令处理元数据。

先决条件

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

流程

1. 设置新的元数据键值对：

   语法

   ```
   rbd image-meta set POOL_NAME/IMAGE_NAME KEY VALUE
   ```

   示例

   ```
   [ceph: root@host01 /]# rbd image-meta set pool1/image1 last_update 2021-06-06
   ```

   本例将 **pool1** 池中 **image1** 镜像的 **last_update** 键设置为 **2021-06-06** 值。

2. 查看一个键的值：

   语法

   ```
   rbd image-meta get POOL_NAME/IMAGE_NAME KEY
   ```

   示例

   ```
   [ceph: root@host01 /]# rbd image-meta get pool1/image1 last_update
   ```

   这个示例查看 **last_update** 键的值。

3. 显示镜像中的所有元数据：

   语法

   ```
   rbd image-meta list POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [ceph: root@host01 /]# rbd image-meta list pool1/image1
   ```

本例列出了 **pool1** 池中 **image1** 镜像设置的元数据。

4. 删除元数据键值对：

语法

```
rbd image-meta remove POOL_NAME/IMAGE_NAME KEY
```

示例

```
[ceph: root@host01 /]# rbd image-meta remove pool1/image1 last_update
```

本例从 **pool1** 池中的 **image1** 镜像中删除 **last_update** 键值对。

5. 覆盖特定镜像的 Ceph 配置文件中设置的 RBD 镜像配置设置：

语法

```
rbd config image set POOL_NAME/IMAGE_NAME  PARAMETER VALUE
```

示例

```
[ceph: root@host01 /]# rbd config image set pool1/image1 rbd_cache false
```

本例禁用 **pool1** 池中 **image1** 镜像的 RBD 缓存。

其它资源

- 如需了解可能配置选项列表，请参阅 *Red Hat Ceph Storage 块设备指南* 中的*块设备常规选项*部分。

2.12. 在池之间移动镜像

您可以在同一集群内的不同池之间移动 RADOS 块设备 (RBD) 镜像。

在此过程中，源镜像会复制到具有所有快照历史记录的目标镜像，也可选择性地复制到源镜像的父镜像中，以帮助保留稀疏性。源镜像是只读的，目标镜像是可写的。目标镜像在迁移过程中链接到源镜像。

您可以在使用新目标镜像时安全地在后台运行此过程。但是，在准备步骤前停止使用目标镜像的所有客户端，以确保更新使用该镜像的客户端以指向新的目标镜像。

重要

krbd 内核模块目前不支持实时迁移。

先决条件

- 停止所有使用该源镜像的客户端。
- 客户端节点的根级别访问权限。

流程

1. 通过创建跨链接源和目标镜像的新目标镜像准备迁移：

   语法

   ```
   rbd migration prepare SOURCE_IMAGE TARGET_IMAGE
   ```

   替换：

   - *SOURCE_IMAGE*，带有要移动的镜像的名称。使用 *POOL/IMAGE_NAME* 格式。
   - *TARGET_IMAGE*，带有新镜像的名称。使用 *POOL/IMAGE_NAME* 格式。

   示例

   ```
   [root@rbd-client ~]# rbd migration prepare pool1/image1 pool2/image2
   ```

2. 验证新目标镜像的状态，这应该为 **prepared** ：

   语法

   ```
   rbd status TARGET_IMAGE
   ```

   示例

   ```
   [root@rbd-client ~]# rbd status pool2/image2
   Watchers: none
   Migration:
           source: pool1/image1 (5e2cba2f62e)
           destination: pool2/image2 (5e2ed95ed806)
           state: prepared
   ```

3. （可选）使用新目标镜像名称重新启动客户端。

4. 将源镜像复制到目标镜像：

   语法

   ```
   rbd migration execute TARGET_IMAGE
   ```

   示例

   ```
   [root@rbd-client ~]# rbd migration execute pool2/image2
   ```

5. 确保迁移已完成：

   示例

   ```
   [root@rbd-client ~]# rbd status pool2/image2
   Watchers:
     watcher=1.2.3.4:0/3695551461 client.123 cookie=123
   Migration:
   ```

```
source: pool1/image1 (5e2cba2f62e)
destination: pool2/image2 (5e2ed95ed806)
state: executed
```

6. 通过删除源镜像和目标镜像之间的跨链接来提交迁移，这也会移除源镜像：

   语法

   ```
   rbd migration commit TARGET_IMAGE
   ```

   示例

   ```
   [root@rbd-client ~]# rbd migration commit pool2/image2
   ```

   如果源镜像是一个或多个克隆的父镜像，请在确保克隆镜像不在使用后使用 **--force** 选项：

   示例

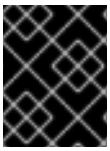
   ```
   [root@rbd-client ~]# rbd migration commit pool2/image2 --force
   ```

7. 如果您在准备步骤后没有重新启动客户端，请使用新目标镜像名称重启客户端。

2.13. 迁移池

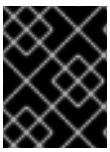
您可以迁移或复制 RADOS 块设备(RBD)镜像。

在此过程中，源镜像会被导出，然后导入。



重要

如果工作负载 **仅包含** RBD 镜像，请使用此迁移过程。工作负载中没有 **rados cppool** 镜像。如果工作负载中存在 **rados cppool** 镜像，请参阅 *存储策略指南中的* 迁移 池。



重要

在运行导出和导入命令时，请确保相关的 RBD 镜像中没有活跃的 I/O。建议您在这个池迁移期间关闭生产环境。

先决条件

- 停止要导出和导入的 RBD 镜像中的所有活动 I/O。
- 客户端节点的根级别访问权限。

流程

- 迁移卷。

  语法

  ```
  rbd export volumes/VOLUME_NAME - | rbd import --image-format 2 -
  volumes_new/VOLUME_NAME
  ```

示例

```
[root@rbd-client ~]# rbd export volumes/volume-3c4c63e3-3208-436f-9585-fee4e2a3de16 - |
rbd import --image-format 2 - volumes_new/volume-3c4c63e3-3208-436f-9585-
fee4e2a3de16
```

- 如果需要使用本地驱动器导入或导出，可以划分命令，首先导出到本地驱动器，然后将文件导入到新池中。

语法

```
rbd export volume/VOLUME_NAME FILE_PATH
rbd import --image-format 2 FILE_PATH volumes_new/VOLUME_NAME
```

示例

```
[root@rbd-client ~]# rbd export volumes/volume-3c4c63e3-3208-436f-9585-fee4e2a3de16
<path of export file>
[root@rbd-client ~]# rbd import --image-format 2 <path> volumes_new/volume-3c4c63e3-
3208-436f-9585-fee4e2a3de16
```

2.14. RBDMAP 服务

systemd 单元文件 **rbdmap.service** 包含在 **ceph-common** 软件包中。**rbdmap.service** 单元执行 **rbdmap** shell 脚本。

此脚本自动为一个或多个 RBD 镜像自动映射和取消 map RADOS 块设备 (RBD)。脚本可以随时手动运行，但典型的用例是在引导时自动挂载 RBD 镜像，并在关机时卸载。脚本采用单个参数，可以是 **map**（用于挂载）或 **unmap**（卸载）RBD 镜像。脚本解析配置文件，默认为 **/etc/ceph/rbdmap**，但可使用名为 **RBDMAPFILE** 的环境变量来覆盖。配置文件的每一行对应于 RBD 镜像。

配置文件格式的格式如下：

IMAGE_SPEC RBD_OPTS

其中 *IMAGE_SPEC* 指定 *POOL_NAME* / *IMAGE_NAME*，或仅使用 *IMAGE_NAME*，在这种情况下，*POOL_NAME* 默认为 **rbd**。*RBD_OPTS* 是要传递到底层 **rbd map** 命令的选项列表。这些参数及其值应指定为用逗号分开的字符串：

OPT1=VAL1,OPT2=VAL2,...,OPT_N=VAL_N

这将导致脚本发出类似如下的 **rbd map** 命令：

语法

```
rbd map POOLNAME/IMAGE_NAME --OPT1 VAL1 --OPT2 VAL2
```

注意

对于包含逗号或相等符号的选项和值，可以使用简单的符号来防止替换它们。

成功后，**rbd map** 操作会将镜像映射到 **/dev/rbdX** 设备，此时会触发一个 **udev** 规则来创建一个友好的设

备名称 symlink，如 **/dev/rbd/*POOL_NAME*/*IMAGE_NAME*** 指向实际映射的设备。要成功挂载或卸载，友好的设备名称必须在 **/etc/fstab** 文件中具有对应的条目。为 RBD 镜像编写 **/etc/fstab** 条目时，最好指定 **noauto** 或 **nofail** 挂载选项。这可防止 init 系统在设备存在前尝试过早挂载该设备。

其它资源

- 有关可能选项的完整列表，请参见 **rbd** man page。

2.15. 配置 RBDMAP 服务

要在引导时或关机时自动映射和挂载或取消 map 和卸载 RADOS 块设备 (RBD)。

先决条件

- 对执行挂载的节点的根级别访问权限。
- 安装 **ceph-common** 软件包.

流程

1. 打开并编辑 **/etc/ceph/rbdmap** 配置文件。
2. 将 RBD 镜像或镜像添加到配置文件中：

   示例

   ```
   foo/bar1    id=admin,keyring=/etc/ceph/ceph.client.admin.keyring
   foo/bar2
   id=admin,keyring=/etc/ceph/ceph.client.admin.keyring,options='lock_on_read,queue_depth=102
   4'
   ```

3. 保存对配置文件的更改。
4. 启用 RBD 映射服务：

   示例

   ```
   [root@client ~]# systemctl enable rbdmap.service
   ```

其它资源

- 如需了解 RBD 系统服务的更多详细信息，请参阅 *Red Hat Ceph Storage 块设备指南* 中的 ***rbdmap*** *service* 部分。

2.16. 持久性 WRITE LOG CACHE

在 Red Hat Ceph Storage 集群中，持久写入日 志(PWL) 缓存为基于 librbd 的 RBD 客户端提供持久的、容错的写回缓存。

PWL 缓存使用日志排序的回写设计，它会在内部维护检查点，以便刷新回集群的写入始终崩溃。如果客户端缓存完全丢失，磁盘镜像仍然一致，但数据也会过时。您可以将 PWL 缓存使用带有持久性内存 (PMEM) 或固态磁盘 (SSD) 作为缓存设备。

对于 PMEM，缓存模式是副本写入日志 (RWL) 和 SSD，缓存模式为 (SSD)。目前，PWL 缓存支持 RWL 和 SSD 模式，并默认禁用。

PWL 缓存的主要优点是：

- 当缓存未满时，PWL 缓存可以提供更高的性能。缓存越大，高性能持续的时间越长。
- PWL 缓存提供持久性，且不比 RBD 缓存慢太多。虽然 RBD 缓存速度更快，但因其本身的易失性，不能保证数据顺序和持久性。
- 在处于稳定状态时，缓存已满，性能会受 I/O 数量的影响。例如，在低 io_depth 的情况下 PWL 会提供更高的性能，但对于高 io_depth，例如 I/O 的数量大于 32 时，性能通常会低于没有缓存的情况。

PMEM 缓存的用例：

- 与 RBD 缓存不同，PWL 缓存具有非易变的特征，适用于在您不希望数据丢失和需要性能的情况下使用。
- RWL 模式提供低延迟。对于突发 I/O，它具有稳定的低延迟，适用于对稳定的低延迟有高要求的情况。
- 在低 I/O 深度或没有太多 inflight I/O 的情况下，RWL 模式还具有高连续且稳定的性能。

SSD 缓存的用例是：

- SSD 模式的优点与 RWL 模式类似。SSD 硬件相对较低且流行，但其性能比 PMEM 稍低。

2.17. 持久性写入日志缓存限制

当使用 Persistent Write Log (PWL) 缓存时，应该考虑一些限制。

- PMEM 和 SSD 的底层实现有所不同，PMEM 的性能更高。目前，PMEM 提供了 "persist on write"，SSD 为 "persist on flush or checkpoint"。在以后的发行版本中，这两种模式将可以被配置。
- 当用户频繁切换并重复打开和关闭镜像时，Ceph 会显示性能不佳。如果启用了 PWL 缓存，则性能会较差。不建议在 Flexible I/O (fio) 测试中设置 **num_jobs**，而是设置多个作业来编写不同的镜像。

2.18. 启用持久写入日志缓存

您可以通过设置 Ceph RADOS 块设备(RBD) **rbd_persistent_cache_mode** 和 **rbd_plugins** 选项，在 Red Hat Ceph Storage 集群上启用持久性写入日志缓存(PWL)。

重要

必须启用 exclusive-lock 功能以启用持久的写入日志缓存。只有在获取 exclusive-lock 后才能加载缓存。对新创建的镜像会默认启用 exclusive-locks，除非由 **rbd create** 命令的 **rbd_default_features** 配置选项或 **--image-feature** 标志覆盖。有关 **exclusive-lock** 功能的详情，请参阅 *启用和禁用镜像功能* 部分。

使用 **ceph config set** 命令，在主机级别上设置持久的写入日志缓存选项。使用 **rbd config pool set** 或 **rbd config image set** 命令在池或镜像级别上设置持久的写入日志缓存选项。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 监控节点的 root 级别访问权限。
- 启用 exclusive-lock 功能。
- 客户端磁盘是持久内存 (PMEM) 或固态磁盘 (SSD)。
- 禁用 RBD 缓存。

流程

1. 启用 PWL 缓存：

   a. 在主机级别，使用 **ceph config set** 命令：

   语法

   ```
   ceph config set client rbd_persistent_cache_mode CACHE_MODE
   ceph config set client rbd_plugins pwl_cache
   ```

   使用 **rwl** 或 **ssd** 替换 *CACHE_MODE*。

   示例

   ```
   [ceph: root@host01 /]# ceph config set client rbd_persistent_cache_mode ssd
   [ceph: root@host01 /]# ceph config set client rbd_plugins pwl_cache
   ```

   b. 在池级别，使用 **rbd config pool set** 命令：

   语法

   ```
   rbd config pool set POOL_NAME rbd_persistent_cache_mode CACHE_MODE
   rbd config pool set POOL_NAME rbd_plugins pwl_cache
   ```

   使用 **rwl** 或 **ssd** 替换 *CACHE_MODE*。

   示例

   ```
   [ceph: root@host01 /]# rbd config pool set pool1 rbd_persistent_cache_mode ssd
   [ceph: root@host01 /]# rbd config pool set pool1 rbd_plugins pwl_cache
   ```

   c. 在镜像级别，使用 **rbd config image set** 命令：

   语法

   ```
   rbd config image set POOL_NAME/IMAGE_NAME rbd_persistent_cache_mode
   CACHE_MODE
   rbd config image set POOL_NAME/IMAGE_NAME rbd_plugins pwl_cache
   ```

   使用 **rwl** 或 **ssd** 替换 *CACHE_MODE*。

示例

```
[ceph: root@host01 /]# rbd config image set pool1/image1 rbd_persistent_cache_mode ssd
[ceph: root@host01 /]# rbd config image set pool1/image1 rbd_plugins pwl_cache
```

2. 可选：在主机、池或镜像级别设置额外的 RBD 选项：

语法

```
rbd_persistent_cache_mode CACHE_MODE
rbd_plugins pwl_cache
rbd_persistent_cache_path /PATH_TO_CACHE_DIRECTORY (1)
rbd_persistent_cache_size PERSISTENT_CACHE_SIZE (2)
```

(1) **rbd_persistent_cache_path** - 一个文件夹来缓存在使用 **rwl** 模式时必须启用直接访问 (DAX) 的数据，以避免性能下降。

(2) **rbd_persistent_cache_size** - 每个镜像的缓存大小，最小缓存大小为 1 GB。缓存越大，性能越好。

a. 为 **rwl** 模式设置额外的 RBD 选项：

示例

```
rbd_cache false
rbd_persistent_cache_mode rwl
rbd_plugins pwl_cache
rbd_persistent_cache_path /mnt/pmem/cache/
rbd_persistent_cache_size 1073741824
```

b. 为 **ssd** 模式设置额外的 RBD 选项：

示例

```
rbd_cache false
rbd_persistent_cache_mode ssd
rbd_plugins pwl_cache
rbd_persistent_cache_path /mnt/nvme/cache
rbd_persistent_cache_size 1073741824
```

其它资源

- 有关使用 DAX 的详情，请参阅 *kernel.org* 中的 *Direct Access* 文章。

2.19. 检查持久性写入日志缓存状态

您可以检查 Persistent Write Log (PWL) 缓存的状态。当获取专用锁定时会使用缓存，当专用锁定被释放时，持久性写入日志缓存会关闭。缓存状态显示有关缓存大小、位置、类型和其他与缓存相关的信息。在缓存打开和关闭时，执行对缓存状态的更新。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 监控节点的 root 级别访问权限。
- 启用 PWL 缓存正在运行的进程。

流程

- 查看 PWL 缓存状态：

  语法

  ```
  rbd status POOL_NAME/IMAGE_NAME
  ```

  示例

  ```
  [ceph: root@host01 /]# rbd status pool1/image1
  Watchers:
      watcher=10.10.0.102:0/1061883624 client.25496 cookie=140338056493088
  Persistent cache state:
      host: host02
      path: /mnt/nvme0/rbd-pwl.rbd.101e5824ad9a.pool
      size: 1 GiB
      mode: ssd
      stats_timestamp: Mon Apr 18 13:26:32 2022
      present: true    empty: false    clean: false
      allocated: 509 MiB
      cached: 501 MiB
      dirty: 338 MiB
      free: 515 MiB
      hits_full: 1450 / 61%
      hits_partial: 0 / 0%
      misses: 924
      hit_bytes: 192 MiB / 66%
      miss_bytes: 97 MiB
  ```

2.20. 清空持久写入日志缓存

您可以使用 **rbd** 命令清除缓存文件，指定 **persistent-cache flush**、池名称和镜像名称，然后再丢弃持久写入日志 (PWL) 缓存。**flush** 命令可将缓存文件显式写入 OSD。如果缓存中断或应用程序意外中断，缓存中的所有条目都会清除到 OSD，以便您可以手动清除数据，然后使缓存**无效**。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 监控节点的 root 级别访问权限。
- 启用 PWL 缓存。

流程

- 清除 PWL 缓存：

语法

```
rbd persistent-cache flush POOL_NAME/IMAGE_NAME
```

示例

```
[ceph: root@host01 /]# rbd persistent-cache flush pool1/image1
```

其它资源

- 如需了解更多详细信息，请参阅 *Red Hat Ceph Storage 块设备指南* 中的 *Discarding persistent write log cache* 部分。

2.21. 丢弃持久写入日志缓存

例如，如果缓存中的数据已过期，您可能需要手动丢弃 Persistent Write Log (PWL) 缓存。您可以使用 **rbd persistent-cache invalidate** 命令丢弃镜像的缓存文件。该命令删除指定镜像的缓存元数据，禁用缓存功能，并删除本地缓存文件（如果存在）。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 监控节点的 root 级别访问权限。
- 启用 PWL 缓存。

流程

- 丢弃 PWL 缓存：

  语法

  ```
  rbd persistent-cache invalidate POOL_NAME/IMAGE_NAME
  ```

  示例

  ```
  [ceph: root@host01 /]# rbd persistent-cache invalidate pool1/image1
  ```

2.22. 使用命令行界面监控 CEPH 块设备的性能

自 Red Hat Ceph Storage 4.1 开始，在 Ceph OSD 和管理器组件中集成了性能指标收集框架。此框架提供了一种内置方法，用于生成和处理构建其他 Ceph 块设备性能监控解决方案的性能指标。

新的 Ceph 管理器模块**rbd_support** 在启用时聚合性能指标。**rbd** 命令具有两个新操作：**iotop** 和 **iostat**。

注意

这些操作的初始使用可能需要大约 30 秒时间来填充数据字段。

先决条件

- 对 Ceph 监控节点的用户级别访问权限.

流程

1. 确保启用了 **rbd_support** Ceph Manager 模块：

   示例

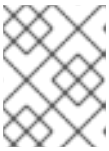
   ```
   [ceph: root@host01 /]# ceph mgr module ls

   {
       "always_on_modules": [
           "balancer",
           "crash",
           "devicehealth",
           "orchestrator",
           "pg_autoscaler",
           "progress",
           "rbd_support",   <--
           "status",
           "telemetry",
           "volumes"
   }
   ```

2. 显示"iotop"的镜像格式：

   示例

   ```
   [user@mon ~]$ rbd perf image iotop
   ```

   注意

   可以使用右箭头键对 write ops、read-ops、write-bytes、read-latency 和 read-latency 列进行动态排序。

3. 显示镜像的"iostat"样式：

   示例

   ```
   [user@mon ~]$ rbd perf image iostat
   ```

   注意

   此命令的输出可以是 JSON 或 XML 格式，然后使用其他命令行工具进行排序。

第 3 章 镜像实时迁移

作为存储管理员，您可以在不同的池之间实时迁移 RBD 镜像，甚至在同一存储集群中使用相同的池。您可以在不同的镜像格式和布局之间迁移，甚至从外部数据源迁移。当启动实时迁移时，源镜像会深度复制到目标镜像，拉取所有快照历史记录，同时保留可能数据的稀疏分配。

重要

目前，**krbd** 内核模块不支持实时迁移。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

3.1. 实时迁移过程

默认情况下，在使用相同存储集群的 RBD 镜像实时迁移过程中，源镜像被标记为只读。所有客户端将输入/输出 (I/O) 重定向到新目标镜像。另外，这个模式还可保留源镜像的父链接来保持稀疏性，或者它可以在迁移过程中扁平化镜像，以删除源镜像的父级依赖项。您可以在仅导入模式中使用实时迁移过程，其中源镜像没有修改。您可以将目标镜像链接到外部数据源，如备份文件、HTTP 文件或 S3 对象。使用新目标镜像时，实时迁移复制过程可以安全地在后台运行。

实时迁移过程包含三个步骤：

准备迁移 ：第一步是创建新的目标镜像，并将目标镜像链接到源镜像。如果没有配置仅导入模式，则源镜像也会链接到目标镜像并标记为只读。试图读取目标镜像中未初始化的数据扩展，将内部读取到源镜像，而写入目标镜像中未初始化的扩展将在内部深度副本中，重叠的镜像扩展会指向目标镜像。

执行迁移 ：这是一个后台操作，可深入地从源镜像中初始化的块到目标。当客户端正在使用新目标镜像时，您可以运行这个步骤。

完成迁移 ：您可以在后台迁移过程完成后提交或中止迁移。提交迁移会删除源和目标镜像之间的跨链接，并在仅导入模式下配置时删除源镜像。中止迁移移除跨链接，并删除目标镜像。

3.2. 格式

您可以使用 **native** 格式在 Red Hat Ceph Storage 集群中描述作为源镜像的原生 RBD 镜像。 **source-spec** JSON 文档以以下方式进行编码：

语法

```
{
    "type": "native",
    "pool_name": "POOL_NAME",
    ["pool_id": "POOL_ID",] (optional, alternative to "POOL_NAME" key)
    ["pool_namespace": "POOL_NAMESPACE",] (optional)
    "image_name": "IMAGE_NAME>",
    ["image_id": "IMAGE_ID",] (optional, useful if image is in trash)
    "snap_name": "SNAP_NAME",
    ["snap_id": "SNAP_ID",] (optional, alternative to "SNAP_NAME" key)
}
```

请注意，**native** 格式不包括 stream 对象，因为它使用了原生 Ceph 操作。例如，若要从镜像 **rbd/ns1/image1@snap1** 导入，可以使用 **source-spec** 进行编码：

示例

```
{
"type": "native",
"pool_name": "rbd",
"pool_namespace": "ns1",
"image_name": "image1",
"snap_name": "snap1"
}
```

您可以使用 **qcow** 格式描述 QEMU copy-on-write(QCOW)块设备。目前支持 QCOW v1 和 v2 格式，除了压缩、加密、备份文件和外部数据文件等高级功能外。您可以将 **qcow** 格式数据链接到任何支持的流源：

示例

```
{
    "type": "qcow",
    "stream": {
      "type": "file",
  "file_path": "/mnt/image.qcow"
    }
}
```

您可以使用 **raw** 格式来描述 thick-provisioned 原始块设备导出，它是 **rbd export –export-format 1 *SNAP_SPEC***。您可以将 **raw** 格式数据链接到任何支持的流源：

示例

```
{
    "type": "raw",
    "stream": {
      "type": "file",
      "file_path": "/mnt/image-head.raw"
    },
    "snapshots": [
        {
            "type": "raw",
            "name": "snap1",
            "stream": {
              "type": "file",
       "file_path": "/mnt/image-snap1.raw"
            }
        },
    ] (optional oldest to newest ordering of snapshots)
}
```

包括 **snapshots** 数组是可选的，目前只支持 thick-provisioned raw 快照导出。

3.3. 流

应用程序流

您可以使用**文件**流从本地可访问的 POSIX 文件源中导入。

语法

```
{
    <format unique parameters>
    "stream": {
        "type": "file",
        "file_path": "FILE_PATH"
    }
}
```

例如，要从位于 **/mnt/image.raw** 的文件导入 raw- 格式的镜像， **source-spec** JSON 文件是：

示例

```
{
    "type": "raw",
    "stream": {
        "type": "file",
        "file_path": "/mnt/image.raw"
    }
}
```

HTTP 流

您可以使用 **HTTP** 流从远程 HTTP 或 HTTPS web 服务器导入。

语法

```
{
    <format unique parameters>
    "stream": {
        "type": "http",
        "url": "URL_PATH"
    }
}
```

例如，要从位于 **http://download.ceph.com/image.raw** 的文件中导入 raw 格式的镜像， **source-spec** JSON 文件为：

示例

```
{
    "type": "raw",
    "stream": {
        "type": "http",
        "url": "http://download.ceph.com/image.raw"
    }
}
```

S3 流

您可以使用 **s3** 流从远程 S3 存储桶导入。

语法

-

```
{
    <format unique parameters>
    "stream": {
        "type": "s3",
        "url": "URL_PATH",
        "access_key": "ACCESS_KEY",
        "secret_key": "SECRET_KEY"
    }
}
```

例如，要从位于 **http://s3.ceph.com/bucket/image.raw** 的文件中导入 raw 格式的镜像，其 source-spec JSON 被编码如下：

示例

```
{
    "type": "raw",
    "stream": {
        "type": "s3",
        "url": "http://s3.ceph.com/bucket/image.raw",
        "access_key": "NX5QOQKC6BH2IDN8HC7A",
        "secret_key": "LnEsqNNqZIpkzauboDcLXLcYaWwLQ3Kop0zAnKIn"
    }
}
```

3.4. 准备实时迁移过程

您可以在同一 Red Hat Ceph Storage 集群中为 RBD 镜像准备默认的实时迁移过程。**rbd migration prepare** 命令接受与 **rbd create** 命令相同的所有布局选项。**rbd create** 命令允许更改不可变镜像的磁盘上的布局。如果您只想更改磁盘布局并希望保留原始镜像名称，请跳过 **migration_target** 参数。在准备实时迁移之前，所有使用源镜像的客户端都必须停止。如果准备步骤发现任何在读/写模式下打开的镜像的客户端，则**准**备步骤将失败。在**准**备步骤完成后，您可以使用新目标镜像重启客户端。

注意

您不能使用源镜像重启客户端，因为它会导致失败。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 两个块设备池。
- 一个块设备镜像。

流程

1. 在存储集群中准备实时迁移：

   语法

   ```
   rbd migration prepare SOURCE_POOL_NAME/SOURCE_IMAGE_NAME
   TARGET_POOL_NAME/SOURCE_IMAGE_NAME
   ```

示例

```
[ceph: root@rbd-client /]# rbd migration prepare sourcepool1/sourceimage1
targetpool1/sourceimage1
```

或者

如果要重命名源镜像：

语法

```
rbd migration prepare SOURCE_POOL_NAME/SOURCE_IMAGE_NAME
TARGET_POOL_NAME/NEW_SOURCE_IMAGE_NAME
```

示例

```
[ceph: root@rbd-client /]# rbd migration prepare sourcepool1/sourceimage1
targetpool1/newsourceimage1
```

在示例中，**newsourceimage1** 是重命名的源镜像。

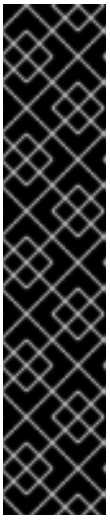
2. 您可以使用以下命令检查实时迁移过程的当前状态：

语法

```
rbd status TARGET_POOL_NAME/SOURCE_IMAGE_NAME
```

示例

```
[ceph: root@rbd-client /]# rbd status targetpool1/sourceimage1
Watchers: none
Migration:
source: sourcepool1/sourceimage1 (adb429cb769a)
destination: targetpool2/testimage1 (add299966c63)
state: prepared
```



重要

在迁移过程中，源镜像被移到 RBD 回收站中，以防止使用错误。

示例

```
[ceph: root@rbd-client /]# rbd info sourceimage1
rbd: error opening image sourceimage1: (2) No such file or directory
```

示例

```
[ceph: root@rbd-client /]# rbd trash ls --all sourcepool1
adb429cb769a sourceimage1
```

3.5. 准备只导入的迁移

您可以运行带有 **--import-only** 的 **rbd migration prepare** 命令来初始一个 **import-only** 实时迁移的过程，使用 **--source-spec** 或 **--source-spec-path** 选项直接通过命令行或通过一个文件传递一个描述如何访问源镜像的 JSON 文件。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 创建存储桶和 S3 对象。

流程

1. 创建 JSON 文件：

   示例

   ```
   [ceph: root@rbd-client /]# cat testspec.json
    {
      "type": "raw",
       "stream": {
           "type": "s3",
           "url": "http:10.74.253.18:80/testbucket1/image.raw",
          "access_key": "RLJOCP6345BGB38YQXI5",
          "secret_key": "oahWRB2ote2rnLy4dojYjDrsvaBADriDDgtSfk6o"
    }
   ```

2. 准备**仅导入**实时迁移过程：

   语法

   ```
   rbd migration prepare --import-only --source-spec-path "JSON_FILE"
   TARGET_POOL_NAME
   ```

   示例

   ```
   [ceph: root@rbd-client /]# rbd migration prepare --import-only --source-spec-path
   "testspec.json" targetpool1
   ```

   注意

   rbd migration prepare 命令接受与 **rbd create** 命令相同的所有镜像选项。

3. 您可以检查**仅导入**实时迁移的状态：

   示例

   ```
   [ceph: root@rbd-client /]# rbd status targetpool1/sourceimage1
   Watchers: none
   Migration:
   source: {"stream":
   {"access_key":"RLJOCP6345BGB38YQXI5","secret_key":"oahWRB2ote2rnLy4dojYjDrsvaBAD
   ```

```
riDDgtSfk6o","type":"s3","url":"http://10.74.253.18:80/testbucket1/image.raw"},"type":"raw"}
destination: targetpool1/sourceimage1 (b13865345e66)
state: prepared
```

3.6. 执行实时迁移过程

为实时迁移准备后，您必须将镜像块从源镜像复制到目标镜像。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 两个块设备池。
- 一个块设备镜像。

流程

1. 执行实时迁移：

   语法

   ```
   rbd migration execute TARGET_POOL_NAME/SOURCE_IMAGE_NAME
   ```

   示例

   ```
   [ceph: root@rbd-client /]# rbd migration execute targetpool1/sourceimage1
   Image migration: 100% complete...done.
   ```

2. 您可以检查迁移块 deep-copy 进程的进度的反馈：

   语法

   ```
   rbd status TARGET_POOL_NAME/SOURCE_IMAGE_NAME
   ```

   示例

   ```
   [ceph: root@rbd-client /]# rbd status targetpool1/sourceimage1
   Watchers: none
   Migration:
   source: sourcepool1/testimage1 (adb429cb769a)
   destination: targetpool1/testimage1 (add299966c63)
   state: executed
   ```

3.7. 提交实时迁移过程

您可以提交迁移，一旦实时迁移已完成，从源镜像中的所有数据块复制到目标镜像。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

- 两个块设备池。
- 一个块设备镜像。

流程

1. 提交迁移，在 deep-copying 完成后：

   语法

   ```
   rbd migration commit TARGET_POOL_NAME/SOURCE_IMAGE_NAME
   ```

   示例

   ```
   [ceph: root@rbd-client /]# rbd migration commit targetpool1/sourceimage1
   Commit image migration: 100% complete...done.
   ```

验证

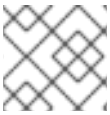
提交实时迁移将删除源和目标镜像之间的跨链接，同时从源池中移除源镜像：

示例

```
[ceph: root@rbd-client /]# rbd trash list --all sourcepool1
```

3.8. 中止实时迁移过程

您可以恢复实时迁移过程。中止实时迁移可恢复准备和执行步骤。



注意

仅当您尚未提交实时迁移时，才能中止。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 两个块设备池。
- 一个块设备镜像。

流程

1. 中止实时迁移过程：

   语法

   ```
   rbd migration abort TARGET_POOL_NAME/SOURCE_IMAGE_NAME
   ```

   示例

```
[ceph: root@rbd-client /]# rbd migration abort targetpool1/sourceimage1
Abort image migration: 100% complete...done.
```

验证

当实时迁移过程中止时，目标镜像会被删除并从源池中恢复对原始源镜像的访问：

示例

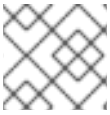
```
[ceph: root@rbd-client /]# rbd ls sourcepool1
sourceimage1
```

第 4 章 镜像加密

作为存储管理员，您可以设置用于加密特定 RBD 镜像的 secret 密钥。镜像级别的加密由 RBD 客户端在内部处理。

注意

krbd 模块不支持镜像级别加密。



注意

您可以使用外部工具（如 **dm-crypt** 或 **QEMU** ）来加密 RBD 镜像。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 7 集群。
- **root** 级别权限。

4.1. 加密格式

默认情况下不加密 RBD 镜像。您可以格式化为其中一个支持的加密格式来加密 RBD 镜像。格式操作将加密元数据保留在 RBD 镜像。加密元数据包含加密格式和版本、密码算法和模式规格等信息，以及用于保护加密密钥的信息。

加密密钥受用户保存的密码保护，该密码不会作为持久数据存储在 RBD 镜像中。加密格式操作要求您指定加密格式、密码算法和模式规格以及密码短语。加密元数据存储在 RBD 镜像中，目前是作为在原始镜像开始时编写的加密标头。这意味着加密镜像的有效镜像大小会小于原始镜像大小。

注意

除非明确（重新）格式化加密镜像，否则加密镜像的克隆本质上使用相同的格式和 secret 加密。

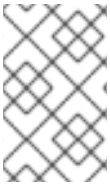
注意

在格式化之前写入 RBD 镜像的任何数据都可能会变得不可读取，即使它可能仍然占用存储资源。启用日志功能的 RBD 镜像无法加密。

4.2. 加密加载

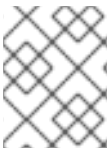
默认情况下，所有 RBD API 都会像未加密的 RBD 镜像一样对待加密的 RBD 镜像。您可以在镜像的任意位置读取或写入原始数据。在镜像中写入原始数据可能会使加密格式的完整性风险。例如，原始数据可能会覆盖位于镜像开头的加密元数据。要安全地对加密的输入/输出(I/O)或维护操作对加密的 RBD 镜像执行，必须在打开镜像后立即应用额外的加密负载操作。

加密负载操作要求您指定加密格式和密码短语来解锁镜像本身的加密密钥及其明确格式化的镜像。已打开的 RBD 镜像的所有 I/O 都是加密或解密克隆的 RBD 镜像，这包括父镜像的 IO。加密密钥由 RBD 客户端保存在内存中，直到镜像关闭为止。



注意

在 RBD 镜像中载入加密后，就无法应用其他加密加载或格式操作。此外，使用打开的镜像上下文检索 RBD 镜像大小和父级重叠的 API 调用会分别返回有效的镜像大小以及有效的父重叠。当 RBD 镜像通过 **rbd-nbd** 将 RBD 镜像映射为块设备时，会自动载入加密。



注意

使用打开的镜像上下文检索镜像大小和父级重叠的 API 调用会返回有效的镜像大小以及有效的父级重叠。

注意

如果加密镜像的克隆被明确格式化，则扁平化或缩减克隆的镜像被透明，因为父数据必须根据克隆的镜像格式重新加密，因为它会从父快照复制。如果在发布 flatten 操作前没有加载加密，则之前在克隆的镜像中访问的任何父数据都可能会变得不可读。

注意

如果明确格式化了加密镜像的克隆，则操作将克隆的镜像缩小为透明。这是因为，在包含快照的克隆镜像或克隆的镜像与对象大小不对齐的情况下，从父快照复制一些数据的操作，类似于扁平化。如果在处理缩小操作前没有加载加密，则之前在克隆的镜像中访问的任何父数据都可能会变得不可读。

4.3. 支持的格式

支持 Linux 统一密钥设置(LUKS)1 和 2。数据布局完全符合 LUKS 规格。**dm-crypt** 或 **QEMU** 等外部 LUKS 兼容工具可以在加密的 RBD 镜像上安全地执行加密的输入/输出(I/O)。另外，您可以通过将原始 LUKS 数据复制到 RBD 镜像来导入外部工具创建的现有 LUKS 镜像。

目前，只支持高级加密标准(AES)128 和 256 加密算法。xts-plain64 目前是唯一支持的加密模式。

要使用 LUKS 格式，请使用以下命令格式化 RBD 镜像：

注意

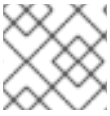
您需要创建一个名为 **passphrase.txt** 的文件并输入密码短语。您可以随机生成密码短语，该密码短语可能包含 NULL 字符。如果密码短语以换行符结尾，它将被剥离。

语法

```
rbd encryption format POOL_NAME/LUKS_IMAGE luks1|luks2 PASSPHRASE_FILE
```

示例

```
[ceph: root@host01 /]# rbd encryption format pool1/luksimage1 luks1 passphrase.bin
```



注意

您可以选择 **luks1** 或 **luks** 加密格式。

加密格式操作生成 LUKS 标头，并将它写入到 RBD 镜像开始时。在标头后面会附加一个 keylot。keyslot

包含随机生成的加密密钥，并由从密码短语读取的密码短语进行保护。默认情况下，xts-plain64 模式的 AES-256（当前推荐的模式）和默认用于其他 LUKS 工具。目前不支持添加或删除额外的密码短语，但可以使用 **cryptsetup** 等工具实现。LUKS 标头大小可能会有所不同，对于 LUKS 最多为 136MiB，但取决于安装的 **libcryptsetup** 版本，它通常最多为 16MiB。为获得最佳性能，加密格式设置数据偏移，使其与镜像对象大小保持一致。例如，在使用配置了 8MiB 对象大小的镜像时，至少需要 8MiB 的开销。

在 LUKS1 中，扇区（最小加密单元）的固定单位为 512 字节。LUKS2 支持较大的扇区，并且获得更好的性能，默认扇区大小设置为最大 4KiB。小于扇区的写入或与扇区启动不一致的写入，请在客户端上触发保护的 **read-modify-write** 链，并具有相当延迟的损失。此类未对齐写入的批处理可能会导致 I/O 争用，从而进一步降低性能。红帽建议在无法保证以 LUKS 扇区一致而使用 RBD 加密时避免使用 RBD 加密。

要映射 LUKS 加密的镜像，请运行以下命令：

语法

```
rbd device map -t nbd -o encryption-format=luks1|luks2,encryption-passphrase-file=passphrase.txt POOL_NAME/LUKS_IMAGE
```

示例

```
[ceph: root@host01 /]# rbd device map -t nbd -o encryption-format=luks1,encryption-passphrase-file=passphrase.txt pool1/luksimage1
```

注意

您可以选择 **luks1** 或 **luks2** 加密格式。

注意

为安全起见，加密格式和加密负载操作都是 CPU 密集型，可能需要几秒钟才能完成。对于加密的 I/O，假设启用了 AES-NI，可能会添加相对较小的微秒延迟，以及 CPU 使用率小的增加。

4.4. 在镜像和克隆中添加加密格式

支持分层客户端加密。克隆的镜像可以使用自己的格式和密码短语进行加密，可能与父镜像的不同。

使用 **rbd encryption format** 命令将加密格式添加到镜像和克隆。给定 LUKS2 格式镜像，您可以创建一个 LUKS2 格式的克隆和 LUKS1- 格式克隆。

先决条件

- 正在运行的 Red Hat Ceph Storage 集群配置有块设备(RBD)。
- 节点的根级别访问权限。

流程

1. 创建 LUKS2- 格式的镜像：

   语法

   ```
   rbd create --size SIZE POOL_NAME/LUKS_IMAGE
   ```

```
rbd encryption format POOL_NAME/LUKS_IMAGE luks1|luks2 PASSPHRASE_FILE
rbd resize --size 50G --encryption-passphrase-file PASSPHRASE_FILE
POOL_NAME/LUKS_IMAGE
```

示例

```
[ceph: root@host01 /]# rbd create --size 50G mypool/myimage
[ceph: root@host01 /]# rbd encryption format mypool/myimage luks2 passphrase.txt
[ceph: root@host01 /]# rbd resize --size 50G --encryption-passphrase-file passphrase.txt
mypool/myimage
```

rbd resize 命令会增加镜像，以补偿与 LUKS2 标头关联的开销。

2. 使用 LUKS2- 格式的镜像，创建具有相同有效大小的 LUKS2- 格式的克隆：

语法

```
rbd snap create POOL_NAME/IMAGE_NAME@SNAP_NAME
rbd snap protect POOL_NAME/IMAGE_NAME@SNAP_NAME
rbd clone POOL_NAME/IMAGE_NAME@SNAP_NAME POOL_NAME/CLONE_NAME
rbd encryption format POOL_NAME/CLONE_NAME luks1 CLONE_PASSPHRASE_FILE
```

示例

```
[ceph: root@host01 /]# rbd snap create mypool/myimage@snap
[ceph: root@host01 /]# rbd snap protect mypool/myimage@snap
[ceph: root@host01 /]# rbd clone mypool/myimage@snap mypool/myclone
[ceph: root@host01 /]# rbd encryption format mypool/myclone luks1 clone-passphrase.bin
```

3. 使用 LUKS2- 格式的镜像，创建一个具有相同有效大小的 LUKS1- 格式克隆：

语法

```
rbd snap create POOL_NAME/IMAGE_NAME@SNAP_NAME
rbd snap protect POOL_NAME/IMAGE_NAME@SNAP_NAME
rbd clone POOL_NAME/IMAGE_NAME@SNAP_NAME POOL_NAME/CLONE_NAME
rbd encryption format POOL_NAME/CLONE_NAME luks1 CLONE_PASSPHRASE_FILE
rbd resize --size SIZE --allow-shrink --encryption-passphrase-file
CLONE_PASSPHRASE_FILE --encryption-passphrase-file PASSPHRASE_FILE
POOL_NAME/CLONE_NAME
```

示例

```
[ceph: root@host01 /]# rbd snap create mypool/myimage@snap
[ceph: root@host01 /]# rbd snap protect mypool/myimage@snap
[ceph: root@host01 /]# rbd clone mypool/myimage@snap mypool/myclone
[ceph: root@host01 /]# rbd encryption format mypool/myclone luks1 clone-passphrase.bin
[ceph: root@host01 /]# rbd resize --size 50G --allow-shrink --encryption-passphrase-file
clone-passphrase.bin --encryption-passphrase-file passphrase.bin mypool/myclone
```

由于 LUKS1 标头通常小于 LUKS2 标头，所以末尾的 **rbd resize** 命令会缩小克隆的镜像来获取不需要的空间允许。

4. 使用 LUKS-1- 格式的镜像，创建具有相同有效大小的 LUKS2- 格式的克隆：

语法

```
rbd resize --size SIZE POOL_NAME/LUKS_IMAGE
rbd snap create POOL_NAME/IMAGE_NAME@SNAP_NAME
rbd snap protect POOL_NAME/IMAGE_NAME@SNAP_NAME
rbd clone POOL_NAME/IMAGE_NAME@SNAP_NAME POOL_NAME/CLONE_NAME
rbd encryption format POOL_NAME/CLONE_NAME luks2 CLONE_PASSPHRASE_FILE
rbd resize --size SIZE --allow-shrink --encryption-passphrase-file PASSPHRASE_FILE
POOL_NAME/LUKS_IMAGE
rbd resize --size SIZE --allow-shrink --encryption-passphrase-file
CLONE_PASSPHRASE_FILE --encryption-passphrase-file PASSPHRASE_FILE
POOL_NAME_/CLONE_NAME
```

示例

```
[ceph: root@host01 /]# rbd resize --size 51G mypool/myimage
[ceph: root@host01 /]# rbd snap create mypool/myimage@snap
[ceph: root@host01 /]# rbd snap protect mypool/myimage@snap
[ceph: root@host01 /]# rbd clone mypool/my-image@snap mypool/myclone
[ceph: root@host01 /]# rbd encryption format mypool/myclone luks2 clone-passphrase.bin
[ceph: root@host01 /]# rbd resize --size 50G --allow-shrink --encryption-passphrase-file
passphrase.bin mypool/myimage
[ceph: root@host01 /]# rbd resize --size 50G --allow-shrink --encryption-passphrase-file
clone-passphrase.bin --encryption-passphrase-file passphrase.bin mypool/myclone
```

由于 LUKS2 标头通常大于 LUKS1 标头，所以开始时的 **rbd resize** 命令会临时增加父镜像，以在父快照中保留一些额外的空间，因此克隆的镜像。这是在克隆的镜像中访问所有父数据所必需的。末尾的 **rbd resize** 命令会将父镜像缩小到其原始大小，不会影响父快照和克隆的镜像，从而获得未使用的保留空间

这同样适用于创建未格式化的镜像的格式克隆，因为未格式化的镜像根本没有标头。

其它资源

- 如需了解有关 *将客户端添加到 **cephadm-ansible** 清单* 的更多详细信息，请参阅 *Red Hat Ceph Storage 安装指南中的配置* Ansible 清单位置部分。

第 5 章 管理快照

作为存储管理员，熟悉 Ceph 的快照功能可帮助您管理存储在 Red Hat Ceph Storage 集群中的镜像的快照和克隆。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

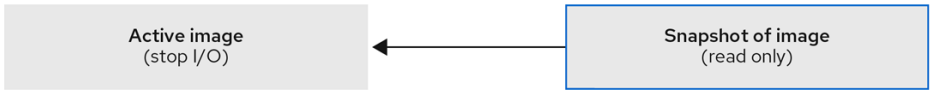
5.1. CEPH 块设备快照

快照是镜像在特定时间点上状态的只读副本。Ceph 块设备的其中一个高级功能是您可以创建镜像的快照来保留镜像状态的历史记录。Ceph 也支持快照分层，允许您快速轻松地克隆镜像，例如虚拟机镜像。Ceph 支持利用 **rbd** 命令和许多更高级别的接口进行块设备快照，包括 **QEMU**、**libvirt**、OpenStack 和 CloudStack。

注意

如果在有 **I/O** 操作时进行快照，则快照可能无法获取镜像的准确或最新的数据，并且快照可能需要克隆到一个信的、可以挂载的映像。红帽建议在进行快照前，停止 **I/O**。如果镜像包含文件系统，则执行快照之前文件系统必须处于一致状态。您可以使用 **fsfreeze** 命令停止 **I/O**。对于虚拟机，**qemu-guest-agent** 可用于在创建快照时自动冻结文件系统。

图 5.1. Ceph 块设备**快照**



其它资源

- 详情请查看 **fsfreeze(8)** 手册页。

5.2. CEPH 用户和密钥环

启用 **cephx** 后，您必须指定用户名或 ID，以及包含用户对应密钥的密钥环的路径。

注意

cephx 默认启用。

您还可以添加 **CEPH_ARGS** 环境变量以避免重新输入以下参数：

语法

```
rbd --id USER_ID --keyring=/path/to/secret [commands]
rbd --name USERNAME --keyring=/path/to/secret [commands]
```

示例

```
[root@rbd-client ~]# rbd --id admin --keyring=/etc/ceph/ceph.keyring [commands]
[root@rbd-client ~]# rbd --name client.admin --keyring=/etc/ceph/ceph.keyring [commands]
```

提示

将用户和 secret 添加到 **CEPH_ARGS** 环境变量，以便您无需每次输入它们。

5.3. 创建块设备快照

创建 Ceph 块设备的快照。

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. 指定 **snap create** 选项、池名称和镜像名称：
   - 方法 1：

     语法

     ```
     rbd --pool POOL_NAME snap create --snap SNAP_NAME IMAGE_NAME
     ```

     示例

     ```
     [root@rbd-client ~]# rbd --pool pool1 snap create --snap snap1 image1
     ```

   - 方法 2：

     语法

     ```
     rbd snap create POOL_NAME/IMAGE_NAME@SNAP_NAME
     ```

     示例

     ```
     [root@rbd-client ~]# rbd snap create pool1/image1@snap1
     ```

5.4. 列出块设备快照

列出块设备快照。

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. 指定池名称和镜像名称：

   语法

   ```
   rbd --pool POOL_NAME --image IMAGE_NAME snap ls
   rbd snap ls POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd --pool pool1 --image image1 snap ls
   [root@rbd-client ~]# rbd snap ls pool1/image1
   ```

5.5. 回滚块设备快照

回滚块设备快照。

注意

将镜像回滚到快照意味着使用快照中的数据覆盖镜像的当前版本。执行回滚所需的时间会随着镜像大小的增加而增加。从快照**克隆快于**将镜像**回**滚到照要，这是返回到预先存在状态的首选方法。

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. 指定 **snap rollback** 选项、池名称、镜像名称和快照名称：

   语法

   ```
   rbd --pool POOL_NAME snap rollback --snap SNAP_NAME IMAGE_NAME
   rbd snap rollback POOL_NAME/IMAGE_NAME@SNAP_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd --pool pool1 snap rollback --snap snap1 image1
   [root@rbd-client ~]# rbd snap rollback pool1/image1@snap1
   ```

5.6. 删除块设备快照

删除 Ceph 块设备的快照。

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. 要删除块设备快照，请指定 **snap rm** 选项、池名称、镜像名称和快照名称：

   语法

   ```
   rbd --pool POOL_NAME snap rm --snap SNAP_NAME IMAGE_NAME
   rbd snap rm POOL_NAME-/IMAGE_NAME@SNAP_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd --pool pool1 snap rm --snap snap2 image1
   [root@rbd-client ~]# rbd snap rm pool1/image1@snap1
   ```

重要

如果镜像具有任何克隆，克隆的镜像会保留对父镜像快照的引用。要删除父镜像快照，您必须首先扁平化子镜像。

注意

Ceph OSD 守护进程异步删除数据，因此删除快照不会立即释放磁盘空间。

其它资源

- 详情请参阅 *Red Hat Ceph Storage 块设备指南* 中的*扁平化克隆镜像*。

5.7. 清除块设备快照

清除块设备快照。

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. 指定 **snap purge** 选项以及特定池中的镜像名称：

   语法

   ```
   rbd --pool POOL_NAME snap purge IMAGE_NAME
   rbd snap purge POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd --pool pool1 snap purge image1
   [root@rbd-client ~]# rbd snap purge pool1/image1
   ```

5.8. 重命名块设备快照

重新命名块设备快照。

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. 重新命名快照：

   语法

   ```
   rbd snap rename POOL_NAME/IMAGE_NAME@ORIGINAL_SNAPSHOT_NAME POOL_NAME/IMAGE_NAME@NEW_SNAPSHOT_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd snap rename data/dataset@snap1 data/dataset@snap2
   ```

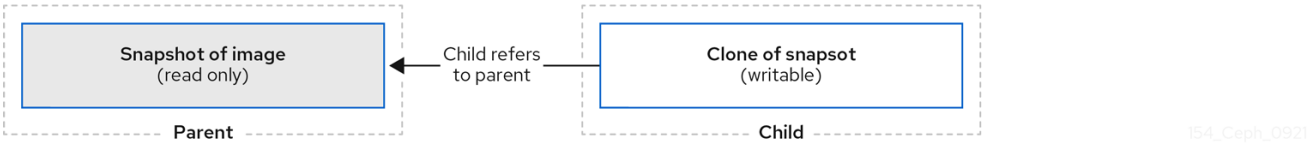
   这会将 **data** 池上 **dataset** 镜像的 **snap1** 快照重命名为 **snap2**。

2. 执行 **rbd help snap rename** 命令，以显示重命名快照的更多详细信息。

5.9. CEPH 块设备分层

Ceph 支持创建许多块设备快照的写时复制 (COW) 或读时复制 (COR) 克隆。快照分层使得 Ceph 块设备客户端能够非常快速地创建镜像。例如，您可以使用写入它的 Linux 虚拟机创建块设备镜像。然后，对镜像执行快照，保护快照，并创建所需数量的克隆。快照是只读的，因此克隆快照可以简化语义-使快速创建克隆成为可能。

图 5.2. Ceph 块设备分层



注意

术语 **父项（parent）**和**子项（child）**表示 Ceph 块设备快照、父项，以及从快照子级克隆的对应映像。这些术语对于以下命令行用法非常重要。

每个克隆的镜像（子镜像）存储对其父镜像的引用，这使得克隆的镜像能够打开父快照并读取它。当克隆**扁平化**时，当快照中的信息完全复制到克隆时，会删除此引用。

快照克隆的行为与任何其他 Ceph 块设备镜像完全相同。您可以读取、写入、克隆和调整克隆的镜像大小。克隆的镜像没有特殊限制。但是，快照的克隆会指向快照，因此在克隆快照前，**必须**会对其进行保护。

快照的克隆可以是写时复制 (COW) 或读时复制 (COR) 克隆。在必须显式启用读取时复制 (COR) 时，始终为克隆启用写时复制 (COW)。当数据写入到克隆中的未分配对象时，写时复制 (COW) 将数据从父项复制到克隆。当父进程从克隆中未分配的对象读取时，从父进程复制数据到克隆。如果克隆中尚不存在对象，则仅从父项读取数据。RADOS 块设备将大型镜像分成多个对象。默认值为 4 MB，所有写时复制 (COW) 和写时复制 (COR) 操作都发生在完整的对象上，这会将 1 字节写入到克隆，如果之前的 COW/COR 操作的克隆中目标对象尚不存在，则会导致从父对象读取 4 MB 对象并写入克隆。

是否启用读取时复制 (COR)，任何通过从克隆读取底层对象无法满足的读取都将重新路由到父对象。由于父项实际上没有限制，这意味着您可以对一个克隆进行克隆，因此，在找到对象或您到达基础父镜像时，这个重新路由将继续进行。如果启用了读取时复制 (COR)，克隆中任何未直接满足的读取会导致从父项读取完整的对象并将该数据写入克隆，以便克隆本身可以满足相同的扩展读取，而无需从父级读取。

这基本上是一个按需、按对象扁平化的操作。当克隆位于与它的父级（位于另一个地理位置）的父级（位于其他地理位置）的高延迟连接中时，这特别有用。读时复制 (COR) 可降低读分化延迟。前几次读取具有较高的延迟，因为它将导致从父进程读取额外的数据，例如，您从克隆中读取 1 字节，但现在 4 MB 必须从父级读取并写入克隆，但将来的所有读取都将从克隆本身提供。

要从快照创建写时复制 (COR) 克隆，您必须通过在 **ceph.conf** 文件的 **[global]** 或 **[client]** 部分添加 **rbd_clone_copy_on_read = true** 来显式启用此功能。

其它资源

- 有关**扁平化**的更多信息，请参阅 *Red Hat Ceph Storage Block Gudie* 中的*扁平克隆镜像*部分。

5.10. 保护块设备快照

克隆访问父快照。如果用户意外删除父快照，则所有克隆都会中断。

您可以将 **set-require-min-compat-client** 参数设置为大于或等于 Ceph 的 mimic 版本。

示例

```
ceph osd set-require-min-compat-client mimic
```

这会默认创建克隆 v2。但是，比 mimic 旧的客户端无法访问这些块设备镜像。

注意

克隆 v2 不需要保护快照。

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

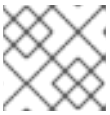
1. 在以下 命令中指定 ***POOL_NAME***、***IMAGE_NAME*** 和 ***SNAP_SHOT_NAME*** ：

   语法

   ```
   rbd --pool POOL_NAME snap protect --image IMAGE_NAME --snap SNAPSHOT_NAME
   rbd snap protect POOL_NAME/IMAGE_NAME@SNAPSHOT_NAME
   ```

示例

```
[root@rbd-client ~]# rbd --pool pool1 snap protect --image image1 --snap snap1
[root@rbd-client ~]# rbd snap protect pool1/image1@snap1
```



注意

您无法删除受保护的快照。

5.11. 克隆块设备快照

克隆块设备快照，以在同一个池或其他池中创建快照的读取或写入子镜像。一种用例是将只读镜像和快照维护为一个池中的模板，然后在另一个池中维护可写克隆。

注意

克隆 v2 不需要保护快照。

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. 要克隆快照，您需要指定父池、快照、子池和镜像名称：

语法

```
rbd snap --pool POOL_NAME --image PARENT_IMAGE --snap SNAP_NAME --dest-pool
POOL_NAME --dest CHILD_IMAGE_NAME
rbd clone POOL_NAME/PARENT_IMAGE@SNAP_NAME
POOL_NAME/CHILD_IMAGE_NAME
```

示例

```
[root@rbd-client ~]# rbd clone --pool pool1 --image image1 --snap snap2 --dest-pool pool2 --
dest childimage1
[root@rbd-client ~]# rbd clone pool1/image1@snap1 pool1/childimage1
```

5.12. 取消保护块设备快照

您必须先取消保护快照，然后才能删除快照。此外，您*不得*删除从克隆引用的快照。您必须扁平化快照的每个克隆，然后才能删除快照。

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. 运行以下命令：

   语法

   ```
   rbd --pool POOL_NAME snap unprotect --image IMAGE_NAME --snap SNAPSHOT_NAME
   rbd snap unprotect POOL_NAME/IMAGE_NAME@SNAPSHOT_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd --pool pool1 snap unprotect --image image1 --snap snap1

   [root@rbd-client ~]# rbd snap unprotect pool1/image1@snap1
   ```

5.13. 列出快照的子项

列出快照的子项。

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. 要列出快照的子项，请执行以下操作：

   语法

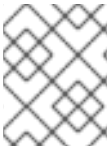
   ```
   rbd --pool POOL_NAME children --image IMAGE_NAME --snap SNAP_NAME
   rbd children POOL_NAME/IMAGE_NAME@SNAPSHOT_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd --pool pool1 children --image image1 --snap snap1
   [root@rbd-client ~]# rbd children pool1/image1@snap1
   ```

5.14. 扁平化克隆的镜像

克隆的镜像保留对父快照的引用。当您从子克隆删除引用到父快照时，您有效地通过将信息从快照复制到克隆来"扁平化"镜像。随着快照大小的增加，扁平化克隆所需的时间会增加。由于扁平化的镜像包含快照的所有信息，因此扁平化的镜像将占用比分层克隆更多的存储空间。



注意

如果镜像上启用 **深度扁平化（deep flatten）** 功能，则默认情况下镜像克隆与其父级解除关联。

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. 要删除与子镜像关联的父镜像快照，您必须首先扁平化子镜像：

   语法

   ```
   rbd --pool POOL_NAME flatten --image IMAGE_NAME
   rbd flatten POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd --pool pool1 flatten --image childimage1
   [root@rbd-client ~]# rbd flatten pool1/childimage1
   ```

第 6 章 镜像 CEPH 块设备

作为存储管理员，您可以通过镜像 Red Hat Ceph Storage 集群之间的数据镜像，为 Ceph 块设备添加另一层冗余。了解和使用 Ceph 块设备镜像功能可帮助您防止数据丢失，如站点故障。镜像 Ceph 块设备有两种配置，单向镜像或双向镜像，您可以在池和单个镜像上配置镜像功能。

先决条件

- 至少运行两个健康的 Red Hat Ceph Storage 集群。
- 两个存储集群之间的网络连接。
- 为每个 Red Hat Ceph Storage 集群访问 Ceph 客户端节点。
- 具有管理员级别功能的 CephX 用户。

6.1. CEPH 块设备镜像

RADOS 块设备 (RBD) 镜像是在两个或多个 Ceph 存储集群之间异步复制 Ceph 块设备镜像的过程。通过在不同的地理位置查找 Ceph 存储集群，RBD 镜像功能可帮助您从站点灾难中恢复。基于日志的 Ceph 块设备镜像可确保镜像所有更改的时间点一致性副本，包括读取和写入、块设备调整大小、快照、克隆和扁平化。

RBD 镜像使用专用锁定和日志记录功能，按照镜像发生的顺序记录对镜像的所有修改。这样可确保镜像的崩溃一致性镜像可用。

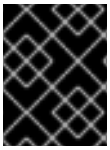
重要

支持镜像块设备镜像的主要和次要池的 CRUSH 层次结构必须具有相同的容量和性能特性，并且必须具有足够的带宽才能确保镜像无延迟。例如，如果您的主存储集群中有 X MB/s 平均写入吞吐量，则网络必须支持连接至次要站点的 N * X 吞吐量，以及 Y% 用于镜像 N 镜像的安全因子。

rbd-mirror 守护进程负责通过从远程主镜像拉取更改，将镜像从一个 Ceph 存储集群同步到另一个 Ceph 存储集群，并将这些更改写入本地的非主镜像。**rbd-mirror** 守护进程可以在单个 Ceph 存储集群上运行，实现单向镜像功能，也可以在两个 Ceph 存储集群上运行，以实现参与镜像关系的双向镜像。

要使 RBD 镜像工作（可使用单向复制或双向复制），进行几个假设：

- 两个存储集群中都存在一个名称相同的池。
- 池包含您要镜像的启用了日志的镜像。



重要

在单向或双向复制中，**rbd-mirror** 的每个实例必须能够同时连接其他 Ceph 存储集群。此外，两个数据中心站点之间网络必须具有足够的带宽才能处理镜像。

单向复制（One-way Replication）

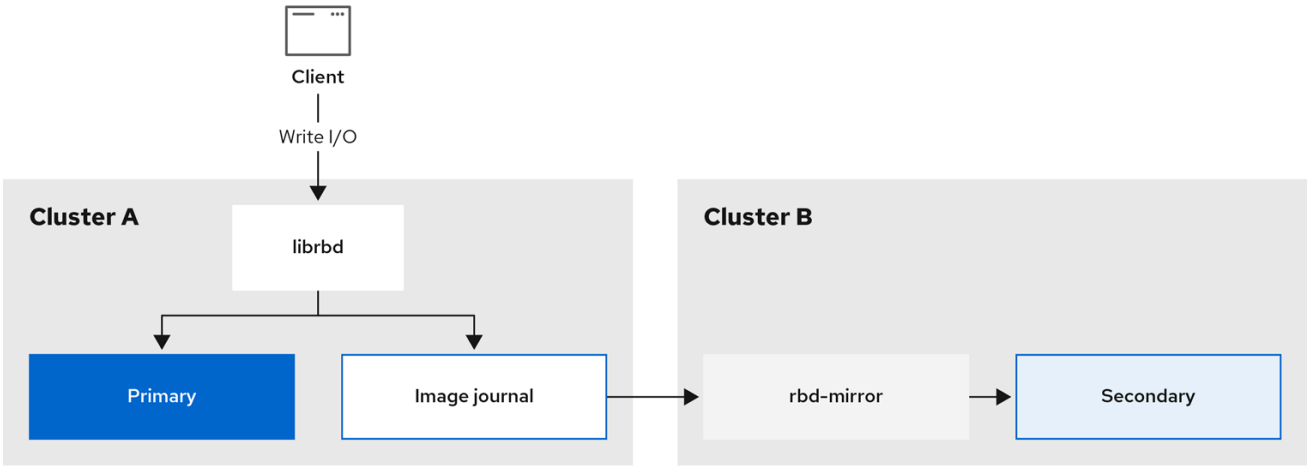
单向镜像意味着一个存储集群中的主要镜像或镜像池会被复制到次要存储集群。单向镜像还支持复制到多个次要存储集群。

在辅助存储群集上，镜像是非主要复制；即 Ceph 客户端无法写入镜像。当数据从主存储集群镜像到次要存储集群时，**rbd-mirror** 只在次要存储集群上运行。

为了进行单向镜像工作，应进行几项假设：

- 您有两个 Ceph 存储集群，希望将镜像从主存储集群复制到辅助存储集群。
- 辅助存储集群附加有运行 **rbd-mirror** 守护进程的 Ceph 客户端节点。**rbd-mirror** 守护进程将连接到主存储集群，将镜像同步到次要存储集群。

图 6.1. 单向镜像



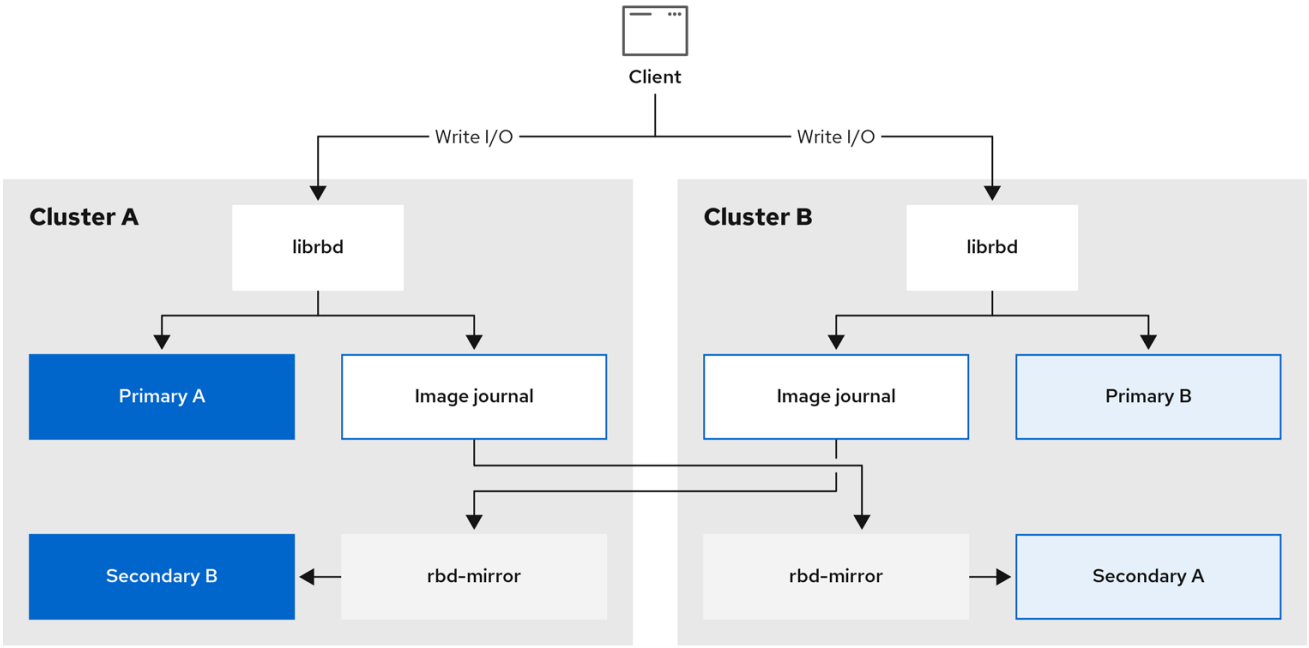
双向复制（Two-way Replication）

双向复制在主集群中添加一个 **rbd-mirror** 守护进程，使得镜像可以在集群上降级并提升到次要集群中。然后可以对次要群集上的镜像进行更改，然后按照相反方向（从次要到主要）进行复制。两个集群都必须运行 **rbd-mirror**，才能在任一集群上提升和降级镜像。目前，仅在两个站点间支持双向复制。

要进行双向镜像工作，请进行几项假设：

- 您有两个存储集群，希望在它们之间以任一方向复制镜像。
- 两个存储集群都附加了一个客户端节点，它们运行 **rbd-mirror** 守护进程。次要存储集群上运行的 **rbd-mirror** 守护进程将连接到主存储集群，将镜像同步到次要存储集群，而主存储集群上运行的 **rbd-mirror** 守护进程将连接到次要存储集群，将镜像同步到主要位置。

图 6.2. 双向镜像



镜像模式

镜像以每个池为基础配置，带有镜像对等存储集群。Ceph 支持两种镜像模式，具体取决于池中镜像的类型。

池模式

启用了日志记录功能的池中的所有镜像都会被镜像(mirror)。

镜像模式

只有池中的特定镜像子集才会被镜像(mirror)。您必须为每个镜像单独启用镜像功能。

镜像状态

镜像是否可以修改取决于其状态：

- 可以修改处于主要状态的镜像。
- 处于非主要状态的镜像无法修改。

镜像在镜像上首次启用镜像时自动提升为主版本。升级可能发生：

- 通过在池模式中启用镜像来隐式执行镜像。
- 通过启用特定镜像的镜像来显式启用。

可以降级主镜像并提升非主镜像。

其它资源

- 详情请参阅 *Red Hat Ceph Storage Block Device 指南* 中的 *对池启用镜像*部分。
- 如需了解更多详细信息，请参阅 *Red Hat Ceph Storage 块设备指南* 中的*启用镜像镜像* 部分。
- 如需了解更多详细信息，请参阅 *Red Hat Ceph Storage 块设备指南* 中的*镜像提升和降级*小节。

6.1.1. 基于日志和基于快照的镜像概述

RADOS 块设备(RBD)镜像可以通过两种模式在两个 Red Hat Ceph Storage 集群间异步镜像：

基于日志的镜像

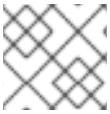
此模式使用 RBD 日志镜像功能来确保两个 Red Hat Ceph Storage 集群之间的特定时间点和崩溃时的复制一致性。实际镜像不会被修改，直到每个写入 RBD 镜像都记录到相关的日志中。远程集群从这个日志中读取，并将更新重新显示到其镜像的本地副本。由于对 RBD 镜像的每个写入会导致对 Ceph 集群进行两个写入，因此写入延迟几乎会加倍，使用 RBD 日志镜像功能。

基于快照的镜像

此模式使用定期调度或手动创建 RBD 镜像镜像快照，在两个 Red Hat Ceph Storage 集群间复制一致的 RBD 镜像。远程集群决定两个镜像快照之间的任何数据或元数据更新，并将 deltas 复制到其镜像的本地副本。RBD **fast-diff** 镜像功能可快速确定更新的数据块，而无需扫描完整的 RBD 镜像。在故障转移场景中使用前，需要同步两个快照之间的完整 delta。在故障转移时，任何部分应用的 deltas 都会被回滚。

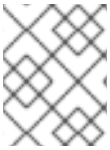
6.2. 使用命令行界面配置单向镜像

此流程配置池从主存储集群到辅助存储集群的单向复制。



注意

使用单向复制时，您可以镜像到多个次要存储集群。



注意

本节中的示例通过将主镜像作为 **site-a** 引用主存储集群和您将要复制镜像的辅助存储集群作为 **site-b** 来区分两个存储集群。这些示例中使用的池名称称为 **data**。

先决条件

- 至少两个健康状态并运行 Red Hat Ceph Storage 集群。
- 对每个存储集群的 Ceph 客户端节点的根级别访问权限。
- 具有管理员级别功能的 CephX 用户。

流程

1. 在两个站点上登录到 **cephadm** shell：

   示例

   ```
   [root@site-a ~]# cephadm shell
   [root@site-b ~]# cephadm shell
   ```

2. 在 **site-b** 上，调度在二级集群中部署 mirror 守护进程：

   语法

   ```
   ceph orch apply rbd-mirror --placement=NODENAME
   ```

示例

```
[ceph: root@site-b /]# ceph orch apply rbd-mirror --placement=host04
```

注意

nodename 是要在二级集群中配置镜像的主机。

3. 在 **site-a** 上的镜像上启用日志功能。

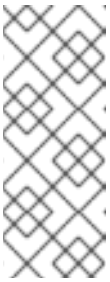
   a. 对于**新镜像**，使用 **--image-feature** 选项：

   语法

   ```
   rbd create IMAGE_NAME --size MEGABYTES --pool POOL_NAME --image-feature
   FEATURE FEATURE
   ```

   示例

   ```
   [ceph: root@site-a /]# rbd create image1 --size 1024 --pool data --image-feature
   exclusive-lock,journaling
   ```

   注意

   如果已经启用了 **exclusive-lock**，则使用 **journaling** 作为唯一参数，否则会返回以下错误：

   ```
   one or more requested features are already enabled
   (22) Invalid argument
   ```

   b. 对于**现有镜像**，请使用 **rbd feature enable** 命令：

   语法

   ```
   rbd feature enable POOL_NAME/IMAGE_NAME FEATURE, FEATURE
   ```

   示例

   ```
   [ceph: root@site-a /]# rbd feature enable data/image1 exclusive-lock, journaling
   ```

   c. 要默认在所有新镜像中启用日志，请使用 **ceph config set** 命令设置配置参数：

   示例

   ```
   [ceph: root@site-a /]# ceph config set global rbd_default_features 125
   [ceph: root@site-a /]# ceph config show mon.host01 rbd_default_features
   ```

4. 在两个存储集群中选择 mirroring 模式（池模式或镜像模式）。

   a. 启用 **池模式** ：

语法

```
rbd mirror pool enable POOL_NAME MODE
```

示例

```
[ceph: root@site-a /]# rbd mirror pool enable data pool
[ceph: root@site-b /]# rbd mirror pool enable data pool
```

这个示例启用对名为 **data** 的完整池进行镜像。

b. 启用 **镜像模式** ：

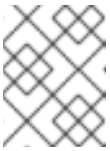
语法

```
rbd mirror pool enable POOL_NAME MODE
```

示例

```
[ceph: root@site-a /]# rbd mirror pool enable data image
[ceph: root@site-b /]# rbd mirror pool enable data image
```

这个示例在名为 **data** 的池上启用镜像模式镜像。



注意

要在池中为特定镜像启用镜像功能，请参阅 *Red Hat Ceph Storage 块设备指南中的 启用镜像镜像* 部分。

c. 验证两个站点都已成功启用了镜像：

语法

```
rbd mirror pool info POOL_NAME
```

示例

```
[ceph: root@site-a /]# rbd mirror pool info data
Mode: pool
Site Name: c13d8065-b33d-4cb5-b35f-127a02768e7f

Peer Sites: none

[ceph: root@site-b /]# rbd mirror pool info data
Mode: pool
Site Name: a4c667e2-b635-47ad-b462-6faeeee78df7

Peer Sites: none
```

5. 在 Ceph 客户端节点上，引导存储集群对等点。

a. 创建 Ceph 用户帐户，并将存储集群对等注册到池：

语法

```
rbd mirror pool peer bootstrap create --site-name PRIMARY_LOCAL_SITE_NAME
POOL_NAME > PATH_TO_BOOTSTRAP_TOKEN
```

示例

```
[ceph: root@rbd-client-site-a /]# rbd mirror pool peer bootstrap create --site-name site-a
data > /root/bootstrap_token_site-a
```

注意

这个示例 bootstrap 命令创建 **client.rbd-mirror.site-a** 和 **client.rbd-mirror-peer** Ceph 用户。

b. 将 bootstrap 令牌文件复制到 **site-b** 存储集群。

c. 在 **site-b** 存储集群中导入 bootstrap 令牌：

语法

```
rbd mirror pool peer bootstrap import --site-name SECONDARY_LOCAL_SITE_NAME --
direction rx-only POOL_NAME PATH_TO_BOOTSTRAP_TOKEN
```

示例

```
[ceph: root@rbd-client-site-b /]# rbd mirror pool peer bootstrap import --site-name site-b -
-direction rx-only data /root/bootstrap_token_site-a
```

注意

对于单向 RBD 镜像功能，您必须使用 **--direction rx-only** 参数，因为在引导对等时双向镜像是默认设置。

6. 要验证镜像状态，请从主站点和次要站点的 Ceph monitor 节点运行以下命令：

语法

```
rbd mirror image status POOL_NAME/IMAGE_NAME
```

示例

```
[ceph: root@mon-site-a /]# rbd mirror image status data/image1
image1:
  global_id:   c13d8065-b33d-4cb5-b35f-127a02768e7f
  state:       up+stopped
  description: remote image is non-primary
  service:     host03.yuoosv on host03
  last_update: 2021-10-06 09:13:58
```

在这里，**up** 表示 **rbd-mirror** 守护进程正在运行，**stopped** 意味着此镜像不是从另一个存储集群复制的目标。这是因为镜像是这个存储集群的主要部分。

示例

```
[ceph: root@mon-site-b /]# rbd mirror image status data/image1
image1:
  global_id:   c13d8065-b33d-4cb5-b35f-127a02768e7f
```

其它资源

- 如需了解更多详细信息，请参阅 *Red Hat Ceph Storage 块设备指南中的* Ceph *块设备镜像* 部分。
- 有关 Ceph *用户* 的更多详细信息，请参见 *Red Hat Ceph Storage 管理指南中的用户管理一节*。

6.3. 使用命令行界面配置双向镜像

此流程配置主存储集群和辅助存储集群之间的池的双向复制。

注意

使用双向复制时，您只能在两个存储集群之间镜像。

注意

本节中的示例通过将主镜像作为 **site-a** 引用主存储集群和您将要复制镜像的辅助存储集群作为 **site-b** 来区分两个存储集群。这些示例中使用的池名称称为 **data**。

先决条件

- 至少两个健康状态并运行 Red Hat Ceph Storage 集群。
- 对每个存储集群的 Ceph 客户端节点的根级别访问权限。
- 具有管理员级别功能的 CephX 用户。

流程

1. 在两个站点上登录到 **cephadm** shell：

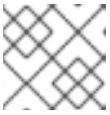
   示例

   ```
   [root@site-a ~]# cephadm shell
   [root@site-b ~]# cephadm shell
   ```

2. 在 **site-a** 主集群中运行以下命令：

   示例

   ```
   [ceph: root@site-a /]# ceph orch apply rbd-mirror --placement=host01
   ```



注意

nodename 是您要配置镜像的主机。

3. 在 **site-b** 上，调度在二级集群中部署 mirror 守护进程：

语法

```
ceph orch apply rbd-mirror --placement=NODENAME
```

示例

```
[ceph: root@site-b /]# ceph orch apply rbd-mirror --placement=host04
```

注意

nodename 是要在二级集群中配置镜像的主机。

4. 在 **site-a** 上的镜像上启用日志功能。

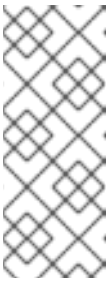
   a. 对于**新镜像**，使用 **--image-feature** 选项：

   语法

   ```
   rbd create IMAGE_NAME --size MEGABYTES --pool POOL_NAME --image-feature
   FEATURE FEATURE
   ```

   示例

   ```
   [ceph: root@site-a /]# rbd create image1 --size 1024 --pool data --image-feature
   exclusive-lock,journaling
   ```

   注意

   如果已经启用了 **exclusive-lock**，则使用 **journaling** 作为唯一参数，否则会返回以下错误：

   ```
   one or more requested features are already enabled
   (22) Invalid argument
   ```

   b. 对于**现有镜像**，请使用 **rbd feature enable** 命令：

   语法

   ```
   rbd feature enable POOL_NAME/IMAGE_NAME FEATURE, FEATURE
   ```

   示例

   ```
   [ceph: root@site-a /]# rbd feature enable data/image1 exclusive-lock, journaling
   ```

c. 要默认在所有新镜像中启用日志，请使用 **ceph config set** 命令设置配置参数：

示例

```
[ceph: root@site-a /]# ceph config set global rbd_default_features 125
[ceph: root@site-a /]# ceph config show mon.host01 rbd_default_features
```

5. 在两个存储集群中选择 mirroring 模式（池模式或镜像模式）。

a. 启用 **池模式** ：

语法

```
rbd mirror pool enable POOL_NAME MODE
```

示例

```
[ceph: root@site-a /]# rbd mirror pool enable data pool
[ceph: root@site-b /]# rbd mirror pool enable data pool
```

这个示例启用对名为 **data** 的完整池进行镜像。

b. 启用 **镜像模式** ：

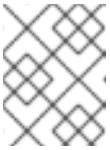
语法

```
rbd mirror pool enable POOL_NAME MODE
```

示例

```
[ceph: root@site-a /]# rbd mirror pool enable data image
[ceph: root@site-b /]# rbd mirror pool enable data image
```

这个示例在名为 **data** 的池上启用镜像模式镜像。



注意

要在池中为特定镜像启用镜像功能，请参阅 *Red Hat Ceph Storage 块设备指南中的 启用镜像镜像* 部分。

c. 验证两个站点都已成功启用了镜像：

语法

```
rbd mirror pool info POOL_NAME
```

示例

```
[ceph: root@site-a /]# rbd mirror pool info data
Mode: pool
Site Name: c13d8065-b33d-4cb5-b35f-127a02768e7f
```

```
Peer Sites: none

[ceph: root@site-b /]# rbd mirror pool info data
Mode: pool
Site Name: a4c667e2-b635-47ad-b462-6faeeee78df7

Peer Sites: none
```

6. 在 Ceph 客户端节点上，引导存储集群对等点。

   a. 创建 Ceph 用户帐户，并将存储集群对等注册到池：

   语法

   ```
   rbd mirror pool peer bootstrap create --site-name PRIMARY_LOCAL_SITE_NAME POOL_NAME > PATH_TO_BOOTSTRAP_TOKEN
   ```

   示例

   ```
   [ceph: root@rbd-client-site-a /]# rbd mirror pool peer bootstrap create --site-name site-a data > /root/bootstrap_token_site-a
   ```

   注意

   这个示例 bootstrap 命令创建 **client.rbd-mirror.site-a** 和 **client.rbd-mirror-peer** Ceph 用户。

   b. 将 bootstrap 令牌文件复制到 **site-b** 存储集群。

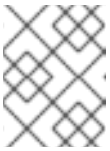
   c. 在 **site-b** 存储集群中导入 bootstrap 令牌：

   语法

   ```
   rbd mirror pool peer bootstrap import --site-name SECONDARY_LOCAL_SITE_NAME --direction rx-tx POOL_NAME PATH_TO_BOOTSTRAP_TOKEN
   ```

   示例

   ```
   [ceph: root@rbd-client-site-b /]# rbd mirror pool peer bootstrap import --site-name site-b --direction rx-tx data /root/bootstrap_token_site-a
   ```

   注意

   --direction 参数是可选的，因为在 bootstrapping peers 时双向镜像是默认设置。

7. 要验证镜像状态，请从主站点和次要站点的 Ceph monitor 节点运行以下命令：

   语法

   ```
   rbd mirror image status POOL_NAME/IMAGE_NAME
   ```

示例

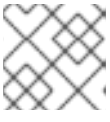
```
[ceph: root@mon-site-a /]# rbd mirror image status data/image1
image1:
  global_id:   a4c667e2-b635-47ad-b462-6faeeee78df7
  state:       up+stopped
  description: local image is primary
  service:     host03.glsdbv on host03.ceph.redhat.com
  last_update: 2021-09-16 10:55:58
  peer_sites:
    name: a
    state: up+stopped
    description: replaying,
{"bytes_per_second":0.0,"entries_behind_primary":0,"entries_per_second":0.0,"non_primary_p
osition":{"entry_tid":3,"object_number":3,"tag_tid":1},"primary_position":
{"entry_tid":3,"object_number":3,"tag_tid":1}}
    last_update: 2021-09-16 10:55:50
```

在这里，**up** 表示 **rbd-mirror** 守护进程正在运行，**stopped** 意味着此镜像不是从另一个存储集群复制的目标。这是因为镜像是这个存储集群的主要部分。

示例

```
[ceph: root@mon-site-b /]# rbd mirror image status data/image1
image1:
  global_id:   a4c667e2-b635-47ad-b462-6faeeee78df7
  state:       up+replaying
  description: replaying,
{"bytes_per_second":0.0,"entries_behind_primary":0,"entries_per_second":0.0,"non_primary_p
osition":{"entry_tid":3,"object_number":3,"tag_tid":1},"primary_position":
{"entry_tid":3,"object_number":3,"tag_tid":1}}
  service:     host05.dtisty on host05
  last_update: 2021-09-16 10:57:20
  peer_sites:
    name: b
    state: up+stopped
    description: local image is primary
    last_update: 2021-09-16 10:57:28
```

如果镜像处于 **up+replaying** 状态，则镜像可以正常工作。在这里，**up** 表示 **rbd-mirror** 守护进程正在运行，**replaying** 表示此镜像是从另一个存储集群复制的目标。



注意

根据站点之间的连接，镜像可能需要很长时间才能同步镜像。

其它资源

- 如需了解更多详细信息，请参阅 *Red Hat Ceph Storage 块设备指南中的* Ceph *块设备镜像* 部分。
- 有关 Ceph *用户* 的更多详细信息，请参见 *Red Hat Ceph Storage 管理指南中的用户管理一节*。

6.4. 镜像 CEPH 块设备的管理

作为存储管理员，您可以执行各种任务来帮助您管理 Ceph 块设备镜像环境。您可以执行以下任务：

- 查看有关存储群集对等点的信息.
- 添加或删除对等存储群集。
- 获取池或镜像的镜像状态。
- 启用对池或镜像的镜像。
- 禁用对池或镜像的镜像。
- 延迟块设备复制。
- 提升和降级镜像。

先决条件

- 至少运行两个健康的 Red Hat Ceph Storage 集群。
- 对 Ceph 客户端节点的根级别访问权限。
- 单向或双向 Ceph 块设备镜像关系。
- 具有管理员级别功能的 CephX 用户。

6.4.1. 查看有关同级的信息

查看有关存储集群对等点的信息。

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. 查看对等点的信息：

   语法

   ```
   rbd mirror pool info POOL_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror pool info data
   Mode: pool
   Site Name: a

   Peer Sites:

   UUID: 950ddadf-f995-47b7-9416-b9bb233f66e3
   Name: b
   ```

```
Mirror UUID: 4696cd9d-1466-4f98-a97a-3748b6b722b3
Direction: rx-tx
Client: client.rbd-mirror-peer
```

6.4.2. 启用对池的镜像

在两个对等集群中运行以下命令，在池上启用镜像功能。

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. 在池上启用镜像：

   语法

   ```
   rbd mirror pool enable POOL_NAME MODE
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror pool enable data pool
   ```

   这个示例启用对名为 **data** 的完整池进行镜像。

   示例

   ```
   [root@rbd-client ~]# rbd mirror pool enable data image
   ```

   这个示例在名为 **data** 的池上启用镜像模式镜像。

其它资源

- 详情请参阅 *Red Hat Ceph Storage 块设备指南* 中的*镜像 Ceph 块设备*一节。

6.4.3. 禁用对池的镜像

在禁用镜像前，删除对等集群。

注意

当您禁用对池的镜像时，您还会在池中在镜像模式中单独启用镜像的镜像禁用它。

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

流程

1. 在池上禁用镜像：

   语法

   ```
   rbd mirror pool disable POOL_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror pool disable data
   ```

   此示例禁用名为 **data** 的池的镜像。

6.4.4. 启用镜像镜像

在两个对等存储集群中，以镜像模式对整个池启用镜像功能。

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. 为池中的特定镜像启用镜像功能：

   语法

   ```
   rbd mirror image enable POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror image enable data/image2
   ```

   本例启用对 **data** 池中的 **image2** 镜像启用镜像。

其它资源

- 详情请参阅 *Red Hat Ceph Storage Block Device 指南* 中的 *对池启用镜像*部分。

6.4.5. 禁用镜像镜像

您可以在镜像中禁用 Ceph Block Device 镜像。

先决条件

- 一个运行的 Red Hat Ceph Storage 集群，带有基于快照的镜像配置。
- 节点的根级别访问权限。

流程

1. 禁用特定镜像的镜像：

   语法

   ```
   rbd mirror image disable POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror image disable data/image2
   ```

   本例禁用 **data** 池中 **image2** 镜像的镜像。

其它资源

- 如需了解有关 *将客户端添加到 **cephadm-ansible** 清单* 的更多详细信息，请参阅 *Red Hat Ceph Storage 安装指南中的配置* Ansible 清单位置部分。

6.4.6. 镜像提升和降级

您可以升级或降级池中的镜像。

注意

不要强制提升仍在同步的非主镜像，因为镜像在提升后无效。

先决条件

- 一个运行的 Red Hat Ceph Storage 集群，带有基于快照的镜像配置。
- 节点的根级别访问权限。

流程

1. 将镜像降级为非主要镜像：

   语法

   ```
   rbd mirror image demote POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror image demote data/image2
   ```

   本例降级 **data** 池中的 **image2** 镜像。

2. 将镜像提升为主要步骤：

   语法

   ```
   rbd mirror image promote POOL_NAME/IMAGE_NAME
   ```

示例

```
[root@rbd-client ~]# rbd mirror image promote data/image2
```

本例提升了 **data** 池中的 **image2**。

根据您使用的镜像类型，请参阅通过*单向镜像从灾难中恢复*，或者*通过双向镜像从灾难中恢复*。

语法

```
rbd mirror image promote --force POOL_NAME/IMAGE_NAME
```

示例

```
[root@rbd-client ~]# rbd mirror image promote --force data/image2
```

当降级无法传播到对等 Ceph 存储群集时，请使用强制提升。例如，由于集群失败或通信中断。

其它资源

- 有关详细信息，请参阅 *Red Hat Ceph Storage 块设备指南* 中的*故障切换*部分。

6.4.7. 镜像重新同步

您可以重新同步镜像。如果两个对等集群之间状态不一致，**rbd-mirror** 守护进程不会尝试镜像导致不一致的镜像。

先决条件

- 一个运行的 Red Hat Ceph Storage 集群，带有基于快照的镜像配置。
- 节点的根级别访问权限。

流程

1. 请求到主镜像的重新同步：

   语法

   ```
   rbd mirror image resync POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror image resync data/image2
   ```

   这个示例请求在 **data** 池中重新同步 **image2**。

其它资源

- 要因为灾难而需要从不一致的状态中恢复，请参阅*通过单向镜像从灾难中恢复*，或者*通过双向镜像从灾难中恢复*。

6.4.8. 添加存储集群对等集群

为 **rbd-mirror** 守护进程添加一个存储集群 peer，以发现其对等存储集群。例如，要将 **site-a** 存储集群添加为 **site-b** 存储集群的对等点，然后从 **site-b** 存储集群中的客户端节点按照以下步骤操作。

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. 将 peer 注册到池：

   语法

   ```
   rbd --cluster CLUSTER_NAME mirror pool peer add POOL_NAME
   PEER_CLIENT_NAME@PEER_CLUSTER_NAME -n CLIENT_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd --cluster site-b mirror pool peer add data client.site-a@site-a -n
   client.site-b
   ```

6.4.9. 删除存储集群 peer

通过指定对等 UUID 来删除存储群集 peer。

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. 指定池名称和同级通用唯一标识符 (UUID)。

   语法

   ```
   rbd mirror pool peer remove POOL_NAME PEER_UUID
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror pool peer remove data 7e90b4ce-e36d-4f07-8cbc-
   42050896825d
   ```

   提示

   若要查看对等 UUID，可使用 **rbd mirror pool info** 命令。

6.4.10. 获取池的镜像状态

您可获取存储集群中池的镜像状态。

先决条件

- 一个运行的 Red Hat Ceph Storage 集群，带有基于快照的镜像配置。
- 节点的根级别访问权限。

流程

1. 获取镜像池概述：

   语法

   ```
   rbd mirror pool status POOL_NAME
   ```

   示例

   ```
   [root@site-a ~]# rbd mirror pool status data
   health: OK
   daemon health: OK
   image health: OK
   images: 1 total
       1 replaying
   ```

   提示

   要输出池中每个镜像的状态详情，请使用 **--verbose** 选项。

6.4.11. 获取单个镜像的镜像状态

您可以通过运行 **mirror image status** 命令获取镜像的镜像状态。

先决条件

- 一个运行的 Red Hat Ceph Storage 集群，带有基于快照的镜像配置。
- 节点的根级别访问权限。

流程

1. 获取已镜像镜像的状态：

   语法

   ```
   rbd mirror image status POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@site-a ~]# rbd mirror image status data/image2
   image2:
   ```

```
global_id:   1e3422a2-433e-4316-9e43-1827f8dbe0ef
state:       up+unknown
description: remote image is non-primary
service:     pluto008.yuoosv on pluto008
last_update: 2021-10-06 09:37:58
```

本例获取 **data** 池中 **image2** 镜像的状态。

6.4.12. 延迟块设备复制

无论您使用的是单向复制还是双向复制，您都可以延迟 RADOS 块设备 (RBD) 镜像镜像之间的复制。如果您要在复制到次要镜像之前恢复对主镜像的更改，则可能需要实施延迟复制。

注意

延迟块设备复制仅适用于基于日志的镜像。

为实施延迟复制，目标存储集群内的 **rbd-mirror** 守护进程应设置 **rbd_mirroring_replay_delay = *MINIMUM_DELAY_IN_SECONDS*** 配置选项。此设置可以在 **rbd-mirror** 守护进程使用的 **ceph.conf** 文件中全局应用，也可以在单个镜像基础上应用。

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. 要使用特定镜像的延迟复制，在主镜像上运行以下 **rbd** CLI 命令：

   语法

   ```
   rbd image-meta set POOL_NAME/IMAGE_NAME conf_rbd_mirroring_replay_delay MINIMUM_DELAY_IN_SECONDS
   ```

   示例

   ```
   [root@rbd-client ~]# rbd image-meta set vms/vm-1 conf_rbd_mirroring_replay_delay 600
   ```

   本例在 **vms** 池中设置镜像 **vm-1** 的最小复制延迟 10 分钟。

6.4.13. 将基于日志的镜像转换为基于快照的镜像

您可以通过禁用镜像并启用快照，将基于日志的镜像转换为基于快照的镜像。

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@rbd-client ~]# cephadm shell
   ```

2. 为池中的特定镜像禁用镜像：

   语法

   ```
   rbd mirror image disable POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [ceph: root@rbd-client /]# rbd mirror image disable mirror_pool/mirror_image
   Mirroring disabled
   ```

3. 为镜像启用基于快照的镜像：

   语法

   ```
   rbd mirror image enable POOL_NAME/IMAGE_NAME snapshot
   ```

   示例

   ```
   [ceph: root@rbd-client /]# rbd mirror image enable mirror_pool/mirror_image snapshot
   Mirroring enabled
   ```

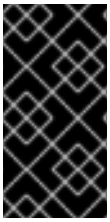
   本例为 **mirror_pool** 池中 **mirror_image** 镜像启用基于快照的镜像。

6.4.14. 创建镜像 mirror-snapshot

在使用基于快照的镜像功能时，创建镜像 mirror-snapshot，以镜像 RBD 镜像已更改的内容。

先决条件

- 至少运行两个健康的 Red Hat Ceph Storage 集群。
- 对 Red Hat Ceph Storage 集群的 Ceph 客户端节点的根级别访问权限。
- 具有管理员级别功能的 CephX 用户。
- 访问创建快照镜像的 Red Hat Ceph Storage 集群。



重要

默认情况下，保留最多 5 个镜像 mirror-snapshot。如果达到限制，则最新镜像 mirror-snapshot 会自动被删除。如果需要，可以通过 **rbd_mirroring_max_mirroring_snapshots** 配置覆盖限制。镜像 mirror-snapshot 会在镜像被删除或禁用镜像时自动删除。

流程

- 创建 image-mirror 快照：

  语法

  ```
  rbd --cluster CLUSTER_NAME mirror image snapshot POOL_NAME/IMAGE_NAME
  ```

  示例

  ```
  [root@site-a ~]# rbd mirror image snapshot data/image1
  ```

其它资源

- 详情请参阅 *Red Hat Ceph Storage 块设备指南* 中的*镜像 Ceph 块设备*一节。

6.4.15. 调度 mirror-snapshot

在定义 mirror-snapshot 调度时，可以自动创建 mirror-snapshots。mirror-snapshot 可以按池或镜像级别进行全局调度。可以在任何级别上定义多个 mirror-snapshot 调度，但只有与单个镜像的镜像匹配的最具体的快照调度才会运行。

6.4.15.1. 创建 mirror-snapshot 调度

您可以使用 **snapshot schedule** 命令创建 mirror-snapshot 调度。

先决条件

- 至少运行两个健康的 Red Hat Ceph Storage 集群。
- 对 Red Hat Ceph Storage 集群的 Ceph 客户端节点的根级别访问权限。
- 具有管理员级别功能的 CephX 用户。
- 访问需要调度镜像的镜像的 Red Hat Ceph Storage 集群。

流程

1. 创建 mirror-snapshot 调度：

   语法

   ```
   rbd --cluster CLUSTER_NAME mirror snapshot schedule add --pool POOL_NAME --image IMAGE_NAME INTERVAL [START_TIME]
   ```

   只有在集群名称与默认名称 **ceph** 不同时，才应使用 *CLUSTER_NAME*。间隔可以分别使用 d、h 或 m 后缀以天、小时或分钟为单位指定。可选的 START_TIME 可以使用 ISO 8601 时间格式指定。

   示例

   ```
   [root@site-a ~]# rbd mirror snapshot schedule add --pool data --image image1 6h
   ```

示例

```
[root@site-a ~]# rbd mirror snapshot schedule add --pool data --image image1 24h 14:00:00-05:00
```

其它资源

- 详情请参阅 *Red Hat Ceph Storage 块设备指南* 中的*镜像 Ceph 块设备*一节。

6.4.15.2. 列出特定级别的所有快照计划

您可以在特定级别列出所有快照计划。

先决条件

- 至少运行两个健康的 Red Hat Ceph Storage 集群。
- 对 Red Hat Ceph Storage 集群的 Ceph 客户端节点的根级别访问权限。
- 具有管理员级别功能的 CephX 用户。
- 访问需要调度镜像的镜像的 Red Hat Ceph Storage 集群。

流程

1. 使用可选池或镜像名称列出特定全局、池或镜像级别的所有快照调度：

   语法

   ```
   rbd --cluster site-a mirror snapshot schedule ls --pool POOL_NAME --recursive
   ```

   此外，还可指定 `**--recursive** 选项来列出指定级别的所有调度，如下所示：

   示例

   ```
   [root@rbd-client ~]# rbd mirror snapshot schedule ls --pool data --recursive
   POOL  NAMESPACE IMAGE  SCHEDULE
   data  -  -  every 1d starting at 14:00:00-05:00
   data  -  image1  every 6h
   ```

其它资源

- 详情请参阅 *Red Hat Ceph Storage 块设备指南* 中的*镜像 Ceph 块设备*一节。

6.4.15.3. 删除 mirror-snapshot 调度

您可以使用 **snapshot schedule remove** 命令删除 mirror-snapshot 调度。

先决条件

- 至少运行两个健康的 Red Hat Ceph Storage 集群。
- 对 Red Hat Ceph Storage 集群的 Ceph 客户端节点的根级别访问权限。

- 具有管理员级别功能的 CephX 用户。
- 访问需要调度镜像的镜像的 Red Hat Ceph Storage 集群。

流程

1. 删除 mirror-snapshot 调度：

   语法

   ```
   rbd --cluster CLUSTER_NAME mirror snapshot schedule remove --pool POOL_NAME --image IMAGE_NAME INTERVAL START_TIME
   ```

   间隔可以分别使用 d、h 和 m 后缀来以天数、小时或分钟为单位指定。可选的 START_TIME 可以使用 ISO 8601 时间格式指定。

   示例

   ```
   [root@site-a ~]# rbd mirror snapshot schedule remove --pool data --image image1 6h
   ```

   示例

   ```
   [root@site-a ~]# rbd mirror snapshot schedule remove --pool data --image image1 24h 14:00:00-05:00
   ```

其它资源

- 详情请参阅 *Red Hat Ceph Storage 块设备指南* 中的*镜像 Ceph 块设备*一节。

6.4.15.4. 查看要创建的下一个快照的状态

查看要为基于快照的镜像 RBD 镜像创建下一快照的状态。

先决条件

- 至少运行两个健康的 Red Hat Ceph Storage 集群。
- 对 Red Hat Ceph Storage 集群的 Ceph 客户端节点的根级别访问权限。
- 具有管理员级别功能的 CephX 用户。
- 访问需要调度镜像的镜像的 Red Hat Ceph Storage 集群。

流程

1. 查看要创建的下一个快照的状态：

   语法

   ```
   rbd --cluster site-a mirror snapshot schedule status [--pool POOL_NAME] [--image IMAGE_NAME]
   ```

   示例

```
[root@rbd-client ~]# rbd mirror snapshot schedule status
SCHEDULE    TIME       IMAGE
2021-09-21 18:00:00 data/image1
```

其它资源

- 详情请参阅 *Red Hat Ceph Storage 块设备指南* 中的*镜像 Ceph 块设备*一节。

6.5. 从灾难中恢复

作为存储管理员，您可以通过了解如何从配置了镜像功能的另一个存储集群恢复数据，为最终的硬件故障做好准备。

在示例中，主存储集群称为 **site-a**，辅助存储集群称为 **site-b**。此外，存储集群还拥有一个含有两个镜像，**image1** 和 **image2** 的 **data** 池。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 配置了单向或双向镜像。

6.5.1. 灾难恢复

在两个或多个 Red Hat Ceph Storage 集群间异步复制块数据可减少停机时间，并防止发生重大数据中心故障时出现数据丢失。这些故障具有广泛的影响，也称为 **大刀片**，并且可能源自对电网和危险性的影响。

客户数据需要在这些情况下受到保护。卷必须遵循一致性和效率，并在恢复点目标 (RPO) 和恢复时间目标 (RTO) 目标内进行复制。此解决方案称为广域网灾难恢复 (WAN-DR)。

在这种情况下，很难恢复主系统和数据中心。恢复的最快速方法是将应用程序故障转移到备用的 Red Hat Ceph Storage 集群（灾难恢复站点），并使集群能够运行最新可用数据副本。用于从这些故障场景中恢复的解决方案由应用程序指导：

- **恢复点目标 (RPO)**：在最坏的情况下，应用程序允许的数据丢失的数量。
- **恢复时间目标(RTO)** ：使用最新可用数据副本使应用程序重新上线所需的时间。

其它资源

- 详情请参阅 *Red Hat Ceph Storage 块设备指南* 中的*镜像 Ceph 块设备*一节。
- 请参阅 *Red Hat Ceph Storage 数据安全和硬化指南* 中的*加密传输*部分，以了解更多有关通过加密状态通过线路传输数据的信息。

6.5.2. 使用单向镜像从灾难中恢复

要使用单向镜像功能，可以从灾难中恢复，请使用以下步骤：它们显示在主集群终止后如何切换到次集群，以及如何恢复故障。关闭可以按照一定顺序进行，也可以不按照一定顺序进行。

重要

单向镜像支持多个次要站点。如果使用额外的次集群，请选择一个二级集群来切换到它。在故障恢复期间从同一集群进行同步。

6.5.3. 使用双向镜像从灾难中恢复

要使用双向镜像功能，可以从灾难中恢复，请使用以下步骤：它们演示了如何在主集群终止后切换到次要集群中的镜像数据，以及如何故障恢复。关闭可以按照一定顺序进行，也可以不按照一定顺序进行。

6.5.4. 有序关闭后故障转移

正常关闭后故障转移到次存储集群。

先决条件

- 至少两个正在运行的 Red Hat Ceph Storage 集群。
- 节点的根级别访问权限。
- 使用单向镜像配置的池镜像或镜像镜像。

流程

1. 停止使用主镜像的所有客户端。此步骤取决于哪些客户端使用该镜像。例如，从使用该镜像的任何 OpenStack 实例分离卷。
2. 在 **site-a** 集群中的监控节点中运行以下命令来降级位于 **site-a** 集群中的主镜像：

   语法

   ```
   rbd mirror image demote POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror image demote data/image1
   [root@rbd-client ~]# rbd mirror image demote data/image2
   ```

3. 在 **site-b** 集群中的监控节点中运行以下命令来提升位于 **site-b** 集群中的非主镜像：

   语法

   ```
   rbd mirror image promote POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror image promote data/image1
   [root@rbd-client ~]# rbd mirror image promote data/image2
   ```

4. 经过一段时间后，检查 **site-b** 集群中监控节点中的镜像状态。它们应当显示 **up+stopped** 状态，并列为主要状态：

   ```
   [root@rbd-client ~]# rbd mirror image status data/image1
   image1:
     global_id:   08027096-d267-47f8-b52e-59de1353a034
     state:       up+stopped
     description: local image is primary
     last_update: 2019-04-17 16:04:37
   ```

```
[root@rbd-client ~]# rbd mirror image status data/image2
image2:
  global_id:   596f41bc-874b-4cd4-aefe-4929578cc834
  state:       up+stopped
  description: local image is primary
  last_update: 2019-04-17 16:04:37
```

5. 恢复对镜像的访问。此步骤取决于哪些客户端使用该镜像。

其它资源

- 请参阅 *Red Hat OpenStack Platform 指南* 中的*块存储和卷*章节。

6.5.5. 非有序关闭后故障转移

非有序关闭后故障转移到次要存储集群。

先决条件

- 至少两个正在运行的 Red Hat Ceph Storage 集群。
- 节点的根级别访问权限。
- 使用单向镜像配置的池镜像或镜像镜像。

流程

1. 验证主存储集群是否已关闭。
2. 停止使用主镜像的所有客户端。此步骤取决于哪些客户端使用该镜像。例如，从使用该镜像的任何 OpenStack 实例分离卷。
3. 从 **site-b** 存储集群中的 Ceph 监控节点提升非主镜像。使用 **--force** 选项，因为降级无法传播到 **site-a** 存储集群：

   语法

   ```
   rbd mirror image promote --force POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror image promote --force data/image1
   [root@rbd-client ~]# rbd mirror image promote --force data/image2
   ```

4. 检查 **site-b** 存储集群中 Ceph 监控节点的镜像状态。它们应当显示 **up+stopping_replay** 状态。描述应该 **强制提升**，这意味着它处于间歇性状态。等待状态变为 **up+stopped** 以验证站点已成功提升。

   示例

   ```
   [root@rbd-client ~]# rbd mirror image status data/image1
   image1:
     global_id:   08027096-d267-47f8-b52e-59de1353a034
     state:       up+stopping_replay
   ```

```
    description: force promoted
    last_update: 2023-04-17 13:25:06

[root@rbd-client ~]# rbd mirror image status data/image1
image1:
  global_id:   08027096-d267-47f8-b52e-59de1353a034
  state:       up+stopped
  description: force promoted
  last_update: 2023-04-17 13:25:06
```

其它资源

- 请参阅 *Red Hat OpenStack Platform 指南*中的*块存储和卷*章节。

6.5.6. 准备故障恢复

如果两个存储集群最初只配置为单向镜像，为了避免故障，请配置主存储集群以进行镜像，以便按照相反方向复制镜像。

在故障恢复场景中，必须先删除无法访问的现有对等点，然后才能向现有集群添加新对等点。

先决条件

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

流程

1. 登录到 Cephadm shell：

   示例

   ```
   [root@rbd-client ~]# cephadm shell
   ```

2. 在 **site-a** 存储集群上，运行以下命令：

   示例

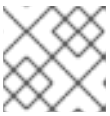
   ```
   [ceph: root@rbd-client /]# ceph orch apply rbd-mirror --placement=host01
   ```

3. 删除任何无法访问的对等点。

   重要

   此步骤必须在启动和运行的对等站点上运行。

   注意

   多个对等点仅支持单方向镜像。

   a. 获取池 UUID：

语法

```
rbd mirror pool info POOL_NAME
```

示例

```
[ceph: root@host01 /]# rbd mirror pool info pool_failback
```

b. 删除无法访问的对等点：

语法

```
rbd mirror pool peer remove POOL_NAME PEER_UUID
```

示例

```
[ceph: root@host01 /]# rbd mirror pool peer remove pool_failback f055bb88-6253-4041-923d-08c4ecbe799a
```

4. 创建名称与对等镜像池相同的块设备池。

a. 要创建 rbd 池，请执行以下操作：

语法

```
ceph osd pool create POOL_NAME PG_NUM
ceph osd pool application enable POOL_NAME rbd
rbd pool init -p POOL_NAME
```

示例

```
[root@rbd-client ~]# ceph osd pool create pool1
[root@rbd-client ~]# ceph osd pool application enable pool1 rbd
[root@rbd-client ~]# rbd pool init -p pool1
```

5. 在 Ceph 客户端节点上，引导存储集群对等点。

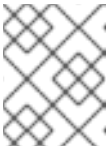
a. 创建 Ceph 用户帐户，并将存储集群对等注册到池：

语法

```
rbd mirror pool peer bootstrap create --site-name LOCAL_SITE_NAME POOL_NAME > PATH_TO_BOOTSTRAP_TOKEN
```

示例

```
[ceph: root@rbd-client-site-a /]# rbd mirror pool peer bootstrap create --site-name site-a data > /root/bootstrap_token_site-a
```



注意

这个示例 bootstrap 命令创建 **client.rbd-mirror.site-a** 和 **client.rbd-mirror-peer** Ceph 用户。

b. 将 bootstrap 令牌文件复制到 **site-b** 存储集群。

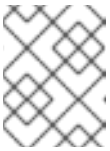
c. 在 **site-b** 存储集群中导入 bootstrap 令牌：

语法

```
rbd mirror pool peer bootstrap import --site-name LOCAL_SITE_NAME --direction rx-only POOL_NAME PATH_TO_BOOTSTRAP_TOKEN
```

示例

```
[ceph: root@rbd-client-site-b /]# rbd mirror pool peer bootstrap import --site-name site-b --direction rx-only data /root/bootstrap_token_site-a
```



注意

对于单向 RBD 镜像功能，您必须使用 **--direction rx-only** 参数，因为在引导对等时双向镜像是默认设置。

6. 在 **site-a** 存储集群中的监控节点中，验证 **site-b** 存储集群是否已成功添加为对等集群：

示例

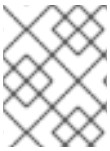
```
[ceph: root@rbd-client /]# rbd mirror pool info -p data
Mode: image
Peers:
  UUID                                 NAME   CLIENT
  d2ae0594-a43b-4c67-a167-a36c646e8643 site-b client.site-b
```

其它资源

- 如需更多信息，请参阅 *Red Hat Ceph Storage Administration Guide* *中的用户管理* 一章。

6.5.6.1. 返回主存储集群失败

当以前的主存储集群恢复时，失败回主存储集群。



注意

如果您在镜像级别上调度了快照，则需要重新添加计划，因为镜像重新同步操作会更改 RBD 镜像 ID，之前的调度会变得过时。

先决条件

- 至少两个正在运行的 Red Hat Ceph Storage 集群。
- 节点的根级别访问权限。

- 使用单向镜像配置的池镜像或镜像镜像。

流程

1. 再次检查 **site-b** 集群中监控节点的镜像状态。它们应该显示 **up-stopped** 状态，描述应该会指出 **local image is primary** ：

   示例

   ```
   [root@rbd-client ~]# rbd mirror image status data/image1
   image1:
     global_id:   08027096-d267-47f8-b52e-59de1353a034
     state:       up+stopped
     description: local image is primary
     last_update: 2019-04-22 17:37:48
   [root@rbd-client ~]# rbd mirror image status data/image2
   image2:
     global_id:   08027096-d267-47f8-b52e-59de1353a034
     state:       up+stopped
     description: local image is primary
     last_update: 2019-04-22 17:38:18
   ```

2. 从 **site-a** 存储集群的 Ceph 监控节点确定镜像是否仍然是主镜像：

   语法

   ```
   rbd mirror pool info POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd info data/image1
   [root@rbd-client ~]# rbd info data/image2
   ```

   在命令的输出中，查找 **mirroring primary: true** 或 **mirroring primary: false** 以确定状态。

3. 从 **site-a** 存储集群中的 Ceph monitor 节点运行以下命令来降级列为主要镜像：

   语法

   ```
   rbd mirror image demote POOL_NAME/IMAGE_NAME
   ```

   示例

   ```
   [root@rbd-client ~]# rbd mirror image demote data/image1
   ```

4. 如果未按顺序关闭，则仅重新同步镜像。在 **site-a** 存储集群中的监控节点上运行以下命令，以重新同步从 **site-b** 到 **site-a** 的镜像：

   语法

   ```
   rbd mirror image resync POOL_NAME/IMAGE_NAME
   ```

示例

```
[root@rbd-client ~]# rbd mirror image resync data/image1
Flagged image for resync from primary
[root@rbd-client ~]# rbd mirror image resync data/image2
Flagged image for resync from primary
```

5. 一段时间后，通过验证镜像是否处于 **up+replaying** 状态确保完成镜像重新同步。通过在 **site-a** 存储集群中的监控节点中运行以下命令来检查其状态：

语法

```
rbd mirror image status POOL_NAME/IMAGE_NAME
```

示例

```
[root@rbd-client ~]# rbd mirror image status data/image1
[root@rbd-client ~]# rbd mirror image status data/image2
```

6. 在 **site-b** 存储集群中的 Ceph monitor 节点上运行以下命令来降级 **site-b** 存储集群中的镜像：

语法

```
rbd mirror image demote POOL_NAME/IMAGE_NAME
```

示例

```
[root@rbd-client ~]# rbd mirror image demote data/image1
[root@rbd-client ~]# rbd mirror image demote data/image2
```

注意

如果有多个次要存储集群，则只需要从提升它的次要存储集群完成。

7. 在 **site-a** 存储集群中的 Ceph monitor 节点中运行以下命令来提升位于 **site-a** 存储集群中的以前主镜像：

语法

```
rbd mirror image promote POOL_NAME/IMAGE_NAME
```

示例

```
[root@rbd-client ~]# rbd mirror image promote data/image1
[root@rbd-client ~]# rbd mirror image promote data/image2
```

8. 检查 **site-a** 存储集群中 Ceph 监控节点的镜像状态。它们应当显示 **up+stopped** 状态，描述应该为 **local image is primary**:

语法

```
rbd mirror image status POOL_NAME/IMAGE_NAME
```

示例

```
[root@rbd-client ~]# rbd mirror image status data/image1
image1:
  global_id:   08027096-d267-47f8-b52e-59de1353a034
  state:       up+stopped
  description: local image is primary
  last_update: 2019-04-22 11:14:51
[root@rbd-client ~]# rbd mirror image status data/image2
image2:
  global_id:   596f41bc-874b-4cd4-aefe-4929578cc834
  state:       up+stopped
  description: local image is primary
  last_update: 2019-04-22 11:14:51
```

6.5.7. 删除双向镜像

恢复失败后，您可以移除双向镜像功能，并禁用 Ceph 块设备镜像服务。

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. 将 **site-b** 存储集群作为对等集群从 **site-a** 存储集群中删除：

   示例

   ```
   [root@rbd-client ~]# rbd mirror pool peer remove data client.remote@remote --cluster local
   [root@rbd-client ~]# rbd --cluster site-a mirror pool peer remove data client.site-b@site-b -n
   client.site-a
   ```

2. 在 **site-a** 客户端中停止并禁用 **rbd-mirror** 守护进程：

   语法

   ```
   systemctl stop ceph-rbd-mirror@CLIENT_ID
   systemctl disable ceph-rbd-mirror@CLIENT_ID
   systemctl disable ceph-rbd-mirror.target
   ```

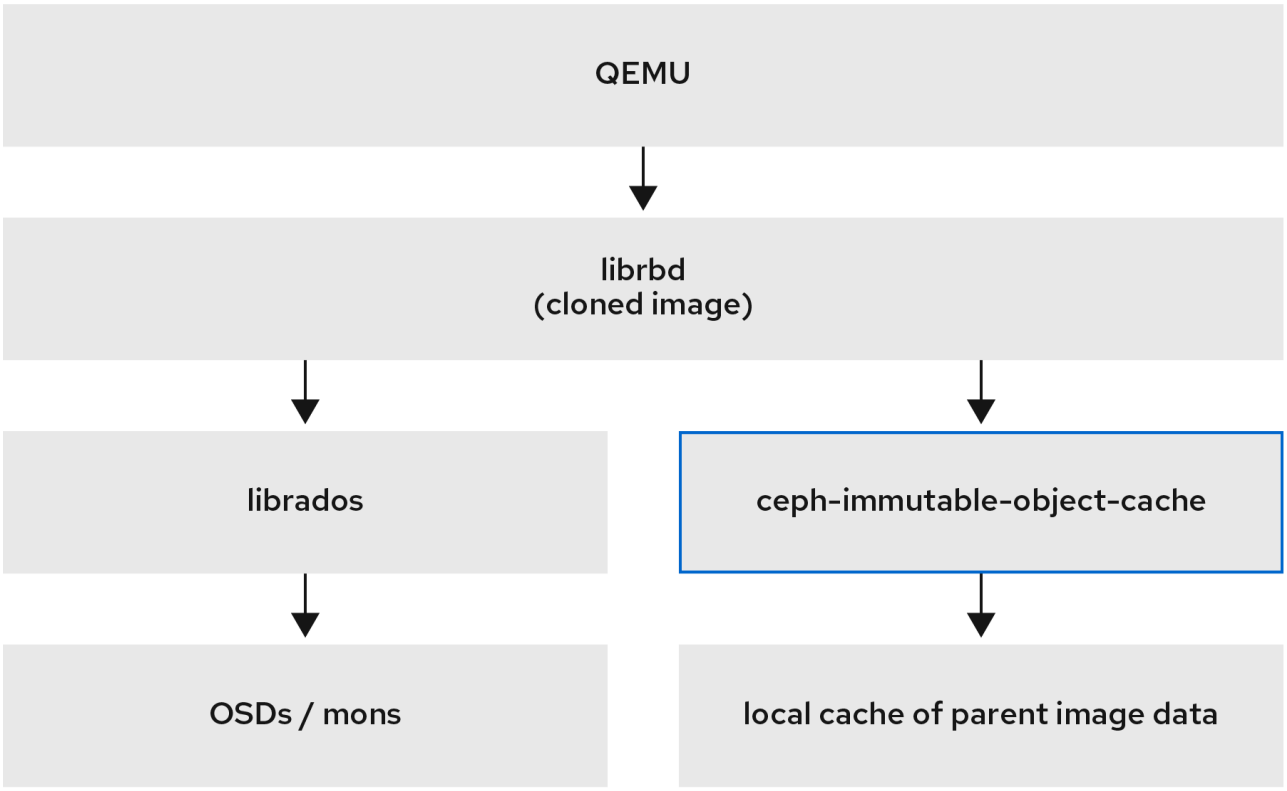
   示例

   ```
   [root@rbd-client ~]# systemctl stop ceph-rbd-mirror@site-a
   [root@rbd-client ~]# systemctl disable ceph-rbd-mirror@site-a
   [root@rbd-client ~]# systemctl disable ceph-rbd-mirror.target
   ```

第 7 章 管理 CEPH-IMMUTABLE-OBJECT-CACHE 守护进程

作为存储管理员，使用 **ceph-immutable-object-cache** 守护进程来缓存本地磁盘上的父镜像内容。这个缓存位于本地缓存目录中。以后在该数据上读取使用本地缓存。

图 7.1. Ceph 不可变缓存守护进程



7.1. CEPH-IMMUTABLE-OBJECT-CACHE 守护进程的解释

克隆块设备镜像通常只修改一小部分父镜像。例如，在虚拟桌面接口(VDI)中，虚拟机从同一基础镜像克隆，最初仅由主机名和 IP 地址不同。在启动过程中，如果您使用父镜像的本地缓存，这个速度会在缓存主机上读取。这个更改减少了客户端到集群网络流量。

使用 ceph-immutable-object-cache 守护进程的原因

ceph-immutable-object-cache 守护进程是 Red Hat Ceph Storage 的一部分。它是可扩展、开源和分布式存储系统。它使用 RADOS 协议连接到本地集群，根据默认的搜索路径查找 **ceph.conf** 文件，monitor 地址以及它们的验证信息，如 **/etc/ceph/*CLUSTER*.conf**, **/etc/ceph/*CLUSTER*.keyring**, 和 **/etc/ceph/*CLUSTER.NAME*.keyring**, 其中 *CLUSTER* 是集群的一个用户友好的名称，*NAME* 是用于连接的 RADOS 用户，例如 **client.ceph-immutable-object-cache**。

守护进程的主要组件

ceph-immutable-object-cache 守护进程有以下部分：

- 基于域套接字的进程间通信(IPC)：守护进程在启动时侦听本地域套接字，并等待来自 **librbd** 客户端的连接。
- 最早使用的(LRU)基于升级或降级策略：守护进程在每个缓存文件中维护 cache-hits 的内存统计信息。如果容量到达配置的阈值，它会降级冷缓存。

- 基于文件的缓存存储：守护进程维护基于文件的简单缓存存储。在提升 RADOS 对象时，从 RADOS 集群获取并存储在本地缓存目录中。

打开每个克隆的 RBD 镜像时，**librbd** 会尝试通过其 Unix 域套接字连接到 cache 守护进程。成功连接后，**librbd** 会在后续读取时与 守护进程协调。如果有没有缓存的读取，守护进程会将 RADOS 对象提升到本地缓存目录，因此对象上的下一个读取将从缓存中服务。守护进程还维护简单的 LRU 统计数据，以便在需要时让它驱除冷缓存文件。

注意

为提高性能，使用 SSD 作为底层存储。

7.2. 配置 CEPH-IMMUTABLE-OBJECT-CACHE 守护进程

ceph-immutable-object-cache 是 Ceph 集群内的 RADOS 对象对象缓存的守护进程。

重要

要使用 **ceph-immutable-object-cache** 守护进程，您必须能够连接 RADOS 集群。

守护进程将对象提升到本地目录。这些缓存对象服务将来会读取。您可以通过安装 **ceph-immutable-object-cache** 软件包来配置守护进程。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 至少一个 SSD 用于缓存。

流程

1. 启用 RBD 共享只读父镜像缓存。在 **/etc/ceph/ceph.conf** 文件的 **[client]** 部分添加以下参数：

   示例

   ```
   [root@ceph-host01 ~]# vi /etc/ceph/ceph.conf

   [client]
   rbd parent cache enabled = true
   rbd plugins = parent_cache
   ```

   重启集群。

2. 安装 **ceph-immutable-object-cache** 软件包：

   示例

   ```
   [root@ceph-host1 ~]# dnf install ceph-immutable-object-cache
   ```

3. 创建唯一的 Ceph 用户 ID，密钥环：

   语法

```
ceph auth get-or-create client.ceph-immutable-object-cache.USER_NAME mon 'profile rbd'
osd 'profile rbd-read-only'
```

示例

```
[root@ceph-host1 ~]# ceph auth get-or-create client.ceph-immutable-object-cache.user mon
'profile rbd' osd 'profile rbd-read-only'

[client.ceph-immutable-object-cache.user]
 key = AQCVPH1gFgHRAhAAp8ExRIsoxQK4QSYSRoVJLw==
```

复制此密钥环。

4. 在 **/etc/ceph** 目录中，创建一个文件并粘贴密钥环：

示例

```
[root@ceph-host1 ]# vi /etc/ceph/ceph.client.ceph-immutable-object-cache.user.keyring

[client.ceph-immutable-object-cache.user]
 key = AQCVPH1gFgHRAhAAp8ExRIsoxQK4QSYSRoVJLw
```

5. 启用守护进程：

语法

```
systemctl enable ceph-immutable-object-cache@ceph-immutable-object-
cache.USER_NAME
```

将 *USER_NAME* 指定为守护进程实例。

示例

```
[root@ceph-host1 ~]# systemctl enable ceph-immutable-object-cache@ceph-immutable-
object-cache.user

Created symlink /etc/systemd/system/ceph-immutable-object-cache.target.wants/ceph-
immutable-object-cache@ceph-immutable-object-cache.user.service →
/usr/lib/systemd/system/ceph-immutable-object-cache@.service.
```

6. 启动 **ceph-immutable-object-cache** 守护进程：

语法

```
systemctl start ceph-immutable-object-cache@ceph-immutable-object-cache.USER_NAME
```

示例

```
[root@ceph-host1 ~]# systemctl start ceph-immutable-object-cache@ceph-immutable-object-
cache.user
```

验证

- 检查配置的状态：

  语法

  ```
  systemctl status ceph-immutable-object-cache@ceph-immutable-object-cache.USER_NAME
  ```

  示例

  ```
  [root@ceph-host1 ~]# systemctl status ceph-immutable-object-cache@ceph-immutable-object-cache.user

  ● ceph-immutable-object-cache@ceph-immutable-object-cache.user>
    Loaded: loaded (/usr/lib/systemd/system/ceph-immutable-objec>
    Active: active (running) since Mon 2021-04-19 13:49:06 IST; >
  Main PID: 85020 (ceph-immutable-)
     Tasks: 15 (limit: 49451)
    Memory: 8.3M
    CGroup: /system.slice/system-ceph\x2dimmutable\x2dobject\x2d>
            └─85020 /usr/bin/ceph-immutable-object-cache -f --cl>
  ```

7.3. CEPH-IMMUTABLE-OBJECT-CACHE 守护进程的通用设置

列出了 **ceph-immutable-object-cache** 守护进程的一些重要通用设置。

immutable_object_cache_sock

描述

用于 librbd 客户端和 ceph-immutable-object-cache 守护进程间的通信域套接字的路径。

类型

字符串

默认

/var/run/ceph/immutable_object_cache_sock

immutable_object_cache_path

描述

不可变对象缓存数据目录。

类型

字符串

默认

/tmp/ceph_immutable_object_cache

immutable_object_cache_max_size

描述

不可变缓存的最大大小。

类型

大小

默认

1G

immutable_object_cache_watermark

描述

缓存的 high-water 标记。该值介于零和一之间。如果缓存大小达到这个阈值，守护进程将开始基于 LRU 统计信息删除冷缓存。

类型

浮点值

默认

0.9

7.4. CEPH-IMMUTABLE-OBJECT-CACHE 守护进程的 QOS 设置

ceph-immutable-object-cache 守护进程支持节流支持各个设置。

immutable_object_cache_qos_schedule_tick_min

描述

不可变对象缓存的最小调度选择。

类型

Milliseconds

默认

50

immutable_object_cache_qos_iops_limit

描述

用户定义的不可变对象缓存 IO 操作限制每秒。

类型

整数

默认

0

immutable_object_cache_qos_iops_burst

描述

用户定义的不可变对象缓存 IO 操作的突发限制。

类型

整数

默认

0

immutable_object_cache_qos_iops_burst_seconds

描述

用户定义的 burst 持续时间，单位为不可变对象缓存 IO 操作。

类型

秒

默认

1

immutable_object_cache_qos_bps_limit

描述

用户定义的不可变对象缓存每秒 IO 字节限制。

类型

整数

默认

0

immutable_object_cache_qos_bps_burst

描述

用户定义的不可变对象缓存 IO 字节的突发限制。

类型

整数

默认

0

immutable_object_cache_qos_bps_burst_seconds

描述

所需的读操作突发限制。

类型

秒

默认

1

第 8 章 RBD 内核模块

作为存储管理员，您可以通过 **rbd** 内核模块访问 Ceph 块设备。您可以映射和取消映射块设备，并显示这些映射。此外，您可以通过 **rbd** 内核模块获取镜像列表。

重要

用户可以使用 Red Hat Enterprise Linux (RHEL) 以外的 Linux 发行版本中的内核客户端，但并不被支持。如果在存储集群中使用这些内核客户端时发现问题，红帽会解决这些问题，但是如果发现根本原因在内核客户端一侧，则软件供应商必须解决这个问题。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

8.1. 创建 CEPH 块设备并从 LINUX 内核模块客户端使用它

作为存储管理员，您可以在 Red Hat Ceph Storage 控制面板中为 Linux 内核模块客户端创建 Ceph 块设备。作为系统管理员，您可以使用命令行将该块设备映射到 Linux 客户端，并进行分区、格式化和挂载。之后，您可以为其读取和写入文件。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 一个 Red Hat Enterprise Linux 客户端。

8.1.1. 使用仪表板为 Linux 内核模块客户端创建 Ceph 块设备

您可以通过仅启用它支持的功能，使用控制面板 Web 界面为 Linux 内核模块客户端创建 Ceph 块设备。

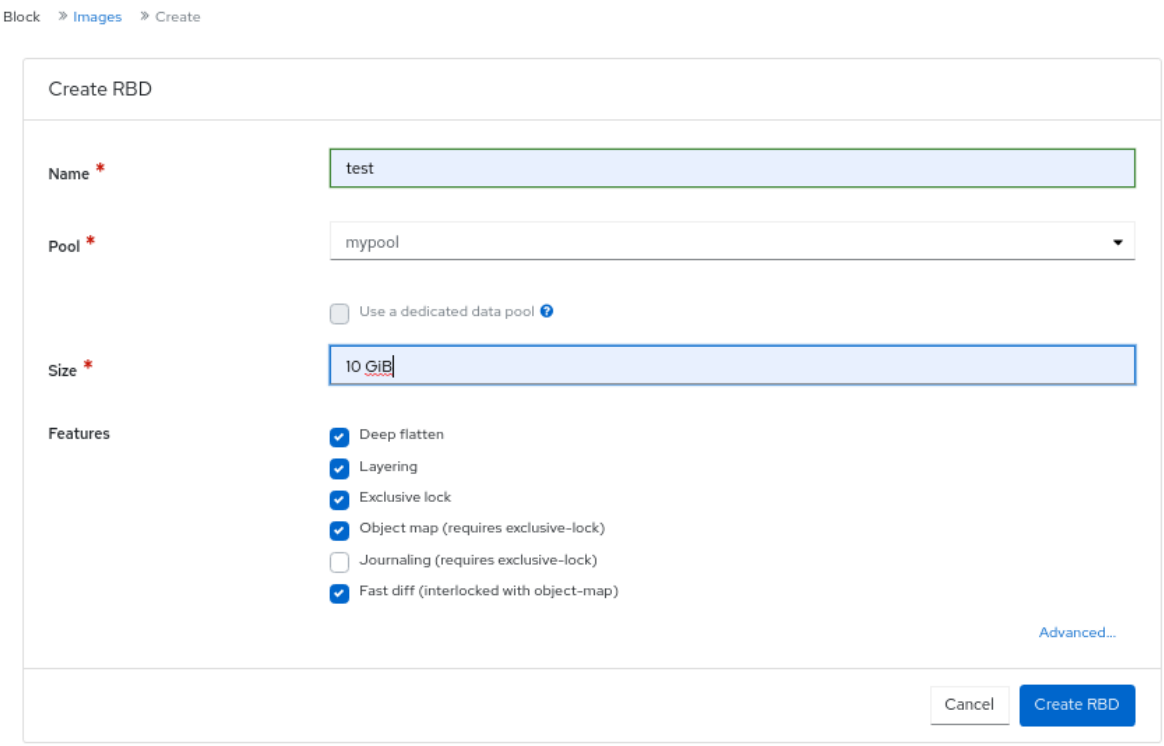
内核模块客户端支持如 Deep flatten、Layering、Exclusive 锁定、Object map 和 Fast diff 等功能。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 创建并启用复制 RBD 池。

流程

1. 在 *Block* 下拉菜单中选择 *Images*。
2. 点 *Create*。
3. 在 *Create RBD* 窗口中，输入镜像名称，选择 RBD 启用池，选择支持的功能：



4. 点 *Create RBD*。

验证

- 您将获得一个成功创建镜像的通知。

其它资源

- 有关更多信息，请参阅 *Red Hat Ceph Storage Block Device Guide* 中的*使用命令行在 Linux 上映射和挂载 Ceph 块设备*。
- 有关更多信息，请参阅 *Red Hat Ceph Storage 仪表板指南*。

8.1.2. 使用命令行映射并挂载 Ceph 块设备到 Linux 上

您可以使用 Linux **rbd** 内核模块从 Red Hat Enterprise Linux 客户端映射 Ceph 块设备。映射之后，您可以对其进行分区、格式化和挂载，以便您可以将文件写入到其中。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 创建使用控制面板的 Linux 内核模块客户端的 Ceph 块设备。
- 一个 Red Hat Enterprise Linux 客户端。

流程

1. 在 Red Hat Enterprise Linux 客户端节点上，启用 Red Hat Ceph Storage 7 Tools 存储库：

```
[root@rbd-client ~]# subscription-manager repos --enable=rhceph-7-tools-for-rhel-9-x86_64-rpms
```

2. 安装 **ceph-common** RPM 软件包：

```
[root@rbd-client ~]# dnf install ceph-common
```

3. 将 Ceph 配置文件从 monitor 节点复制到客户端节点：

语法

```
scp root@MONITOR_NODE:/etc/ceph/ceph.conf /etc/ceph/ceph.conf
```

示例

```
[root@rbd-client ~]# scp root@cluster1-node2:/etc/ceph/ceph.conf /etc/ceph/ceph.conf
root@192.168.0.32's password:
ceph.conf                                     100%  497   724.9KB/s   00:00
[root@client1 ~]#
```

4. 将密钥文件从 monitor 节点复制到客户端节点：

语法

```
scp root@MONITOR_NODE:/etc/ceph/ceph.client.admin.keyring
/etc/ceph/ceph.client.admin.keyring
```

示例

```
[root@rbd-client ~]# scp root@cluster1-node2:/etc/ceph/ceph.client.admin.keyring
/etc/ceph/ceph.client.admin.keyring
root@192.168.0.32's password:
ceph.client.admin.keyring                     100%  151   265.0KB/s   00:00
[root@client1 ~]#
```

5. 映射镜像：

语法

```
rbd map --pool POOL_NAME IMAGE_NAME --id admin
```

示例

```
[root@rbd-client ~]# rbd map --pool block-device-pool image1 --id admin
/dev/rbd0
[root@client1 ~]#
```

6. 在块设备中创建分区表：

语法

```
parted /dev/MAPPED_BLOCK_DEVICE mklabel msdos
```

示例

```
[root@rbd-client ~]# parted /dev/rbd0 mklabel msdos
Information: You may need to update /etc/fstab.
```

7. 为 XFS 文件系统创建分区：

语法

```
parted /dev/MAPPED_BLOCK_DEVICE mkpart primary xfs 0% 100%
```

示例

```
[root@rbd-client ~]# parted /dev/rbd0 mkpart primary xfs 0% 100%
Information: You may need to update /etc/fstab.
```

8. 格式化分区：

语法

```
mkfs.xfs /dev/MAPPED_BLOCK_DEVICE_WITH_PARTITION_NUMBER
```

示例

```
[root@rbd-client ~]# mkfs.xfs /dev/rbd0p1
meta-data=/dev/rbd0p1            isize=512    agcount=16, agsize=163824 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=2621184, imaxpct=25
         =                       sunit=16     swidth=16 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=16 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

9. 创建要挂载新文件系统的目录：

语法

```
mkdir PATH_TO_DIRECTORY
```

示例

```
[root@rbd-client ~]# mkdir /mnt/ceph
```

10. 挂载文件系统：

语法

```
mount /dev/MAPPED_BLOCK_DEVICE_WITH_PARTITION_NUMBER
PATH_TO_DIRECTORY
```

示例

```
[root@rbd-client ~]# mount /dev/rbd0p1 /mnt/ceph/
```

11. 验证文件系统是否已挂载并显示正确的大小：

语法

```
df -h PATH_TO_DIRECTORY
```

示例

```
[root@rbd-client ~]# df -h /mnt/ceph/
Filesystem      Size  Used Avail Use% Mounted on
/dev/rbd0p1      10G  105M  9.9G   2% /mnt/ceph
```

其它资源

- 如需更多信息，请参阅 *使用控制面板为 Linux 内核模块客户端创建 Ceph 块设备*。
- 如需更多信息，请参阅为 Red Hat Enterprise Linux 8 管理文件系统。
- 如需更多信息，请参阅 Red Hat Enterprise Linux 7 的存储管理指南。

8.2. 映射块设备

使用 **rbd** 将镜像名称映射到内核模块。您必须指定镜像名称、池名称和用户名。**RBD** 将加载 RBD 内核模块（如果尚未加载）。

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. 返回镜像列表：

示例

```
[root@rbd-client ~]# rbd list
```

2. 以下是映射镜像的两个选项：

- 将镜像名称映射到内核模块：

语法

```
rbd device map POOL_NAME/IMAGE_NAME --id USER_NAME
```

示例

```
[root@rbd-client ~]# rbd device map rbd/myimage --id admin
```

- 在使用 **cephx** 身份验证时，通过密钥环或包含 secret 的文件指定 secret：

  语法

  ```
  [root@rbd-client ~]# rbd device map POOL_NAME/IMAGE_NAME --id USER_NAME --keyring PATH_TO_KEYRING
  ```

  或者

  ```
  [root@rbd-client ~]# rbd device map POOL_NAME/IMAGE_NAME --id USER_NAME --keyfile PATH_TO_FILE
  ```

8.3. 显示映射的块设备

您可以使用 **rbd** 命令显示哪些块设备镜像映射到内核模块。

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. 显示映射的块设备：

   ```
   [root@rbd-client ~]# rbd device list
   ```

8.4. 取消映射块设备

您可以使用 **unmap** 选项并提供设备名称，通过 **rbd** 命令取消 map 块设备镜像。

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。
- 映射的镜像。

流程

1. 获取设备的规格。

   示例

   ```
   [root@rbd-client ~]# rbd device list
   ```

2. 取消映射块设备镜像：

语法

```
rbd device unmap /dev/rbd/POOL_NAME/IMAGE_NAME
```

示例

```
[root@rbd-client ~]# rbd device unmap /dev/rbd/pool1/image1
```

8.5. 隔离同一池中的隔离命名空间中的镜像

在没有更高级别的系统（如 OpenStack 或 OpenShift Container Storage）的情况下直接使用 Ceph 块设备时，无法限制用户对特定块设备镜像的访问。与 CephX 功能相结合，用户可以限制到特定的池命名空间，以限制对镜像的访问。

您可以使用 RADOS 命名空间（一种新的身份级别）来标识对象，以提供池中 rados 客户端之间的隔离。例如，客户端只能对特定于它们的命名空间具有完全权限。这使得将不同的 RADOS 客户端用于每个租户可行，这对于许多不同租户访问自己的块设备镜像的块设备特别有用。

您可以在同一池中的隔离命名空间内隔离块设备镜像。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 在所有客户端上将所有内核升级到 4x 和 librbd 和 librados。
- monitor 和客户端节点的 root 级别访问权限。

流程

1. 创建 **rbd** 池：

   语法

   ```
   ceph osd pool create POOL_NAME PG_NUM
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph osd pool create mypool 100
   pool 'mypool' created
   ```

2. 将 **rbd** 池与 RBD 应用关联：

   语法

   ```
   ceph osd pool application enable POOL_NAME rbd
   ```

   示例

   ```
   [ceph: root@host01 /]# ceph osd pool application enable mypool rbd
   enabled application 'rbd' on pool 'mypool'
   ```

3. 使用 RBD 应用程序初始化池：

语法

```
rbd pool init -p POOL_NAME
```

示例

```
[ceph: root@host01 /]# rbd pool init -p mypool
```

4. 创建两个命名空间：

语法

```
rbd namespace create --namespace NAMESPACE
```

示例

```
[ceph: root@host01 /]# rbd namespace create --namespace namespace1

[ceph: root@host01 /]# rbd namespace create --namespace namespace2

[ceph: root@host01 /]# rbd namespace ls --format=json
[{"name":"namespace2"},{"name":"namespace1"}]
```

5. 为两个用户提供命名空间的访问权限：

语法

```
ceph auth get-or-create client.USER_NAME mon 'profile rbd' osd 'profile rbd pool=rbd
namespace=NAMESPACE' -o /etc/ceph/client.USER_NAME.keyring
```

示例

```
[ceph: root@host01 /]# ceph auth get-or-create client.testuser mon 'profile rbd' osd 'profile
rbd pool=rbd namespace=namespace1' -o /etc/ceph/client.testuser.keyring

[ceph: root@host01 /]# ceph auth get-or-create client.newuser mon 'profile rbd' osd 'profile
rbd pool=rbd namespace=namespace2' -o /etc/ceph/client.newuser.keyring
```

6. 获取客户端的密钥：

语法

```
ceph auth get client.USER_NAME
```

示例

```
[ceph: root@host01 /]# ceph auth get client.testuser

[client.testuser]
 key = AQDMp61hBf5UKRAAgjQ2In0Z3uwAase7mrlKnQ==
```

```
 caps mon = "profile rbd"
 caps osd = "profile rbd pool=rbd namespace=namespace1"
exported keyring for client.testuser

[ceph: root@host01 /]# ceph auth get client.newuser

[client.newuser]
 key = AQDfp61hVfLFHRAA7D80ogmZl80ROY+AUG4A+Q==
 caps mon = "profile rbd"
 caps osd = "profile rbd pool=rbd namespace=namespace2"
exported keyring for client.newuser
```

7. 创建块设备镜像并使用池中的预定义命名空间：

语法

```
rbd create --namespace NAMESPACE IMAGE_NAME --size SIZE_IN_GB
```

示例

```
[ceph: root@host01 /]# rbd create --namespace namespace1 image01 --size 1G

[ceph: root@host01 /]# rbd create --namespace namespace2 image02 --size 1G
```

8. 可选：获取命名空间和关联的镜像的详情：

语法

```
rbd --namespace NAMESPACE ls --long
```

示例

```
[ceph: root@host01 /]#  rbd --namespace namespace1 ls --long
NAME    SIZE  PARENT FMT PROT LOCK
image01 1 GiB          2

[ceph: root@host01 /]# rbd --namespace namespace2 ls --long
NAME    SIZE  PARENT FMT PROT LOCK
image02 1 GiB          2
```

9. 将 Ceph 配置文件从 Ceph 监控节点复制到客户端节点：

```
scp /etc/ceph/ceph.conf root@CLIENT_NODE:/etc/ceph/
```

示例

```
[ceph: root@host01 /]# scp /etc/ceph/ceph.conf root@host02:/etc/ceph/

root@host02's password:
ceph.conf                                     100%  497   724.9KB/s   00:00
```

10. 将 Ceph 监控节点的 admin 密钥环复制到客户端节点：

语法

```
scp /etc/ceph/ceph.client.admin.keyring root@CLIENT_NODE:/etc/ceph
```

示例

```
[ceph: root@host01 /]# scp /etc/ceph/ceph.client.admin.keyring root@host02:/etc/ceph/

root@host02's password:
ceph.client.admin.keyring                                     100%  151   265.0KB/s   00:00
```

11. 将用户的密钥环从 Ceph 监控节点复制到客户端节点：

语法

```
scp /etc/ceph/ceph.client.USER_NAME.keyring root@CLIENT_NODE:/etc/ceph/
```

示例

```
[ceph: root@host01 /]# scp /etc/ceph/client.newuser.keyring root@host02:/etc/ceph/

[ceph: root@host01 /]# scp /etc/ceph/client.testuser.keyring root@host02:/etc/ceph/
```

12. 映射块设备镜像：

语法

```
rbd map --name NAMESPACE IMAGE_NAME -n client.USER_NAME --keyring
/etc/ceph/client.USER_NAME.keyring
```

示例

```
[ceph: root@host01 /]# rbd map --namespace namespace1 image01  -n  client.testuser --
keyring=/etc/ceph/client.testuser.keyring

/dev/rbd0

[ceph: root@host01 /]# rbd map --namespace namespace2 image02 -n client.newuser --
keyring=/etc/ceph/client.newuser.keyring

/dev/rbd1
```

这不允许访问同一池中的其他命名空间中的用户。

示例

```
[ceph: root@host01 /]# rbd map --namespace namespace2 image02  -n  client.testuser --
keyring=/etc/ceph/client.testuser.keyring

rbd: warning: image already mapped as /dev/rbd1
rbd: sysfs write failed
rbd: error asserting namespace: (1) Operation not permitted
```

```
In some cases useful info is found in syslog - try "dmesg | tail".
2021-12-06 02:49:08.106 7f8d4fde2500 -1 librbd::api::Namespace: exists: error asserting
namespace: (1) Operation not permitted
rbd: map failed: (1) Operation not permitted

[ceph: root@host01 /]# rbd map --namespace namespace1 image01 -n client.newuser --
keyring=/etc/ceph/client.newuser.keyring

rbd: warning: image already mapped as /dev/rbd0
rbd: sysfs write failed
rbd: error asserting namespace: (1) Operation not permitted
In some cases useful info is found in syslog - try "dmesg | tail".
2021-12-03 12:16:24.011 7fcad776a040 -1 librbd::api::Namespace: exists: error asserting
namespace: (1) Operation not permitted
rbd: map failed: (1) Operation not permitted
```

13. 验证该设备：

示例

```
[ceph: root@host01 /]# rbd showmapped

id pool namespace      image   snap device
0  rbd  namespace1  image01 -  /dev/rbd0
1  rbd  namespace2   image02 -  /dev/rbd1
```

第 9 章 使用 CEPH 块设备 PYTHON 模块

rbd python 模块提供对 Ceph 块设备镜像的类文件访问。要使用此内置工具，请导入 **rbd** 和 **rados** Python 模块。

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. 连接到 RADOS 并打开 IO 上下文：

```
cluster = rados.Rados(conffile='my_ceph.conf')
cluster.connect()
ioctx = cluster.open_ioctx('mypool')
```

2. 实例化一个 **:class:rbd.RBD** 对象，用于创建镜像：

```
rbd_inst = rbd.RBD()
size = 4 * 1024**3  # 4 GiB
rbd_inst.create(ioctx, 'myimage', size)
```

3. 要在镜像上执行 I/O，请实例化一个 **:class:rbd.Image** 对象：

```
image = rbd.Image(ioctx, 'myimage')
data = 'foo' * 200
image.write(data, 0)
```

这会将"foo"写入镜像的前 600 字节。请注意，数据不能是 **:type:unicode** - **librbd** 不知道如何处理比 **:c:type:char** 更宽的字符。

4. 关闭镜像、IO 上下文和与 RADOS 的连接：

```
image.close()
ioctx.close()
cluster.shutdown()
```

为了安全起见，每个调用都必须位于单独的 **:finally** 中 ：

```
import rados
import rbd

cluster = rados.Rados(conffile='my_ceph_conf')
try:
    ioctx = cluster.open_ioctx('my_pool')
    try:
        rbd_inst = rbd.RBD()
        size = 4 * 1024**3  # 4 GiB
        rbd_inst.create(ioctx, 'myimage', size)
        image = rbd.Image(ioctx, 'myimage')
        try:
```

```
            data = 'foo' * 200
            image.write(data, 0)
        finally:
            image.close()
    finally:
        ioctx.close()
finally:
    cluster.shutdown()
```

这可能会有问题，Rados、Ioctx 和 Image 类可以用作自动关闭或关闭的上下文管理器。使用它们作为上下文管理器时，上述示例如下：

```
with rados.Rados(conffile='my_ceph.conf') as cluster:
    with cluster.open_ioctx('mypool') as ioctx:
        rbd_inst = rbd.RBD()
        size = 4 * 1024**3  # 4 GiB
        rbd_inst.create(ioctx, 'myimage', size)
        with rbd.Image(ioctx, 'myimage') as image:
            data = 'foo' * 200
            image.write(data, 0)
```

附录 A. CEPH 块设备配置参考

作为存储管理员，您可以通过可用的各种选项，微调 Ceph 块设备的行为。您可以使用此参考来查看默认 Ceph 块设备选项和 Ceph 块设备缓存选项等内容。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

A.1. 块设备默认选项

可以通过创建镜像来覆盖默认设置。Ceph 将创建格式为 **2** 的镜像，没有条带化。

rbd_default_format

描述

如果没有指定其他格式，则使用默认格式 (**2**)格式 **1** 是新镜像的原始格式，兼容所有版本的 **librbd** 和内核模块，但不支持克隆等较新的功能。从版本 3.11 开始，**rbd** 和内核模块支持格式 **2**（条带除外）。格式 **2** 添加了对克隆的支持，且更易于扩展，以在未来允许更多功能。

类型

整数

默认

2

rbd_default_order

描述

如果没有指定其他顺序，默认的顺序。

类型

整数

默认

22

rbd_default_stripe_count

描述

如果未指定任何其他条带数，默认的条带数。更改默认值需要条带 v2 功能。

类型

64-bit Unsigned 整数

默认

0

rbd_default_stripe_unit

描述

如果未指定其他条带单元，默认条带单元。将单元从 **0**（即对象大小）改为其他值需要条带 v2 功能。

类型

64-bit Unsigned 整数

默认

0

rbd_default_features

描述

创建块设备镜像时启用的默认功能。此设置仅适用于格式 2 镜像。设置为：
1: Layering support分层允许您使用克隆。

2: Striping v2 support条带化可在多个对象之间分散数据。条带有助于并行处理连续读/写工作负载。

4: Exclusive locking support.启用后，它要求客户端在进行写入前获得对象锁定。

8: Object map support.块设备是精简配置的 - 这代表仅存储实际存在的数据。对象映射支持有助于跟踪实际存在的对象（将数据存储在驱动器上）。启用对象映射支持可加快克隆或导入和导出稀疏填充镜像的 I/O 操作。

16: Fast-diff support. Fast-diff 支持取决于对象映射支持和专用锁定支持。它向对象映射中添加了另一个属性，这可以更快地生成镜像快照和快照的实际数据使用量之间的差别。

32: Deep-flatten support.深度扁平使 **rbd flatten** 除了镜像本身外还作用于镜像的所有快照。如果没有它，镜像的快照仍会依赖于父级，因此在快照被删除之前，父级将无法删除。深度扁平化使得父级独立于克隆，即使它们有快照。

64: Journaling support.日志记录会按照镜像发生的顺序记录对镜像的所有修改。这样可确保远程镜像的 crash-consistent 镜像在本地可用

启用的功能是数字设置的总和。

类型

整数

默认

61 - 启用了分层、专用锁定、对象映射、fast-diff 和 deep-flatten

重要

当前的默认设置不兼容 RBD 内核驱动程序或较旧的 RBD 客户端。

rbd_default_map_options

描述

大多数选项主要用于调试和基准测试。详情请参阅 **map Options** 下的 **man rbd**。

类型

字符串

默认

""

A.2. 块设备常规选项

rbd_op_threads

描述

块设备操作线程数量。

类型

整数

默认

1

警告

不要更改 **rbd_op_threads** 的默认值，因为将其设置为大于 **1** 的数字可能会导致数据损坏。

rbd_op_thread_timeout

描述

块设备操作线程的超时时间（以秒为单位）。

类型

整数

默认

60

rbd_non_blocking_aio

描述

如果为 **true**，Ceph 将处理来自 worker 线程的块设备异步 I/O 操作，以防止阻止。

类型

布尔值

默认

true

rbd_concurrent_management_ops

描述

处理中并发管理操作的最大数量（例如，删除或调整镜像大小）。

类型

整数

默认

10

rbd_request_timed_out_seconds

描述

维护请求超时前的秒数。

类型

整数

默认

30

rbd_clone_copy_on_read

描述

当设置为 **true** 时，会启用读时复制克隆。

类型

布尔值

默认

false

rbd_enable_alloc_hint

描述

如果为 **true**，则启用分配提示，块设备将向 OSD 后端发出提示，以指示预期的大小对象。

类型

布尔值

默认

true

rbd_skip_partial_discard

描述

如果为 **true**，则块设备在尝试丢弃对象内的范围时将跳过零范围。

类型

布尔值

默认

false

rbd_tracing

描述

将这个选项设置为 **true** 以启用 Linux Trace Toolkit Next Generation User Space Tracer (LTTng-UST) 追踪点。详情请参阅 使用 RBD Replay 功能跟踪 RADOS 块设备(RBD) 工作负载。

类型

布尔值

默认

false

rbd_validate_pool

描述

将此选项设置为 **true**，以验证空池以实现 RBD 兼容性。

类型

布尔值

默认

true

rbd_validate_names

描述

将此选项设置为 **true** 以验证镜像规格。

类型

布尔值

默认

true

A.3. 块设备缓存选项

Ceph 块设备的用户空间实施（即 **librbd** ）无法利用 Linux 页面缓存，因此它包含了自己的内存中缓存，称为 **RBD caching（RBD 缓存）**。Ceph 块设备缓存的行为与行为良好的硬盘缓存一样。当操作系统发送阻碍或清空请求时，所有脏数据都会写入 Ceph OSD。这意味着，使用回写缓存和使用功能良好的物理硬盘和正确发送清除（即 Linux 内核 2.6.32 或更高版本）的虚拟机一样安全。缓存使用最早使用 (LRU) 算法，在回写模式中，它可以联合相邻的请求来获得更好的吞吐量。

Ceph 块设备支持回写缓存。若要启用回写缓存，可将 **rbd_cache = true** 设置为 Ceph 配置文件的 **[client]** 部分。默认情况下，**librbd** 不执行任何缓存。写入和读取直接进入存储集群，只有数据处于所有副本的磁盘中时写入才会返回。启用缓存时，写入会立即返回，除非存在超过 **rbd_cache_max_dirty** 未清空字节。在这种情况下，写入会触发 write-back 并拦截，直到清空了充足的字节为止。

Ceph 块设备支持直写缓存。您可以设置缓存的大小，您可以设置从回写缓存切换到直写缓存的目标和限制。若要启用 write-through 模式，可将 **rbd_cache_max_dirty** 设置为 0。这意味着，只有在数据处于所有副本的磁盘上时写入才会返回，但读取可能来自缓存。缓存位于客户端上的内存中，每个 Ceph 块设备镜像都有自己的内存。由于缓存对客户端而言是本地的，如果其他人访问该镜像，则没有一致性。在 Ceph 块设备上运行其他文件系统（如 GFS 或 OCFS）将无法用于启用缓存。

默认情况下，Ceph 块设备的 Ceph 配置设置必须在 Ceph 配置文件的 **[client]** 部分中设置，默认为 **/etc/ceph/ceph.conf**。

设置包括：

rbd_cache

描述

为 RADOS 块设备 (RBD) 启用缓存。

类型

布尔值

必需

否

默认

true

rbd_cache_size

描述

以字节为单位的 RBD 缓存大小。

类型

64 位整数

必需

否

默认

32 MiB

rbd_cache_max_dirty

描述

以字节为单位的 **脏** 限制，达到时缓存将触发回写。如果为 **0**，则使用直写缓存。

类型

64 位整数

必需

否

约束

必须小于 **rbd 缓存大小**。

默认

24 MiB

rbd_cache_target_dirty

描述

缓存开始将数据写入数据存储前的 **脏目标**。不要阻止写入到缓存。

类型

64 位整数

必需

否

约束

必须小于 **rbd cache max dirty**。

默认

16 MiB

rbd_cache_max_dirty_age

描述

在开始回写前，脏数据在缓存中的秒数。

类型

浮点值

必需

否

默认

1.0

rbd_cache_max_dirty_object

描述

对象的脏限制 - 设为 **0**，用于从 **rbd_cache_size** 自动计算。

类型

整数

默认

0

rbd_cache_block_writes_upfront

描述

如果为 **true**，它将在 **aio_write** 调用完成前阻止写入缓存。如果为 **false**，它将在调用 **aio_completion** 之前阻止。

类型

布尔值

默认

false

rbd_cache_writethrough_until_flush

描述

以直写模式开始，并在收到第一个 flush 请求后切换到回写模式。如果 rbd 上运行的虚拟机太旧而无法发送清空，如 Linux 中的 virtio 驱动程序 2.6.32 之前，启用此设置比较保守，但安全设置。

类型

布尔值

必需

否

默认

true

A.4. 块设备父级和子读选项

rbd_balance_snap_reads

描述

Ceph 通常从Primary OSD 读取对象。由于读取不可变，您可以启用此功能来平衡 Primary OSD 和副本之间的 snap 读取。

类型

布尔值

默认

false

rbd_localize_snap_reads

描述

rbd_balance_snap_reads 将随机化副本以读取快照。如果启用 **rbd_localize_snap_reads**，块设备将查看 CRUSH map，以查找最接近或本地 OSD 以读取快照。

类型

布尔值

默认

false

rbd_balance_parent_reads

描述

Ceph 通常从Primary OSD 读取对象。由于读取不可变，您可以启用此功能来平衡 Primary OSD 和副本之间的父读取。

类型

布尔值

默认

false

rbd_localize_parent_reads

描述

rbd_balance_parent_reads 将随机化副本以读取父项。如果启用 **rbd_localize_parent_reads**，块设备将查找 CRUSH map 来查找最接近或本地 OSD 以读取父级。

类型

布尔值

默认

true

A.5. 块设备读取预置选项

RBD 支持 read-ahead/prefetching 来优化小顺序读取。在虚拟机的情况下，这通常由客户机操作系统处理，但启动加载器可能不会产生高效的读取问题。如果禁用缓存，则会自动禁用 read-ahead。

rbd_readahead_trigger_requests

描述

触发 read-ahead 所需的连续读取请求数。

类型

整数

必填

否

默认

10

rbd_readahead_max_bytes

描述

read-ahead 请求的最大大小.如果为零，则禁用 read-ahead。

类型

64 位整数

必填

否

默认

512 KiB

rbd_readahead_disable_after_bytes

描述

从 RBD 镜像读取了这一字节后，对该镜像禁用 read-ahead，直到关闭为止。这允许客户机操作系统在启动后接管读头。如果为零，则启用 read-ahead。

类型

64 位整数

必填

否

默认

50 MiB

A.6. 块设备黑名单选项

rbd_blocklist_on_break_lock

描述

是否阻止其锁定的客户端被阻止。

类型

布尔值

默认

true

rbd_blocklist_expire_seconds

描述

blocklist 的秒数 - 为 OSD 默认设置为 0。

类型

整数

默认

0

A.7. 块设备日志选项

rbd_journal_order

描述

转换到计算日志对象最大大小的位数。该值介于 **12** 到 **64** 之间。

类型

32-bit Unsigned 整数

默认

24

rbd_journal_splay_width

描述

活动日志对象的数量。

类型

32-bit Unsigned 整数

默认

4

rbd_journal_commit_age

描述

提交时间间隔（以秒为单位）。

类型

双精确浮动点数

默认

5

rbd_journal_object_flush_interval

描述

每个日志对象每个待处理提交的最大数量。

类型

整数

默认

0

rbd_journal_object_flush_bytes

描述

每个日志对象最多待处理字节数。

类型

整数

默认

0

rbd_journal_object_flush_age

描述

等待提交的最大时间间隔（以秒为单位）。

类型

双精确浮动点数

默认

0

rbd_journal_pool

描述

为日志对象指定池。

类型

字符串

默认

""

A.8. 块设备配置覆盖选项

全局级别和池级别的块设备配置覆盖选项。

全局级别

可用密钥

rbd_qos_bps_burst

描述

所需的 IO 字节突发限制。

类型

整数

默认

0

rbd_qos_bps_limit

描述

每秒 IO 字节数所需的限制。

类型

整数

默认

0

rbd_qos_iops_burst

描述

IO 操作所需的突发限制。

类型

整数

默认

0

rbd_qos_iops_limit

描述

每秒所需的 IO 操作限制。

类型

整数

默认

0

rbd_qos_read_bps_burst

描述

所需的读字节突发限制。

类型

整数

默认

0

rbd_qos_read_bps_limit

描述

所需的每秒读取字节数限制。

类型

整数

默认

0

rbd_qos_read_iops_burst

描述

所需的读操作突发限制。

类型

整数

默认

0

rbd_qos_read_iops_limit

描述

每秒读取操作所需的限制。

类型

整数

默认

0

rbd_qos_write_bps_burst

描述

写入字节所需的突发限制。

类型

整数

默认

0

rbd_qos_write_bps_limit

描述

所需的每秒写入字节数限制。

类型

整数

默认

0

rbd_qos_write_iops_burst

描述

写入操作所需的突发限制。

类型

整数

默认

0

rbd_qos_write_iops_limit

描述

每秒写入操作的预期突发限制。

类型

整数

默认

0

以上键可用于以下目的：

rbd config global set *CONFIG_ENTITY KEY VALUE*

描述

设置全局级配置覆盖。

rbd config global get *CONFIG_ENTITY KEY*

描述

获取全局配置覆盖。

rbd config global list *CONFIG_ENTITY*

描述

列出全局级别配置覆盖。

rbd config global remove *CONFIG_ENTITY KEY*

描述

删除全局级配置覆盖。

池级别

rbd config pool set *POOL_NAME KEY VALUE*

描述

设置池级配置覆盖。

rbd config pool get *POOL_NAME KEY*

描述

获取池级配置覆盖。

rbd config pool list *POOL_NAME*

描述

列出池级配置覆盖。

rbd config pool remove *POOL_NAME KEY*

描述

删除池级配置覆盖。

注意

CONFIG_ENTITY 是全局、客户端或客户端 ID。***KEY*** 是配置键。***VALUE*** 是配置值。***POOL_NAME*** 是池的名称。

A.9. 块设备输入和输出选项

Red Hat Ceph Storage 的常规输入和输出选项。

rbd_compression_hint

描述

在写入操作中发送到 OSD 的 hint。如果设置为 **compressible**，OSD **bluestore_compression_mode** 的设置为 **passive**，OSD 会尝试压缩数据。如果设置为 **incompressible**，并且 OSD **bluestore_compression_mode** 设置为 **aggressive**，则 OSD 不会尝试压缩数据。

类型

Enum

必填

否

默认

none

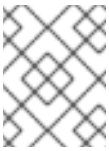
值

none, **compressible**, **incompressible**

rbd_read_from_replica_policy

描述

确定哪些 OSD 接收读操作的策略。如果设置为 **default**，则每个 PG 的 Primary OSD 将始终用于读取操作。如果设置为 **balance**，则读取操作将在副本集内发送到随机选择的 OSD。如果设置为 **localize**，则读取操作将发送到由 CRUSH map 和 **crush_location** 配置选项决定的最接近的 OSD，其中 **crush_location** 使用 **key=value** 表示。**key** 与 CRUSH map 的 key 一致。



注意

此功能要求存储集群配置与最新版本的 Red Hat Ceph Storage 的最低兼容 OSD 版本。

类型

Enum

必填

否

默认

default

值

default, **balance**, **localize**