SUSE Enterprise Storage 7.1

管理和操作指南

管理和操作指南

SUSE Enterprise Storage 7.1

作者：Tomáš Ba#ant、Alexandra Settle、Liam Proven

出版日期：2025-03-20

https://documentation.suse.com

版权所有 © 2020–2025 SUSE LLC 和贡献者。保留所有权利。

除非另有说明，否则本文档根据知识共享署名-相同方式共享 4.0 国际 (Creative Commons Attribution-ShareAlike 4.0 International, CC-BY-SA 4.0) 获得许可：https://creativecommons.org/licenses/by-sa/4.0/legalcode。

有关 SUSE 商标，请参见 http://www.suse.com/company/legal/ 。所有第三方商标均是其各自所有者的财产。商标符号（®、™ 等）代表 SUSE 及其关联公司的商标。星号 (*) 代表第三方商标。

本指南力求涵盖所有细节，但这不能确保本指南准确无误。SUSE LLC 及其关联公司、作者和译者对于可能出现的错误或由此造成的后果皆不承担责任。

目录

关于本指南 xviii

1 可用文档 xviii

2 提供反馈 xix

3 文档约定 xx

4 支持 xxi

SUSE Enterprise Storage 支持声明 xxii • 技术预览 xxiii

5 Ceph 贡献者 xxiii

6 本指南中使用的命令和命令提示符 xxiii

与 Salt 相关的命令 xxiv • 与 Ceph 相关的命令 xxiv • 一般的 Linux 命令 xxv • 附加信息 xxv

I CEPH DASHBOARD 1

1 关于 Ceph Dashboard 2

2 仪表盘的 Web 用户界面 3

2.1 登录 3

2.2 实用程序菜单 5

2.3 主菜单 6

2.4 内容窗格 7

2.5 常用 Web UI 功能 7

2.6 仪表盘控件 7

状态控件 7 • 容量控件 8 • 性能控件 9

3 管理 Ceph Dashboard 用户和角色 11
3.1 列出用户 11
3.2 添加新用户 11
3.3 编辑用户 12
3.4 删除用户 12
3.5 列出用户角色 13
3.6 添加自定义角色 13
3.7 编辑自定义角色 15
3.8 删除自定义角色 15
4 查看集群内部 16
4.1 查看集群节点 16
4.2 访问集群库存 16
4.3 查看 Ceph Monitor 17
4.4 显示服务 18
4.5 显示 Ceph OSD 19
添加 OSD 22
4.6 查看集群配置 25
4.7 查看 CRUSH 索引 25
4.8 查看 Manager 模块 26
4.9 查看日志 27
4.10 查看监控 27

5 管理存储池 28

5.1 添加新存储池 29

5.2 删除存储池 29

5.3 编辑存储池的选项 30

6 管理 RADOS 块设备 31

6.1 查看有关 RBD 的详细信息 31

6.2 查看 RBD 的配置 32

6.3 创建 RBD 32

6.4 删除 RBD 34

6.5 创建 RADOS 块设备快照 34

6.6 RBD 镜像 34

配置主集群和次集群 36 • 启用 `rbd-mirror` 守护进程 36 • 禁用镜像 37 • 引导对等 38 • 删除集群对等 39 • 在 Ceph Dashboard 中配置存储池复制 39 • 确认 RBD 映像复制是否有效 43

6.7 管理 iSCSI 网关 46

添加 iSCSI 目标 47 • 编辑 iSCSI 目标 49 • 删除 iSCSI 目标 49

6.8 RBD 服务质量 (QoS) 49

全局配置选项 50 • 针对新存储池配置选项 50 • 针对现有存储池配置选项 51 • 配置选项 51 • 使用新 RBD 映像创建 RBD QoS 选项 51 • 编辑现有映像的 RBD QoS 选项 51 • 在复制或克隆映像时更改配置选项 52

7 管理 NFS Ganesha 53

7.1 创建 NFS 导出项 54

7.2 删除 NFS 导出项 55

7.3 编辑 NFS 导出项 55

8 管理 CephFS 57

8.1 查看 CephFS 概览 57

9 管理对象网关 59

9.1 查看对象网关 59

9.2 管理对象网关用户 60

添加新网关用户 61 • 删除网关用户 63 • 编辑网关用户详细信息 63

9.3 管理对象网关存储桶 63

添加新的存储桶 63 • 查看存储桶详细信息 64 • 编辑存储桶 64 • 删除存储桶 65

10 手动配置 66

10.1 配置 TLS/SSL 支持 66

创建自我签名证书 67 • 使用 CA 签名的证书 67

10.2 更改主机名和端口号 68

10.3 调整用户名和密码 69

10.4 启用对象网关管理前端 69

10.5 启用 iSCSI 管理 70

10.6 启用单点登录 71

11 在命令行上管理用户和角色 73

11.1 管理密码策略 73

11.2 管理用户帐户 74

11.3 用户角色和权限 75

定义安全作用范围 75 • 指定用户角色 76

11.4 代理配置 79
使用反向代理访问仪表盘 79 • 禁用重定向 80 • 配置错误状态码 80 • HAProxy 示例配置 80
11.5 审计 API 请求 81
11.6 在 Ceph Dashboard 中配置 NFS Ganesha 82
配置多个 NFS Ganesha 集群 82
11.7 调试插件 83

II 集群操作 84

12 确定集群状态 85
12.1 检查集群的状态 85
12.2 检查集群健康状况 87
12.3 检查集群的使用率统计数据 96
12.4 检查 OSD 状态 97
12.5 检查填满的 OSD 98
12.6 检查 Monitor 状态 99
12.7 检查归置组状态 100
12.8 储存容量 100
12.9 监控 OSD 和归置组 103
监视 OSD 103 • 指定归置组集 104 • 正在互联 105 • 监控归置组状态 106 • 查找对象位置 111

13 操作任务 112
13.1 修改集群配置 112
13.2 添加节点 112

13.3 删除节点 113

13.4 OSD 管理 115

列出磁盘设备 115 • 擦除磁盘设备 116 • 使用 DriveGroups 规范添加 OSD。 116 • 删除 OSD 127 • 替换 OSD 128

13.5 将 Salt 主控端移至新节点 130

13.6 更新集群节点 132

软件源 132 • 软件源暂存 132 • Ceph 服务停机时间 132 • 运行更新 132

13.7 更新 Ceph 133

启动更新 133 • 监控更新 133 • 取消更新 134

13.8 停止或重引导集群 134

13.9 删除整个 Ceph 集群 135

14 Ceph 服务的操作 136

14.1 操作单个服务 136

14.2 操作服务类型 137

14.3 操作单个节点上的服务 137

标识服务和目标 137 • 操作一个节点上所有服务 138 • 操作一个节点上的单个服务 139 • 查询服务状态 139

14.4 关闭并重启动整个 Ceph 集群 139

15 备份和恢复 141

15.1 备份集群配置和数据 141

备份 `ceph-salt` 配置 141 • 备份 Ceph 配置 141 • 备份 Salt 配置 141 • 备份自定义配置 142

15.2 恢复 Ceph 节点 142

16 监控和告警 144

16.1 配置自定义或本地映像 145

16.2 更新监控服务 146

16.3 禁用监控 147

16.4 配置 Grafana 148

16.5 配置 Prometheus Manager 模块 148

配置网络接口 149 • 配置 scrape_interval 149 • 配置缓存 149 • 启用 RBD 映像监控 150

16.6 Prometheus 安全模型 151

16.7 Prometheus 告警管理器 SNMP 网关 151

III 在集群中存储数据 152

17 存储的数据管理 153

17.1 OSD 设备 154

设备类型 154

17.2 存储桶 162

17.3 规则集 165

迭代节点树 167 • firstn 和 indep 169

17.4 归置组 170

使用归置组 170 • 确定 PG_NUM 的值 172 • 设置归置组数量 173 • 获取归置组数量 174 • 获取集群的 PG 统计数据 174 • 获取卡住的 PG 统计数据 174 • 搜索归置组索引 174 • 检索归置组统计数据 175 • 洗刷归置组 175 • 设置归置组回填和恢复的优先顺序 175 • 还原丢失的对象 176 • 启用 PG 自动扩展器 176

17.5 CRUSH 索引操作 177
编辑 CRUSH 索引 177 • 添加或移动 OSD 178 • **ceph osd reweight** 与 **ceph osd crush reweight** 之间的差异 179 • 删除 OSD 180 • 添加存储桶 180 • 移动存储桶 181 • 删除存储桶 181
17.6 洗刷归置组 181
18 管理存储池 184
18.1 创建存储池 184
18.2 列出存储池 186
18.3 重命名存储池 186
18.4 删除存储池 186
18.5 其他操作 188
将存储池与应用关联 188 • 设置存储池配额 188 • 显示存储池统计数据 188 • 获取存储池值 190 • 设置存储池值 191 • 设置对象副本数 194
18.6 存储池迁移 196
限制 196 • 使用缓存层迁移 196 • 迁移 RBD 映像 199
18.7 存储池快照 200
创建存储池快照 200 • 列出存储池快照 200 • 删除存储池快照 200
18.8 数据压缩 201
启用压缩 201 • 存储池压缩选项 201 • 全局压缩选项 202
19 纠删码存储池 204
19.1 纠删码存储池的先决条件 204
19.2 创建示例纠删码存储池 204

19.3 纠删码配置 205
创建新纠删码配置 207 • 删除纠删码配置 208 • 显示纠删码配置的详细信息 208 • 列出纠删码配置 209

19.4 标记含 RADOS 块设备的纠删码存储池 209

20 RADOS 块设备 210

20.1 块设备命令 210
在副本存储池中创建块设备映像 210 • 在纠删码存储池中创建块设备映像 211 • 列出块设备映像 211 • 检索映像信息 211 • 调整块设备映像的大小 212 • 删除块设备映像 212

20.2 挂载和卸载 212
创建 Ceph 用户帐户 212 • 用户身份验证 213 • 准备 RADOS 块设备以供使用 213 • **rbdmap**：在引导时映射 RBD 设备 216 • 增大 RBD 设备的大小 217

20.3 快照 217
启用和配置 cephx 218 • 快照基础知识 218 • 快照分层 220

20.4 RBD 映像镜像 224
存储池配置 225 • RBD 映像配置 229 • 检查镜像状态 233

20.5 缓存设置 234

20.6 QoS 设置 235

20.7 预读设置 237

20.8 高级特性 237

20.9 使用旧内核客户端映射 RBD 239

20.10 启用块设备和 Kubernetes 240
在 Kubernetes 中使用 Ceph 块设备 243

IV 访问集群数据 247

21 Ceph 对象网关 248

21.1 对象网关限制和命名限制 248

存储桶限制 248 • 存储的对象的限制 248 • HTTP 报头限制 249

21.2 部署对象网关 249

21.3 操作对象网关服务 249

21.4 配置选项 249

21.5 管理对象网关的访问方式 249

访问对象网关 250 • 管理 S3 和 Swift 帐户 252

21.6 HTTP 前端 256

21.7 为对象网关启用 HTTPS/SSL 256

创建自我签名证书 256 • 配置使用 SSL 的对象网关 257

21.8 同步模块 258

配置同步模块 258 • 同步区域 259 • ElasticSearch 同步模块 260 • 云同步模块 264 • 存档同步模块 268

21.9 LDAP authentication 269

身份验证机制 269 • 要求 269 • 将对象网关配置为使用 LDAP 身份验证 270 • 使用自定义搜索过滤器限制用户访问权限 271 • 生成用于 LDAP 身份验证的访问令牌 272

21.10 存储桶索引分片 272

存储桶索引重分片 273 • 新存储桶的存储桶索引分片 275

21.11 OpenStack Keystone 集成 276

配置 OpenStack 277 • 配置 Ceph 对象网关 277

21.12 存储池归置和存储类别 280

显示归置目标 280 • 存储类别 280 • 配置区域组和区域 280 • 归置自定义 283 • 使用存储类别 284

21.13 多站点对象网关 285
要求和假设 286 • 配置主区域 286 • 配置次要区域 292 • 常规的对象网关维护 299 • 执行故障转移和灾难恢复 301

22 Ceph iSCSI 网关 303

22.1 ceph-iscsi 管理的目标 303
连接到 open-iscsi 303 • 连接 Microsoft Windows（Microsoft iSCSI 发起程序） 307 • 连接 VMware 314

22.2 结论 319

23 集群化文件系统 320

23.1 挂载 CephFS 320
准备客户端 320 • 创建秘密密钥文件 321 • 挂载 CephFS 321

23.2 卸载 CephFS 323

23.3 在 /etc/fstab 中挂载 CephFS 323

23.4 多个活动 MDS 守护进程（主动/主动 MDS） 323
使用主动/主动 MDS 323 • 增加 MDS 活动集群的大小 324 • 减小级别数量 325 • 手动将目录树关联到级别 325

23.5 管理故障转移 326
配置待机重放 326

23.6 设置 CephFS 配额 326
CephFS 配额限制 327 • 配置 CephFS 配额 327

23.7 管理 CephFS 快照 328
创建快照 329 • 删除快照 329

24 通过 Samba 导出 Ceph 数据 331

24.1 通过 Samba 共享导出 CephFS 331
配置和导出 Samba 软件包 331 • 单个网关示例 331 • 配置高可用性 335

24.2 加入 Samba 网关和 Active Directory 341
准备 Samba 安装 341 • 确认 DNS 342 • 解析 SRV 记录 342 • 配置 Kerberos 343 • 解析本地主机名 343 • 配置 Samba 344 • 加入 Active Directory 域 347 • 配置名称服务开关 347 • 启动服务 348 • 测试 winbindd 连接性 348

25 NFS Ganesha 350
25.1 创建 NFS 服务 351
25.2 启动或重启动 NFS Ganesha 352
25.3 列出 NFS 恢复存储池中的对象 352
25.4 创建 NFS 导出 352
25.5 确认 NFS 导出 353
25.6 挂载 NFS 导出 354
25.7 多个 NFS Ganesha 集群 354

V 与虚拟化工具集成 355

26 libvirt 和 Ceph 356
26.1 将 Ceph 配置为与 libvirt 搭配使用 356
26.2 准备 VM 管理器 357
26.3 创建 VM 358
26.4 配置 VM 358
26.5 摘要 361

27 Ceph 用作 QEMU KVM 实例的后端 362
27.1 安装 qemu-block-rbd 362
27.2 使用 QEMU 362

27.3 使用 QEMU 创建映像 363

27.4 使用 QEMU 调整映像大小 363

27.5 使用 QEMU 检索映像信息 363

27.6 使用 RBD 运行 QEMU 364

27.7 启用丢弃和 TRIM 364

27.8 设置 QEMU 缓存选项 365

VI 配置集群 367

28 Ceph 集群配置 368

28.1 配置 ceph.conf 文件 368

访问容器映像内部的 ceph.conf 368

28.2 配置数据库 369

配置部分和掩码 369 • 设置和读取配置选项 370 • 在运行时配置守护进程 370

28.3 config-key 储存 373

iSCSI 网关 373

28.4 Ceph OSD 和 BlueStore 374

配置自动缓存大小调整 374

28.5 Ceph 对象网关 375

一般设置 375 • 配置 HTTP 前端 384

29 Ceph Manager 模块 387

29.1 平衡器 388

“crush-compat”模式 388 • 数据平衡的规划和执行 388

29.2 启用遥测模块 390

30 cephx 身份验证 392

30.1 身份验证体系结构 392

30.2 主要管理 394

背景信息 395 • 管理用户 397 • 管理密钥环 402 • 命令行用法 405

A 基于上游“Pacific”小版本的 Ceph 维护更新 406

词汇表 407

关于本指南

本指南重点介绍在部署基本 Ceph 集群（第 2 天操作）之后，作为管理员需要处理的例行任务。此外，还介绍了访问 Ceph 集群中所存储数据的所有支持方法。

SUSE Enterprise Storage 7.1 是 SUSE Linux Enterprise Server 15 SP3 的一个扩展。它融合了 Ceph (http://ceph.com/) 存储项目的功能与 SUSE 的企业工程和支持。SUSE Enterprise Storage 7.1 为 IT 组织提供了部署分布式存储体系结构的能力，该体系结构可支持使用市售硬件平台的许多用例。

1 可用文档

注意：联机文档和最新更新

我们的产品文档可从 https://documentation.suse.com 获取，您也可以在此处找到最新更新，以及浏览或下载各种格式的文档。最新的文档更新以英语版本提供。

此外，您安装的系统的 `/usr/share/doc/manual` 下会提供产品文档。该文档包含在名为 `ses-manual_LANG_CODE` 的 RPM 软件包中。如果系统上尚未安装该包，请进行安装，例如：

```
# zypper install ses-manual_en
```

针对本产品提供的文档如下：

部署指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-deployment.html)

本指南重点介绍如何部署基本 Ceph 集群以及如何部署其他服务。此外，还介绍了从先前的产品版本升级到 SUSE Enterprise Storage 7.1 的步骤。

管理和操作指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-admin.html)

本指南重点介绍在部署基本 Ceph 集群（第 2 天操作）之后，作为管理员需要处理的例行任务。此外，还介绍了访问 Ceph 集群中所存储数据的所有支持方法。

安全强化指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-security.html)

本指南重点介绍如何确保集群的安全。

查错指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-troubleshooting.html)

本指南将带您了解运行 SUSE Enterprise Storage 7.1 时的各种常见问题以及与 Ceph 或对象网关等相关组件有关的其他问题。

SUSE Enterprise Storage for Windows 指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-windows.html)

本指南介绍如何使用 Windows 驱动程序集成、安装和配置 Microsoft Windows 环境和 SUSE Enterprise Storage。

2 提供反馈

欢迎您对此文档提供反馈和贡献。反馈渠道包括：

服务请求和支持

有关产品可用的服务和支持选项，请参见 http://www.suse.com/support/。

要创建服务请求，需在 SUSE Customer Center 注册一个 SUSE 订阅。请转到 https://scc.suse.com/support/requests 并登录，然后点击新建。

Bug 报告

在 https://bugzilla.suse.com/ 中报告文档问题。报告问题需要 Bugzilla 帐户。

要简化此过程，可以使用本文档 HTML 版本中的标题旁边的报告文档 Bug 链接。这样，就会在 Bugzilla 中预先选择正确的产品和类别，并添加当前章节的链接。然后，您便可以立即开始键入 Bug 报告。

贡献

要帮助改进本文档，请使用本文档 HTML 版本中的标题旁边的编辑源代码链接。这些链接会将您转到 GitHub 上的源代码，在其中可以创建拉取请求。参与贡献需要 GitHub 帐户。

有关本文档使用的文档环境的详细信息，请参见软件源的 README（网址：https://github.com/SUSE/doc-ses）。

邮件

您也可以将有关本文档中的错误以及相关反馈发送至：doc-team@suse.com。请在其中包含文档标题、产品版本和文档发布日期。此外，请包含相关的章节号和标题（或者提供 URL），并提供问题的简要说明。

3 文档约定

本文档中使用了以下通知和排版约定：

- /etc/passwd：目录名称和文件名
- PLACEHOLDER：将会使用实际的值替换 PLACEHOLDER
- PATH：环境变量
- **ls**、--help：命令、选项和参数
- user：用户或组的名称
- package_name：软件包的名称
- Alt 、 Alt – F1 ：按键或组合键。按键以大写字母显示，与键盘上的一样。
- 文件、 文件 › 另存为：菜单项，按钮
- AMD/Intel 本段内容仅与 AMD64/Intel 64 体系结构相关。箭头标记文本块的开始位置和结束位置。
  IBM Z, POWER 本段内容仅与 IBM Z 和 POWER 体系结构相关。箭头标记文本块的开始位置和结束位置。
- 第 1 章“示例章节”：到本指南中另一章节的交叉引用。
- 必须使用 root 特权运行的命令。您往往还可以在这些命令前加上 **sudo** 命令，以非特权用户身份来运行它们。

```
# command
> sudo command
```

- 可以由非特权用户运行的命令。

```
> command
```

- 注意

警告：警报通知

在继续操作之前，您必须了解的不可或缺的信息。向您指出有关安全问题、潜在数据丢失、硬件损害或物理危害的警告。

重要：重要通知

在继续操作之前，您必须了解的重要信息。

注意：注意通知

额外信息，例如有关软件版本差异的信息。

提示：提示通知

有用信息，例如指导方针或实用性建议。

- 精简通知

额外信息，例如有关软件版本差异的信息。

有用信息，例如指导方针或实用性建议。

4 支持

下面提供了 SUSE Enterprise Storage 的支持声明和有关技术预览的一般信息。有关产品生命周期的细节，请参见 https://www.suse.com/lifecycle。

如果您有权获享支持，可在 https://documentation.suse.com/sles-15/html/SLES-all/cha-adm-support.html 中查找有关如何收集支持票据所需信息的细节。

4.1 SUSE Enterprise Storage 支持声明

要获得支持，您需要一个适当的 SUSE 订阅。要查看为您提供的具体支持服务，请转到 https://www.suse.com/support/ 并选择您的产品。

支持级别的定义如下：

L1

问题判定，该技术支持级别旨在提供兼容性信息、使用支持、持续维护、信息收集，以及使用可用文档进行基本查错。

L2

问题隔离，该技术支持级别旨在分析数据、重现客户问题、隔离问题领域，并针对级别 1 不能解决的问题提供解决方法，或作为级别 3 的准备级别。

L3

问题解决，该技术支持级别旨在借助工程方法解决级别 2 支持所确定的产品缺陷。

对于签约的客户与合作伙伴，SUSE Enterprise Storage 将为除以下软件包外的其他所有软件包提供 L3 支持：

- 技术预览。
- 声音、图形、字体和作品。
- 需要额外客户合同的软件包。
- 模块 Workstation Extension 随附的某些软件包仅享受 L2 支持。
- 名称以 `-devel` 结尾的软件包（包含头文件和类似的开发人员资源）只能与其主软件包一起获得支持。

SUSE 仅支持使用原始软件包，即，未发生更改且未重新编译的软件包。

4.2 技术预览

技术预览是 SUSE 提供的旨在让用户大致体验未来创新的各种软件包、堆栈或功能。随附这些技术预览只是为了提供方便，让您有机会在自己的环境中测试新的技术。非常希望您能提供反馈！如果您测试了技术预览，请联系 SUSE 代表，将您的体验和用例告知他们。您的反馈对于我们的未来开发非常有帮助。

技术预览存在以下限制：

- 技术预览仍处于开发阶段。因此，它们的功能可能不完备、不稳定，或者在其他方面不适合用于生产。
- 技术预览不受支持。
- 技术预览可能仅适用于特定的硬件体系结构。
- 技术预览的细节和功能可能随时会发生变化。因此，可能无法升级到技术预览的后续版本，而只能进行全新安装。
- SUSE 可能会发现某个预览不符合客户或市场需求，或者未遵循企业标准。可随时从产品中删除技术预览。SUSE 不承诺未来将提供此类技术的受支持版本。

有关产品随附的技术预览的概述，请参见 https://www.suse.com/releasenotes/x86_64/SUSE-Enterprise-Storage/7.1 上的发行说明。

5 Ceph 贡献者

Ceph 项目及其文档是数百个贡献者和组织辛勤工作的成果。有关详细信息，请参见 https://ceph.com/contributors/。

6 本指南中使用的命令和命令提示符

作为 Ceph 集群管理员，您需要通过运行特定命令来配置和调整集群行为。您将需要运行以下几种类型的命令：

6.1 与 Salt 相关的命令

这些命令可帮助您部署 Ceph 集群节点、同时在数个（或所有）集群节点上运行命令，或在您添加或删除集群节点时为您提供协助。最常用的命令是 **ceph-salt** 和 **ceph-salt config**。您需要以 root 身份在 Salt 主控端节点上运行 Salt 命令。通过以下提示符来引入这些命令：

```
root@master #
```

例如：

```
root@master # ceph-salt config ls
```

6.2 与 Ceph 相关的命令

这些是较低级别的命令，用于在命令行上配置和微调集群及其网关的所有方面，例如 **ceph**、**cephadm**、**rbd** 或 **radosgw-admin**。

要运行与 Ceph 相关的命令，您需要拥有 Ceph 密钥的读取访问权限，而密钥的用户权限则定义您在 Ceph 环境内的权限。一种方案是以 root 身份（或通过 **sudo**）运行 Ceph 命令，并使用不受限的默认密钥环“ceph.client.admin.key”。

建议您使用更安全的方案，即为每个管理员用户创建限制性更高的单独密钥，并将其存放在用户可读取的目录中，例如：

```
~/.ceph/ceph.client.USERNAME.keyring
```

提示：Ceph 密钥的路径

要使用自定义管理员用户和密钥环，每次运行 **ceph** 命令（使用 -n client.USER_NAME 和 --keyring PATH/TO/KEYRING 选项）时，都需要指定该密钥的用户名和路径。

为避免出现此情况，请将这些选项包含在各个用户的 ~/.bashrc 文件中的 CEPH_ARGS 变量中。

虽然您可以在任何集群节点上运行与 Ceph 相关的命令，但建议您在管理节点上运行这些命令。本文档使用 cephuser 用户来运行命令，因此通过以下提示符来引入命令：

```
cephuser@adm >
```

例如：

```
cephuser@adm > ceph auth list
```

提示：特定节点的命令

如果文档指示您在集群节点上以特定角色来运行命令，应通过该提示符来寻址。例如：

```
cephuser@mon >
```

6.2.1 运行 `ceph-volume`

从 SUSE Enterprise Storage 7 开始，Ceph 服务以容器化方式运行。如果您需要在 OSD 节点上运行 **`ceph-volume`**，则需要在其前面追加 **`cephadm`** 命令，例如：

```
cephuser@adm > cephadm ceph-volume simple scan
```

6.3 一般的 Linux 命令

与 Ceph 无关的 Linux 命令（例如 **`mount`**、**`cat`** 或 **`openssl`**）可通过 `cephuser@adm >` 或 `#` 提示符来引入，具体取决于相关命令所需的特权。

6.4 附加信息

有关 Ceph 密钥管理的详细信息，请参见第 30.2 节 “主要管理”。

I Ceph Dashboard

1 关于 Ceph Dashboard 2

2 仪表盘的 Web 用户界面 3

3 管理 Ceph Dashboard 用户和角色 11

4 查看集群内部 16

5 管理存储池 28

6 管理 RADOS 块设备 31

7 管理 NFS Ganesha 53

8 管理 CephFS 57

9 管理对象网关 59

10 手动配置 66

11 在命令行上管理用户和角色 73

1 关于 Ceph Dashboard

Ceph Dashboard 是一个内置的基于 Web 的 Ceph 管理和监控应用，负责管理集群的各个方面和对象。在《部署指南》, 第 7 章 “使用 `ceph-salt` 部署引导集群”中部署基本集群后，将自动启用该仪表盘。

Ceph Dashboard for SUSE Enterprise Storage 7.1 添加了更多基于 Web 的管理功能（包括对 Ceph Manager 的监控和应用管理），让 Ceph 管理变得更轻松。现在，您不必了解复杂的 Ceph 相关命令，便可轻松管理和监控您的 Ceph 集群。您可以使用 Ceph Dashboard 的直观界面，也可以使用其内置 REST API。

Ceph Dashboard 模块使用 `ceph-mgr` 托管的 Web 服务器直观呈现有关 Ceph 集群的信息和统计数据。有关 Ceph Manager 的更多详细信息，请参见《部署指南》, 第 1 章 “SES 和 Ceph”，第 1.2.3 节 “Ceph 节点和守护进程”。

2 仪表盘的 Web 用户界面

2.1 登录

要登录 Ceph Dashboard，请将浏览器指向其 URL，包括端口号。运行以下命令可查找地址：

```
cephuser@adm > ceph mgr services | grep dashboard
"dashboard": "https://host:port/",
```

该命令将返回 Ceph Dashboard 所在的 URL。如果您有与此命令相关的问题，请参见《Troubleshooting Guide》, 第 10 章 “Troubleshooting the Ceph Dashboard”, 第 10.1 节 “Locating the Ceph Dashboard”。

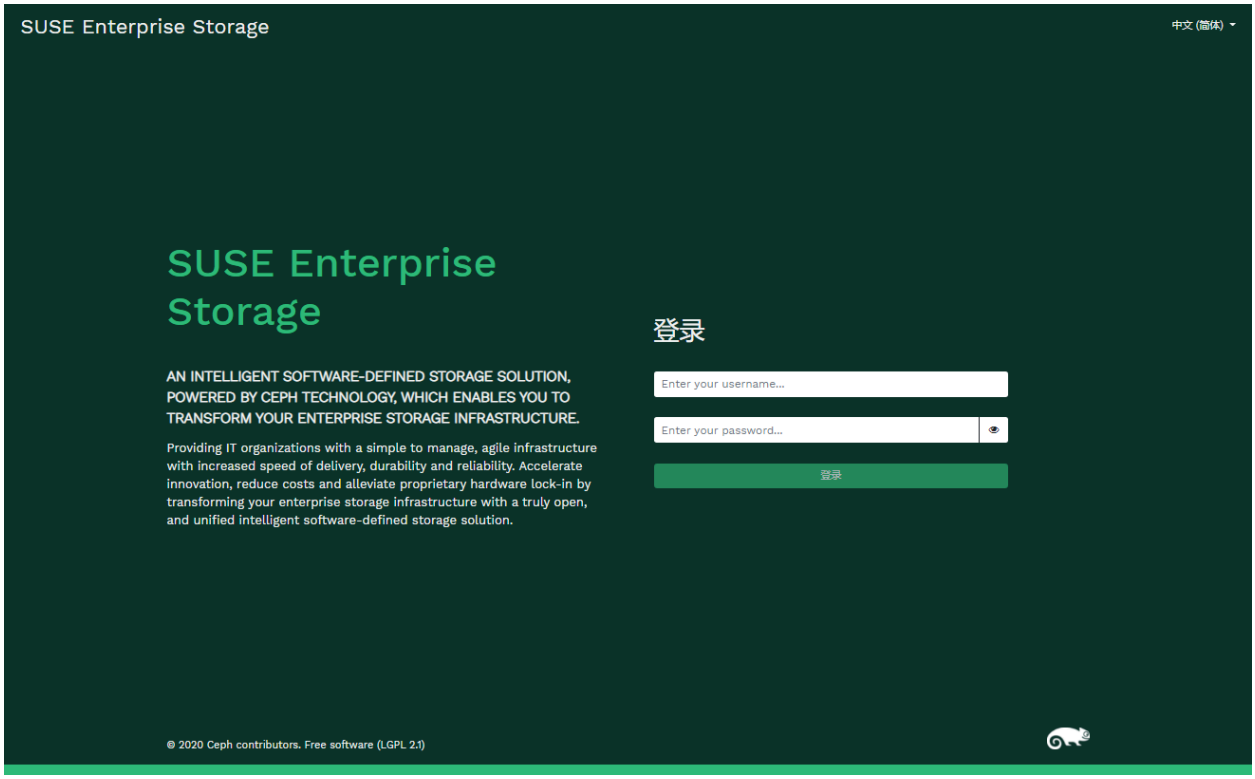


图 2.1： CEPH DASHBOARD 登录屏幕

使用您在集群部署期间创建的身份凭证登录（参见《部署指南》, 第 7 章 “使用 `ceph-salt` 部署引导集群”, 第 7.2.9 节 “配置 Ceph Dashboard 登录身份凭证”）。

提示：自定义用户帐户

如果您不想使用默认的 **admin** 帐户访问 Ceph Dashboard，请创建拥有管理员权限的自定义用户帐户。有关更多详细信息，请参见第 11 章 “在命令行上管理用户和角色”。

重要

一旦可以升级到新的 Ceph 主要版本（代码名称：Pacific），Ceph Dashboard 就会在顶部通知区域显示相关消息。要执行升级，请按照《部署指南》, 第 11 章 “从 SUSE Enterprise Storage 7 升级到版本 7.1”中的说明操作。

图 2.2： 关于新 SUSE ENTERPRISE STORAGE 版本的通知

仪表盘用户界面以图形方式分为几个**区块**：屏幕右上方的**实用程序菜单**，左侧的**主菜单**以及主要的**内容窗格**。

图 2.3： CEPH DASHBOARD 首页

2.2 实用程序菜单

实用程序菜单位于屏幕右上方。该菜单包含与仪表盘更为相关（相较于 Ceph 集群）的常规任务。点击相应选项可以访问以下主题：

- 将仪表盘的语言界面更改为：捷克语、德语、英语、西班牙语、法语、印度尼西亚语、意大利语、日语、韩语、波兰语、葡萄牙语（巴西）和中文。
- 任务和通知
- 查看文档、有关 REST API 的信息或有关仪表盘的更多信息。
- 用户管理和遥测配置。

> **注意**
>
> 有关用户角色的更详细的命令行描述，请参见第 11 章 “在命令行上管理用户和角色”。

- 登录配置；更改密码或登出。

2.3 主菜单

仪表盘的主菜单位于屏幕左侧。它涵盖下列主题：

仪表盘

返回 Ceph Dashboard 首页。

集群

查看有关主机、库存、Ceph Monitor、服务、Ceph OSD、集群配置、CRUSH 索引、Ceph Manager 模块、日志和监控的详细信息。

存储池

查看并管理集群存储池。

块设备

查看详细信息并管理 RADOS 块设备映像、镜像和 iSCSI。

NFS

查看并管理 NFS Ganesha 部署。

注意

如果未部署 NFS Ganesha，则会显示一条信息通知。请参见第 11.6 节 “在 Ceph Dashboard 中配置 NFS Ganesha”。

文件系统

查看并管理 CephFS。

对象网关

查看并管理对象网关的守护进程、用户和存储桶。

注意

如果未部署对象网关，则会显示一条信息通知。请参见第 10.4 节 “启用对象网关管理前端”。

2.4 内容窗格

内容窗格占据仪表盘屏幕的主要部分。仪表盘首页上显示了很多有用的控件，为您提供集群当前状态、容量及性能的概要信息。

2.5 常用 Web UI 功能

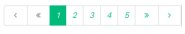
在 Ceph Dashboard 中，您经常会用到**列表** — 例如，存储池列表、OSD 节点列表或 RBD 设备列表。所有列表默认每五秒会自动刷新一次。下列常用控件可帮助您管理或调整这些列表：

点击 可手动触发列表的刷新。

点击 可显示或隐藏表格的相应列。

点击 并输入（或选择）要在一页上显示的行数。

在 中点击并键入要搜索的字符串可过滤行。

使用  可在列表跨多页显示时更改当前显示的页。

2.6 仪表盘控件

每个仪表盘控件显示与运行中 Ceph 集群特定方面相关的特定状态信息。有些控件是活跃链接，点击它们会将您重定向到其所代表主题的相关详细页面。

提示：鼠标悬停时显示更多详细信息

有些图形控件会在您将鼠标移到其上时显示更多详细信息。

2.6.1 状态控件

状态控件为您提供有关集群当前状态的简要概览。

图 2.4： 状态控件

集群状态

显示有关集群健康状况的基本信息。

主机

显示集群节点的总数。

Monitor

显示运行中 Monitor 数量及其仲裁。

OSD

显示 OSD 的总数及处于 **up** 和 **in** 状态的 OSD 数量。

Manager

显示处于活跃和待机状态的 Ceph Manager 守护进程的数量。

对象网关

显示运行中对象网关的数量。

元数据服务器

显示元数据服务器的数量。

iSCSI 网关

显示配置的 iSCSI 网关的数量。

2.6.2 容量控件

容量控件显示有关存储容量的简要信息。

图 2.5： 容量控件

基本容量

显示已用与可用基本存储容量的比例。

对象

显示集群中存储的数据对象数量。

PG 状态

显示按照状态分组的归置组图表。

存储池

显示集群中存储池的数量。

每个 OSD 的 PG 数

显示每个 OSD 的平均归置组数量。

2.6.3 性能控件

性能控件提供 Ceph 客户端的基本性能数据。

图 2.6： 性能控件

客户端读/写

每秒客户端读写操作次数。

客户端吞吐量

每秒与 Ceph 客户端之间传输的数据量（以字节为单位）。

恢复吞吐量

每秒恢复的数据吞吐量。

洗刷

显示洗刷（参见第 17.4.9 节 “洗刷归置组”）状态。其状态可以是非活跃、已启用或活跃。

3 管理 Ceph Dashboard 用户和角色

有关在命令行上通过 Ceph 命令执行仪表盘用户管理的说明，请参见第 11 章 “在命令行上管理用户和角色”。

本节说明如何使用仪表盘 Web 用户界面来管理用户帐户。

3.1 列出用户

点击实用程序菜单中的 ，然后选择用户管理。

该列表包含每个用户的用户名、全名、电子邮件、已指定角色的列表、角色是否启用以及密码超期日期。

图 3.1： 用户管理

3.2 添加新用户

点击表标题左上方的创建可添加新用户。输入其用户名、密码、全名（可选）和电子邮件。

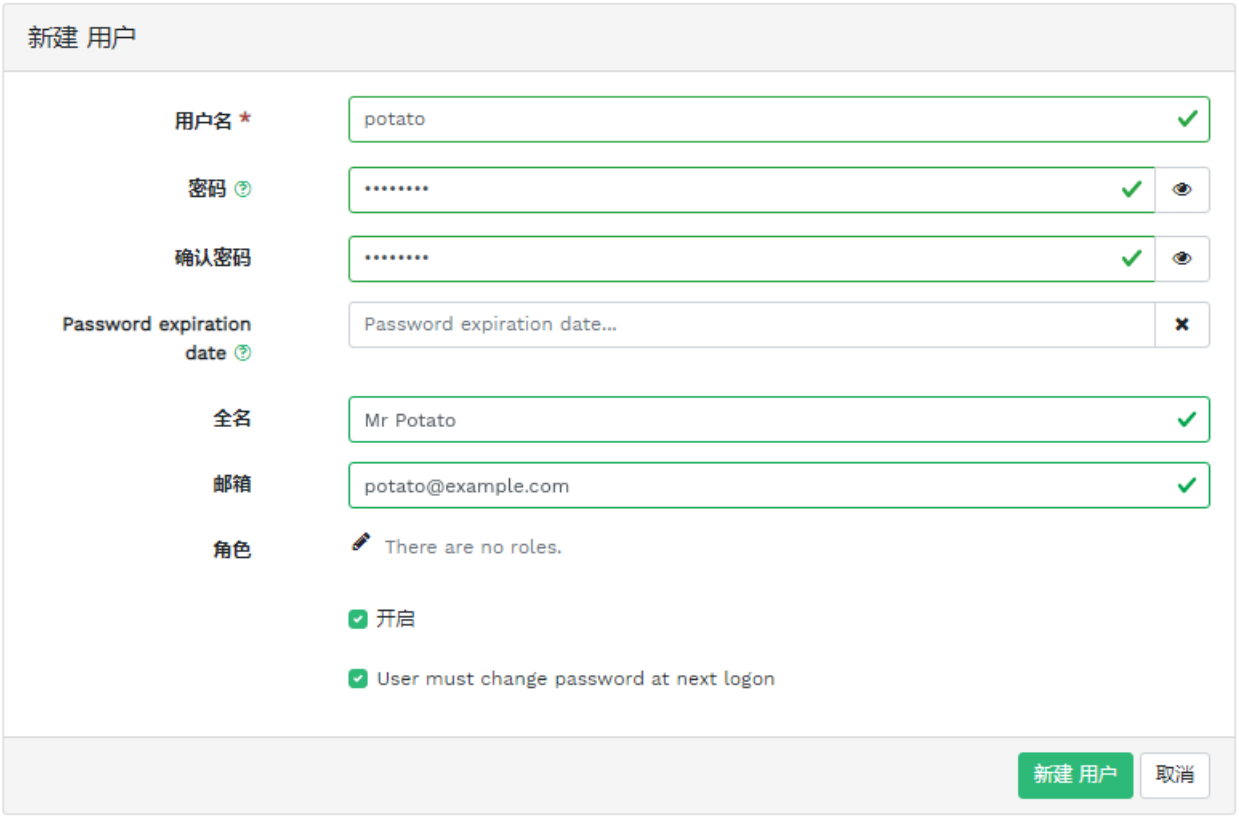


图 3.2： 添加用户

点击铅笔小图标可为用户指定预定义角色。点击创建用户进行确认。

3.3 编辑用户

点击表中某个用户对应的行以高亮显示选择。选择编辑，即可编辑该用户的详细信息。点击编辑用户进行确认。

3.4 删除用户

点击表中某个用户对应的行以高亮显示选择。选择编辑旁的下拉框，然后从列表中选择删除，即可删除用户帐户。激活是的，我确定复选框，然后点击删除用户进行确认。

3.5 列出用户角色

点击实用程序菜单中的 ，然后选择用户管理。然后点击角色选项卡。

该列表包含每个角色的名称、描述及其是否属于系统角色。

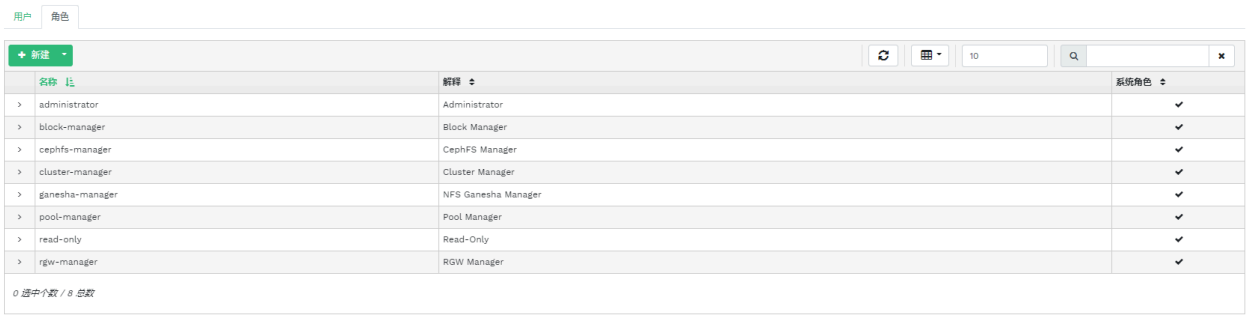


图 3.3： 用户角色

3.6 添加自定义角色

点击表标题左上方的创建可添加新的自定义角色。输入名称和描述，然后在权限旁选择适当的权限。

提示：清除自定义角色

如果您创建了自定义用户角色，并打算以后使用 `ceph-salt purge` 命令删除 Ceph 集群，则需要先清除自定义角色。有关详细信息，请参见第 13.9 节 “删除整个 Ceph 集群”。

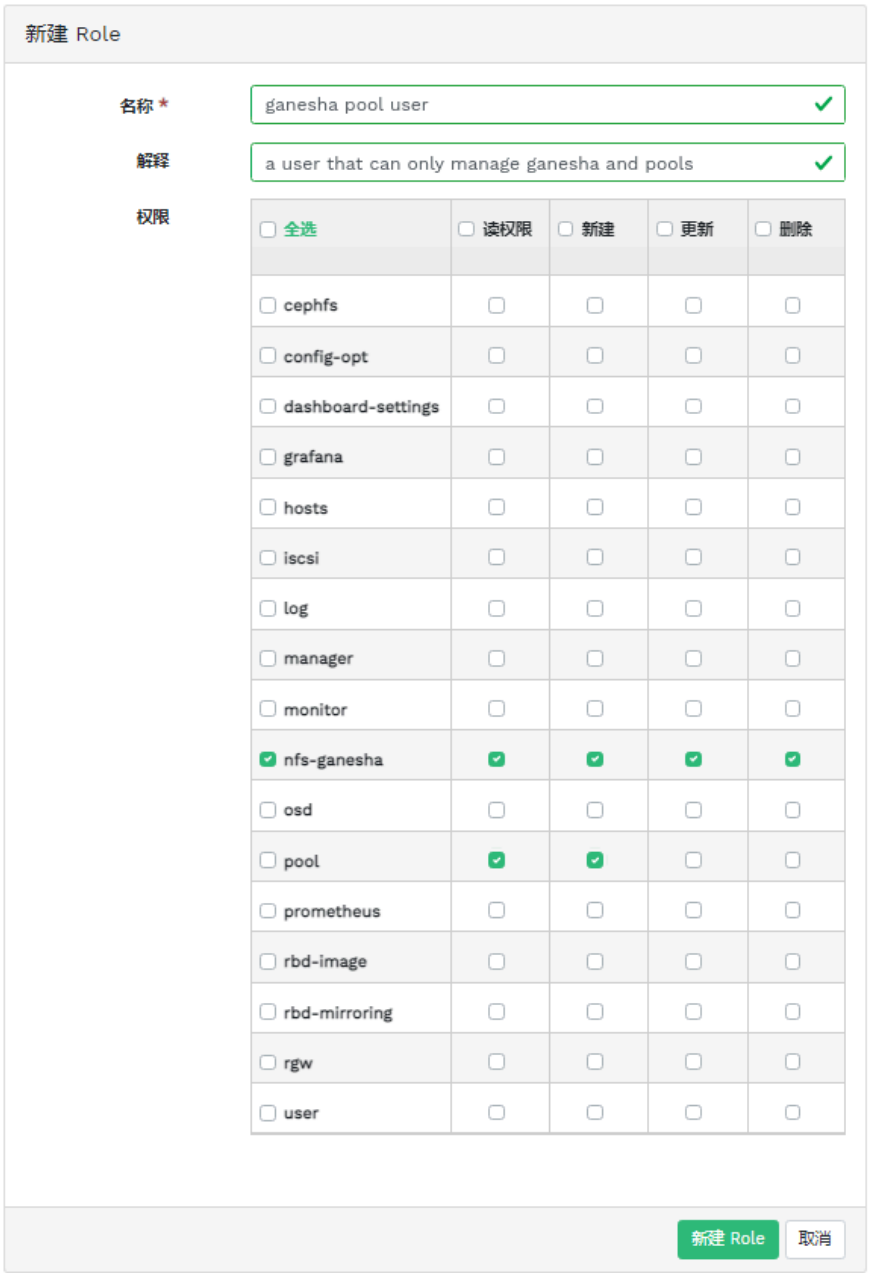


图 3.4： 添加角色

提示：多项目激活

通过激活主题名称前的复选框，您可以激活针对该主题的所有权限。通过激活全选复选框，您可以激活针对所有主题的所有权限。

点击创建角色进行确认。

3.7 编辑自定义角色

点击表中某个自定义角色对应的行以高亮显示选择。选择表标题左上方的编辑，即可编辑自定义角色的描述和权限。点击编辑角色进行确认。

3.8 删除自定义角色

点击表中某个角色对应的行以高亮显示所选项。选择编辑旁的下拉按钮，然后从列表中选择删除可删除角色。激活是的，我确定复选框，然后点击删除角色进行确认。

4 查看集群内部

您可以通过集群菜单项查看有关 Ceph 集群主机、库存、Ceph Monitor、服务、OSD、配置、CRUSH 索引、Ceph Manager、日志和监控文件的详细信息。

4.1 查看集群节点

点击集群 › 主机可查看集群节点列表。

图 4.1： 主机

点击主机名列中某个节点名称旁的下拉箭头可查看节点的性能详细信息。

服务列中会列出每个相关节点上正在运行的所有守护进程。点击守护进程名称可查看其详细配置。

4.2 访问集群库存

点击集群 › 库存可查看设备列表。该列表包含设备路径、类型、可用性、供应商、型号、大小和 OSD。

点击以在主机名列中选择节点名称。选择后，点击标识以标识正在运行主机的设备。这会让相应设备上的 LED 闪烁。选择此操作的时长（介于 1、2、5、10 或 15 分钟之间）。点击执行。

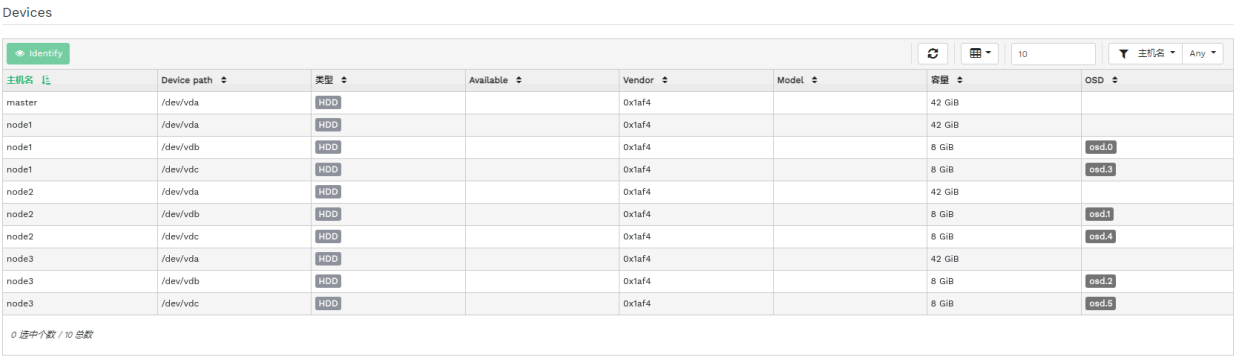


图 4.2： 服务

4.3 查看 Ceph Monitor

点击集群 › Monitor 可查看具有正在运行的 Ceph Monitor 的集群节点列表。内容窗格分为两个视图：状态和仲裁成员或非仲裁成员。

状态表显示有关运行中 Ceph Monitor 的一般统计数据，包括以下项目：

- 集群 ID
- monmap 修改时间
- monmap 版本编号
- 仲裁 con
- 仲裁 mon
- 所需 con
- 所需 mon

仲裁成员和非仲裁成员窗格包含每个 Monitor 的名称、排位号、公用 IP 地址和活动会话数。

点击名称列中的节点名称可查看相关的 Ceph Monitor 配置。

图 4.3： CEPH MONITOR

4.4 显示服务

点击集群 › 服务可查看每个可用服务的详细信息：crash、Ceph Manager 和 Ceph Monitor。该列表包含容器映像名称、容器映像 ID、正在运行项目的状态、大小以及上次刷新时间。

点击服务列中某个服务名称旁的下拉箭头可查看守护进程的详细信息。详细信息列表包含主机名、守护进程类型、守护进程 ID、容器 ID、容器映像名称、容器映像 ID、版本号、状态以及上次刷新时间。

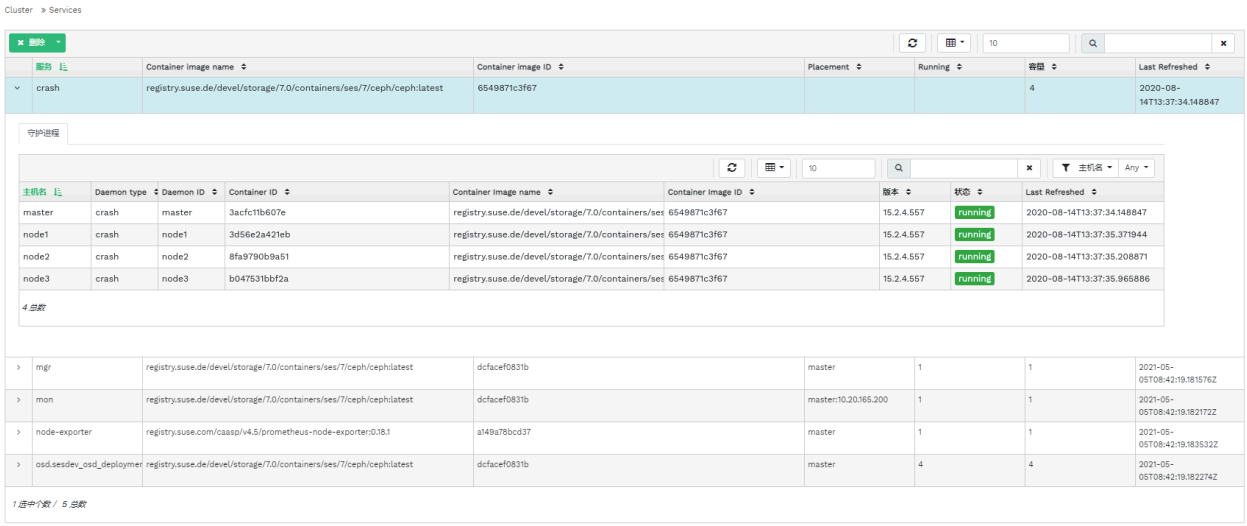


图 4.4： 服务

4.5 显示 Ceph OSD

点击集群 › OSD 可查看具有正在运行的 OSD 守护进程的节点列表。该列表包含每个节点的名称、ID、状态、设备类型、归置组数量、大小、使用率、不同时间的读/写图表和每秒读/写操作次数。

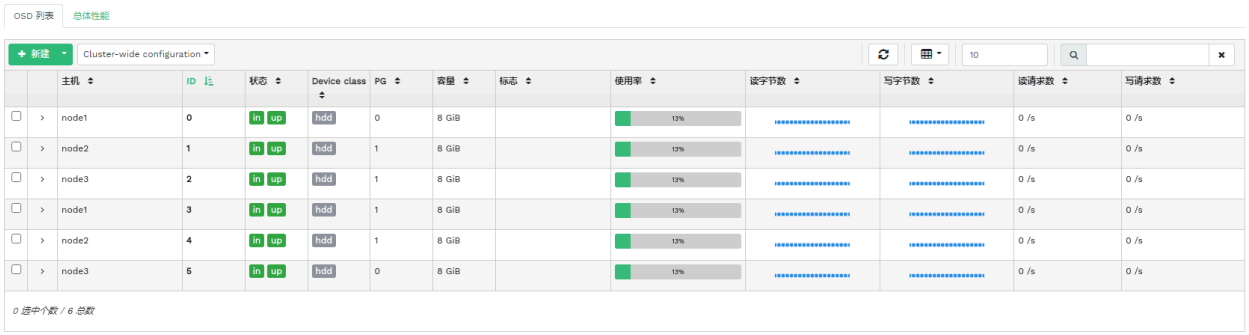


图 4.5： CEPH OSD

从表标题中的集群范围的配置下拉菜单中选择标志，可打开一个弹出窗口。此窗口包含适用于整个集群的标志列表。您可以激活或停用单个标志，然后点击提交进行确认。

图 4.6： OSD 标志

从表标题中的集群范围的配置下拉菜单中选择恢复优先级，可打开一个弹出窗口。此窗口包含适用于整个集群的 OSD 恢复优先级列表。您可以激活首选的优先级配置，并对下面的各个值进行微调。点击提交进行确认。

图 4.7： OSD 恢复优先级

点击主机列中某个节点名称旁的下拉箭头可查看扩展表，其中包含有关设备设置和性能的详细信息。您可以在数个选项卡之间切换，查看属性、元数据、设备健康状况、性能计数器各列表、读写操作次数的图形直方图，以及性能详细信息。

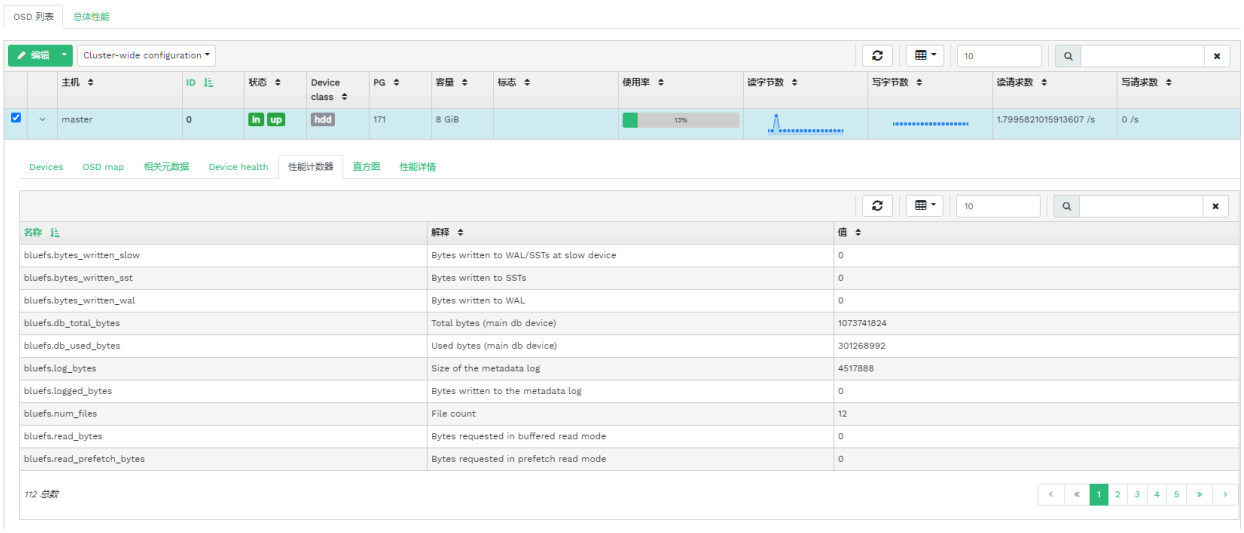


图 4.8： OSD 详细信息

提示：在 OSD 上执行特定任务

点击某个 OSD 节点名称后，将高亮显示表中的对应行。这意味着现在您可以在该节点上执行任务。您可以选择执行以下任何操作：编辑、创建、洗刷、深层洗刷、重设权重、标记为除名、标记为加入、标记为停用、标记为丢失、清除、销毁或删除。

点击表标题左上方创建按钮旁的向下箭头，然后选择要执行的任务。

4.5.1 添加 OSD

要添加新的 OSD，请执行以下步骤：

1. 确认某些集群节点的存储设备的状态为可用。然后点击表标题左上方的向下箭头并选择创建。此操作将打开创建 OSD 窗口。

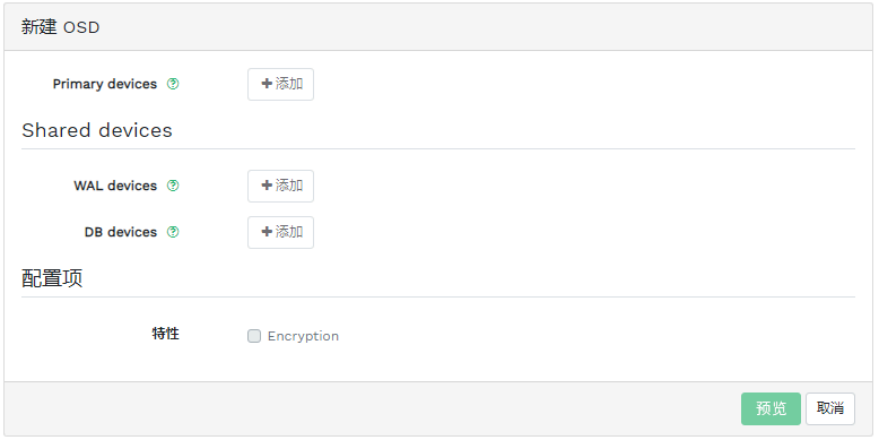


图 4.9： 创建 OSD

2. 要为 OSD 添加主存储设备，请点击添加。在添加存储设备之前，需要在主设备表的右上方指定过滤准则，例如类型 hdd。点击添加进行确认。

图 4.10： **添加主设备**

3. 在更新的创建 OSD 窗口中，可以选择添加共享的 WAL 和 BD 设备，或者启用设备加密。

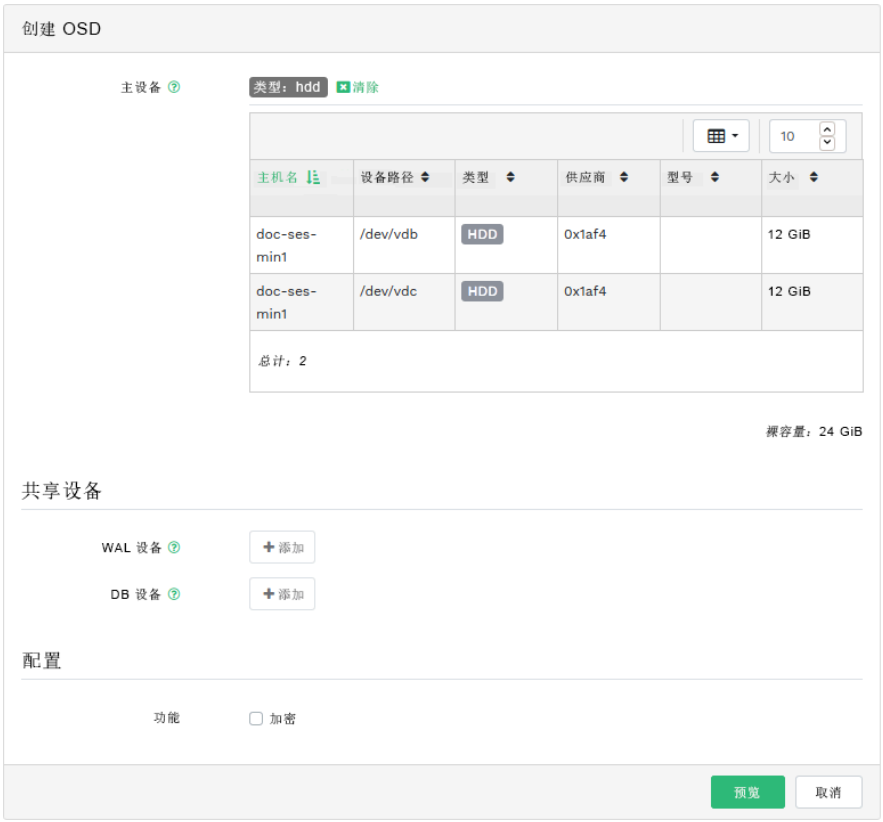


图 4.11： **创建添加了主设备的 OSD**

4. 点击预览可查看您之前所添加设备的 DriveGroups 规范预览。点击创建以确认。

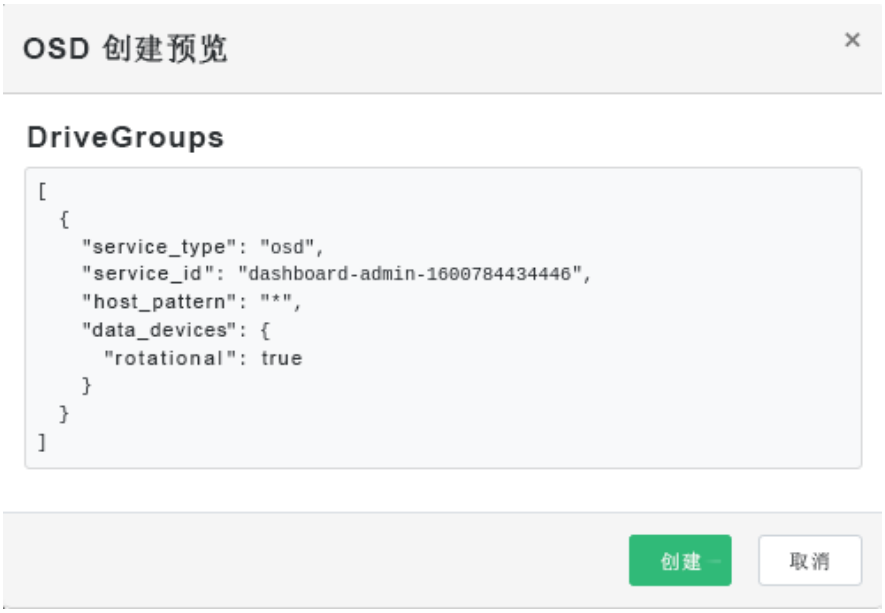


图 4.12：

5. 新设备将被添加到 OSD 列表中。

图 4.13： **新添加的 OSD**

注意

OSD 创建过程中无法查看进度。实际创建好 OSD 需要一些时间。部署 OSD 后，它们将显示在列表中。如果您要检查部署状态，请点击集群 › 日志查看相关日志。

4.6 查看集群配置

点击集群 › 配置可查看 Ceph 集群配置选项的完整列表。该列表包含选项的名称、其简短描述、当前值和默认值以及选项是否可编辑。

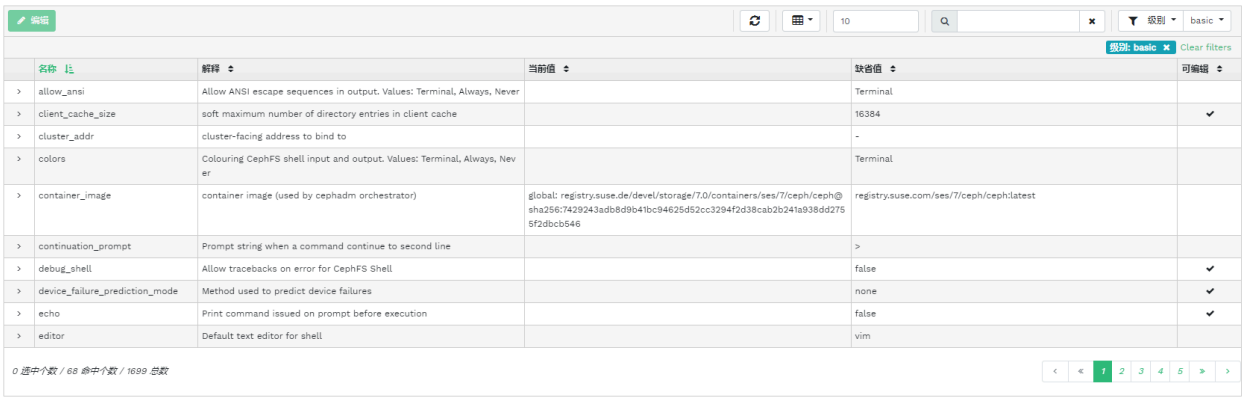


图 4.14： 集群配置

点击名称列中某个配置选项旁的下拉箭头可查看扩展表，其中包含有关该选项的详细信息，例如其值类型、允许的最小值和最大值、是否可以在运行时更新等。

高亮显示特定选项后，您可以通过点击表标题左上方的编辑按钮来编辑它的值。点击保存确认更改。

4.7 查看 CRUSH 索引

点击集群 › CRUSH 索引可查看集群的 CRUSH 索引。有关 CRUSH 索引的更多一般信息，请参见第 17.5 节 “CRUSH 索引操作”。

点击根、节点或单个 OSD 可查看更多详细信息，例如 CRUSH 权重、索引树的深度、OSD 的设备类型等等。

图 4.15： CRUSH 索引

4.8 查看 Manager 模块

点击集群 › Manager 扩展模块可查看可用 Ceph Manager 扩展模块的列表。每行包含扩展模块名称以及该扩展模块当前启用与否的信息。

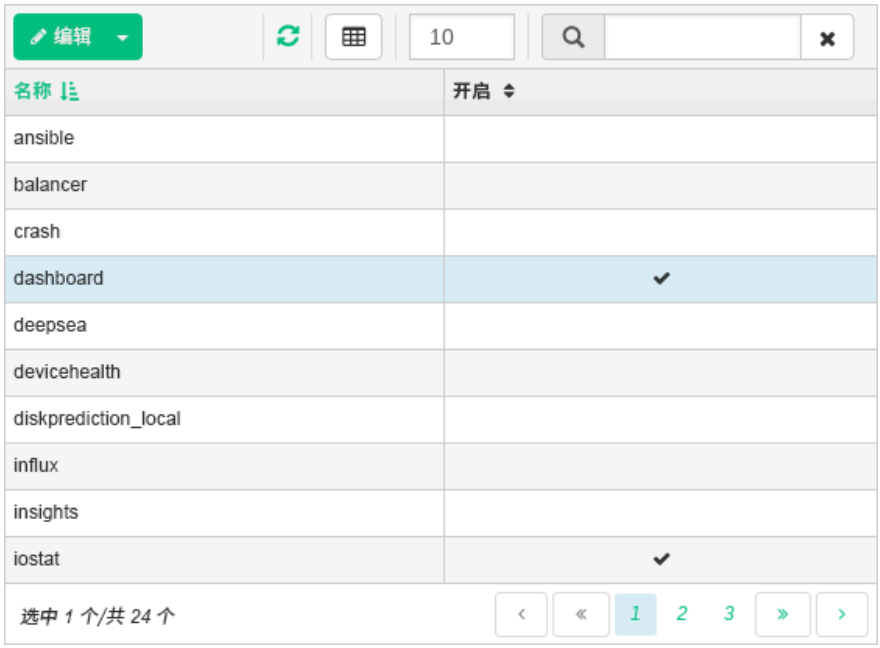


图 4.16： MANAGER 模块

点击名称列中某个模块旁的下拉箭头，可在下面的详细信息表中查看包含详细设置的扩展表。点击表标题左上方的编辑按钮可进行编辑。点击更新确认更改。

点击表标题左上方编辑按钮旁的下拉箭头可启用或禁用模块。

4.9 查看日志

点击集群 › 日志可查看集群最近日志项的列表。每行包含一个时戳、日志项的类型及自身所记录的消息。

点击审计日志选项卡可查看审计子系统的日志项。有关启用或禁用审计的命令，请参见第 11.5 节 “审计 API 请求”。

图 4.17： 日志

4.10 查看监控

点击集群 › 监控可管理和查看有关 Prometheus 告警的详细信息。

如果您启用了 Prometheus，则在此内容窗格中，您可以查看有关活动告警、所有告警或静默的详细信息。

注意

如果您未部署 Prometheus，则会显示一个信息标题以及转到相关文档的链接。

5 管理存储池

提示：存储池的更多信息

有关 Ceph 存储池的更多一般信息，请参见第 18 章 “管理存储池”。有关纠删码存储池特定的信息，请参见第 19 章 “纠删码存储池”。

要列出所有可用的存储池，请点击主菜单中的存储池。

该列表显示每个存储池的名称、类型、相关应用、归置组状态、副本个数、上次更改时间、纠删码配置、crush 规则集、使用率以及读/写统计数据。

图 5.1： 存储池列表

点击名称列中某个存储池名称旁的下拉箭头可查看扩展表，其中包含有关存储池的详细信息，例如一般详细信息、性能详细信息和配置。

5.1 添加新存储池

要添加新存储池，请点击存储池表左上方的创建。在存储池表单中，您可以输入存储池的名称、类型、其应用、压缩模式和配额（包括字节数上限和对象数量上限）。存储池表单会自行预先计算最适合此特定存储池的归置组数量。计算以集群中的 OSD 数量和选定存储池类型及其特定设置为依据。如果手动设置了归置组数量，计算出的数量会替换设置的数量。点击创建存储池进行确认。

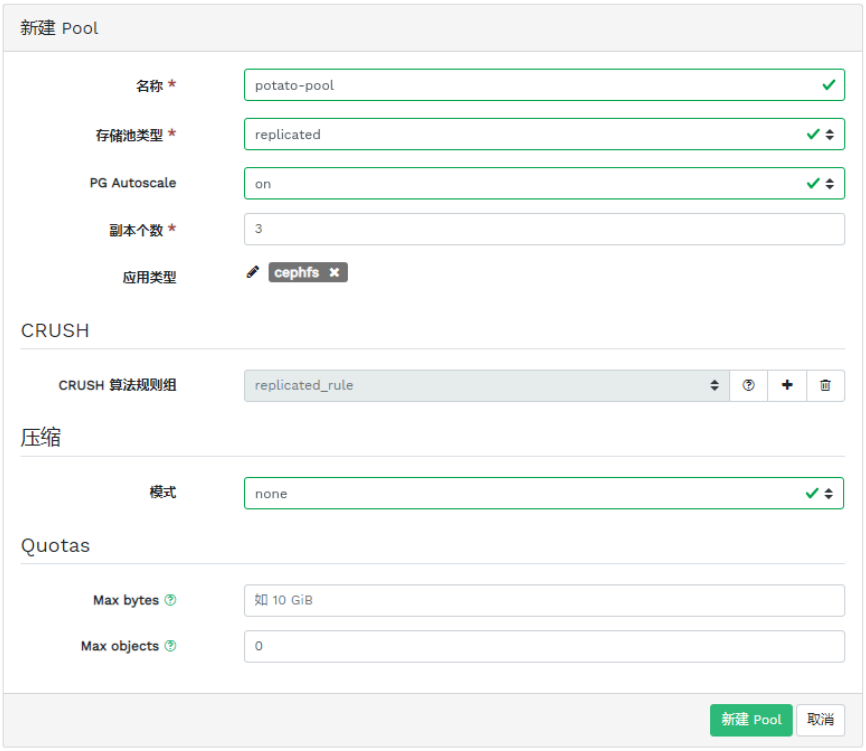


图 5.2： 添加新存储池

5.2 删除存储池

要删除存储池，请选择并高亮显示表中该存储池对应的行。点击创建按钮旁的下拉箭头，然后点击删除。

5.3 编辑存储池的选项

要编辑存储池选项，请选择表中该存储池对应的行，然后点击存储池表左上方的编辑。

您可以更改存储池的名称、增加归置组的数量、更改存储池的应用列表和压缩设置。点击编辑存储池进行确认。

6 管理 RADOS 块设备

要列出所有可用的 RADOS 块设备 (RBD)，请点击主菜单中的块设备 › 映像。

该列表显示有关设备的简要信息，例如设备名称、相关存储池名称、名称空间、设备大小、设备上的对象数量和大小、有关详细信息供应的详细信息以及父项。

图 6.1： RBD 映像列表

6.1 查看有关 RBD 的详细信息

要查看某个设备的更多详细信息，请点击表中对应的行：

详情 | 快照 | 配置项 | 性能

名称	example_rbd_device
存储池	rbd
数据存储池	-
创建	2021/4/30 下午3:02:38
容量	4 MiB
对象数	1
对象大小	4 MiB
特性	deep-flatten exclusive-lock fast-diff layering object-map
供给容量	0 B
总供给容量	0 B
条带单元	4 MiB
条带个数	1
父节点	-
块存储名字前缀	rbd_data.37ff2b17a0d1
order	22
Format Version	2

图 6.2： RBD 详细信息

6.2 查看 RBD 的配置

要查看某个设备的详细配置，请点击表中对应的行，然后点击下方表中的配置选项卡：

详情 快照 配置项 性能

名称	解释	密钥	来源
BPS 突发	指定的 IO 字节数突发上限。	rbd_qos_bps_burst	映像
BPS 上限	指定的每秒 IO 字节数上限。	rbd_qos_bps_limit	映像
IOPS 突发	指定的 IO 操作次数突发上限。	rbd_qos_iops_burst	映像
IOPS 上限	指定的每秒 IO 操作次数上限。	rbd_qos_iops_limit	映像
读 BPS 突发	指定的读取的字节数突发上限。	rbd_qos_read_bps_burst	映像
读 BPS 上限	指定的每秒内读取的字节数上限。	rbd_qos_read_bps_limit	映像
读 IOPS 突发	指定的读操作次数突发上限。	rbd_qos_read_iops_burst	映像
读 IOPS 上限	指定的每秒读操作次数上限。	rbd_qos_read_iops_limit	映像
写 BPS 突发	指定的写入的字节数突发上限。	rbd_qos_write_bps_burst	映像
写 BPS 上限	指定的每秒内写入的字节数上限。	rbd_qos_write_bps_limit	映像

图 6.3： RBD 配置

6.3 创建 RBD

要添加新设备，请点击表标题左上方的创建，然后在创建 RBD 屏幕上执行以下操作：

图 6.4： 添加新的 RBD

1. 输入新设备的名称。有关命名限制，请参见《部署指南》, 第 2 章 “硬件要求和建议”，第 2.11 节 “名称限制”。
2. 选择要在其中创建新 RBD 设备且指定了 `rbd` 应用的存储池。
3. 指定新设备的大小。
4. 为设备指定其他选项。要微调设备参数，请点击高级设置，然后输入对象大小、条带单元或条带个数的值。要输入服务质量 (QoS) 限制，请点击服务质量 (QoS)，然后输入限制。
5. 点击创建 RBD 进行确认。

6.4 删除 RBD

要删除设备，请选择表中该设备对应的行。点击创建按钮旁的下拉箭头，然后点击删除。点击删除 RBD 确认删除。

提示：将 RBD 移至回收站

删除 RBD 的操作无法撤消。如果您选择将设备移至回收站，那么稍后还可以将其恢复，只需在主表的回收站选项卡上选择该设备，然后点击表标题左上方的恢复。

6.5 创建 RADOS 块设备快照

要创建 RADOS 块设备快照，请选择表中该设备对应的行，将会显示详细的配置内容窗格。选择快照选项卡，然后点击表标题左上方的创建。输入快照的名称，然后点击创建 RBD 快照进行确认。

选择快照后，您可以在设备上执行其他操作，例如重命名、保护、克隆、复制或删除。回滚可根据当前快照恢复设备的状态。

图 6.5： RBD 快照

6.6 RBD 镜像

RADOS 块设备映像可以在两个 Ceph 集群之间异步镜像。您可以使用 Ceph Dashboard 在两个或更多集群之间配置 RBD 映像复制。此功能有两种模式：

基于日志

此模式使用 RBD 日志映像功能来确保集群之间的复制在时间点和崩溃时保持一致。

基于快照

此模式使用定期安排或手动创建的 RBD 映像镜像快照，以在集群之间复制崩溃时保持一致的 RBD 映像。

镜像是基于对等集群中的每个存储池进行配置的，可以对存储池中的特定映像子集进行配置，也可以配置为在仅使用基于日志的镜像时自动镜像存储池中的所有映像。

镜像是使用 **rbd** 命令配置的，SUSE Enterprise Storage 7.1 中默认会安装该工具。`rbd-mirror` 守护进程负责从 remote 对等集群提取映像更新，并将它们应用于 local 集群中的映像。有关启用 第 6.6.2 节 “启用 rbd-mirror 守护进程” 守护进程的详细信息，请参见`rbd-mirror`。

根据复制需求，RBD 块设备镜像可以配置为单向或双向复制：

单向复制

当数据仅会从主集群镜像到次集群时，`rbd-mirror` 守护进程仅在次集群上运行。

双向复制

当数据从一个集群上的主映像镜像到另一个集群上的非主映像（反之亦然）时，`rbd-mirror` 守护进程将在两个集群上运行。

重要

`rbd-mirror` 守护进程的每个实例都必须能够同时连接到本地和远程 Ceph 集群，例如所有 Monitor 和 OSD 主机。此外，网络必须在两个数据中心之间有足够的带宽来处理镜像工作负载。

提示：一般信息

有关 RADOS 块设备镜像的一般信息和命令行方法，请参见第 20.4 节 “RBD 映像镜像”。

6.6.1 配置主集群和次集群

主集群是用于创建包含映像的原始存储池的集群。**次**集群是用于从**主**集群复制存储池或映像的集群。

注意：相对命名

在复制上下文中，**主**与**次**这两个术语有可能是相对的，因为相较于集群来说，它们与单个存储池更为相关。例如，在双向复制中，可将一个存储池从**主**集群镜像到**次**集群，与此同时，也可以将另一个存储池从**次**集群镜像到**主**集群。

6.6.2 启用 rbd-mirror 守护进程

以下过程说明如何使用 **rbd** 命令来执行配置镜像的基本管理任务。镜像在 Ceph 集群中逐池进行配置。

两个对等集群上都应执行存储池配置步骤。为清楚起见，这些过程假设名为“primary”和“secondary”的两个集群可从单台主机访问。

`rbd-mirror` 守护进程负责执行实际的集群数据复制。

1. 重命名 `ceph.conf` 和密钥环文件，并将其从主要主机复制到次要主机：

```
cephuser@secondary > cp /etc/ceph/ceph.conf /etc/ceph/primary.conf
cephuser@secondary > cp /etc/ceph/ceph.admin.client.keyring \
 /etc/ceph/primary.client.admin.keyring
cephuser@secondary > scp PRIMARY_HOST:/etc/ceph/ceph.conf \
 /etc/ceph/secondary.conf
cephuser@secondary > scp  PRIMARY_HOST:/etc/ceph/ceph.client.admin.keyring
 \
 /etc/ceph/secondary.client.admin.keyring
```

2. 要使用 **rbd** 针对存储池启用镜像，请指定 **mirror pool enable**、存储池名称和镜像模式：

```
cephuser@adm > rbd mirror pool enable POOL_NAME MODE
```

注意

镜像模式可以是 `image` 或 `pool`。例如：

```
cephuser@secondary > rbd --cluster primary mirror pool enable image-
pool image
cephuser@secondary > rbd --cluster secondary mirror pool enable
 image-pool image
```

3. 在 Ceph Dashboard 上，导航到块设备 › 镜像。守护进程表的左侧会显示处于活跃状态的运行中 `rbd-mirror` 守护进程及其健康状况。

图 6.6： 运行 `rbd-mirror` 守护进程

6.6.3 禁用镜像

要使用 **`rbd`** 针对存储池禁用镜像，请指定 **`mirror pool disable`** 命令和存储池名称：

```
cephuser@adm > rbd mirror pool disable POOL_NAME
```

使用这种方法对存储池禁用镜像时，还会对已为其明确启用镜像的所有映像（该存储池中）禁用镜像。

6.6.4 引导对等

为了使 `rbd-mirror` 发现其对等集群，需要将对等注册到存储池，并需要创建用户帐户。此过程可以使用 **`rbd`** 通过 **`mirror pool peer bootstrap create`** 和 **`mirror pool peer bootstrap import`** 命令自动完成。

要使用 **`rbd`** 手动创建新的引导令牌，请指定 **`mirror pool peer bootstrap create`** 命令、存储池名称以及描述本地集群的可选站点名称：

```
cephuser@adm > rbd mirror pool peer bootstrap create [--site-name local-site-
name] pool-name
```

`mirror pool peer bootstrap create` 的输出将是应提供给 **`mirror pool peer bootstrap import`** 命令的令牌。例如，在主集群上：

```
cephuser@adm > rbd --cluster primary mirror pool peer bootstrap create --site-
name primary
  image-pool
 eyJmc2lkIjoiOWY1MjgyZGItYjg5OS00NTk2LTgwOTgtMzIwYzFmYzM5NmYzIiwiY2xpZW50X2lkIjoicmJkL
 \

 W1pcnJvci1wZWVyIiwia2V5IjoiQVFBUnczOWQwdkhvQmhBQVlMM1I4RmR5dHNJQU50bkFTZ0lOTVE9PSIsIm1vbl9o
 \
  joiW3YyOjE5Mi4xNjguMS4zOjY4MjAsdjE6MTkyLjE2OC4xLjM6NjgyMV0ifQ==
```

要使用 **`rbd`** 命令手动导入另一个集群创建的引导令牌，请指定 **`mirror pool peer bootstrap import`** 命令、存储池名称、所创建令牌的文件路径（或设为“-”以从标准输入读取），以及描述本地集群的可选站点名称和镜像方向（默认设为 `rx-tx` 以进行双向镜像，但也可设为 `rx-only` 以进行单向镜像）：

```
cephuser@adm > rbd mirror pool peer bootstrap import [--site-name local-site-
name] \
[--direction rx-only or rx-tx] pool-name token-path
```

例如，在次集群上：

```
cephuser@adm > cat >>EOF < token
```

```
eyJmc2lkIjoiOWY1MjgyZGItYjg5OS00NTk2LTgwOTgtMzIwYzFmYzM5NmYzIiwiY2xpZW50X2lkIjoicmJkLW1pcn
 \
Jvci1wZWVyIiwia2V5IjoiQVFBUnczOWQwdkhvQmhBQVlMM1I4RmR5dHNJQU50bkFTZ0lOTVE9PSIsIm1vbl9ob3N0I
 \
joiW3YyOjE5Mi4xNjguMS4zOjY4MjAsdjE6MTkyLjE2OC4xLjM6NjgyMV0ifQ==
EOF
cephuser@adm > rbd --cluster secondary mirror pool peer bootstrap import --site-
name secondary image-pool token
```

6.6.5 删除集群对等

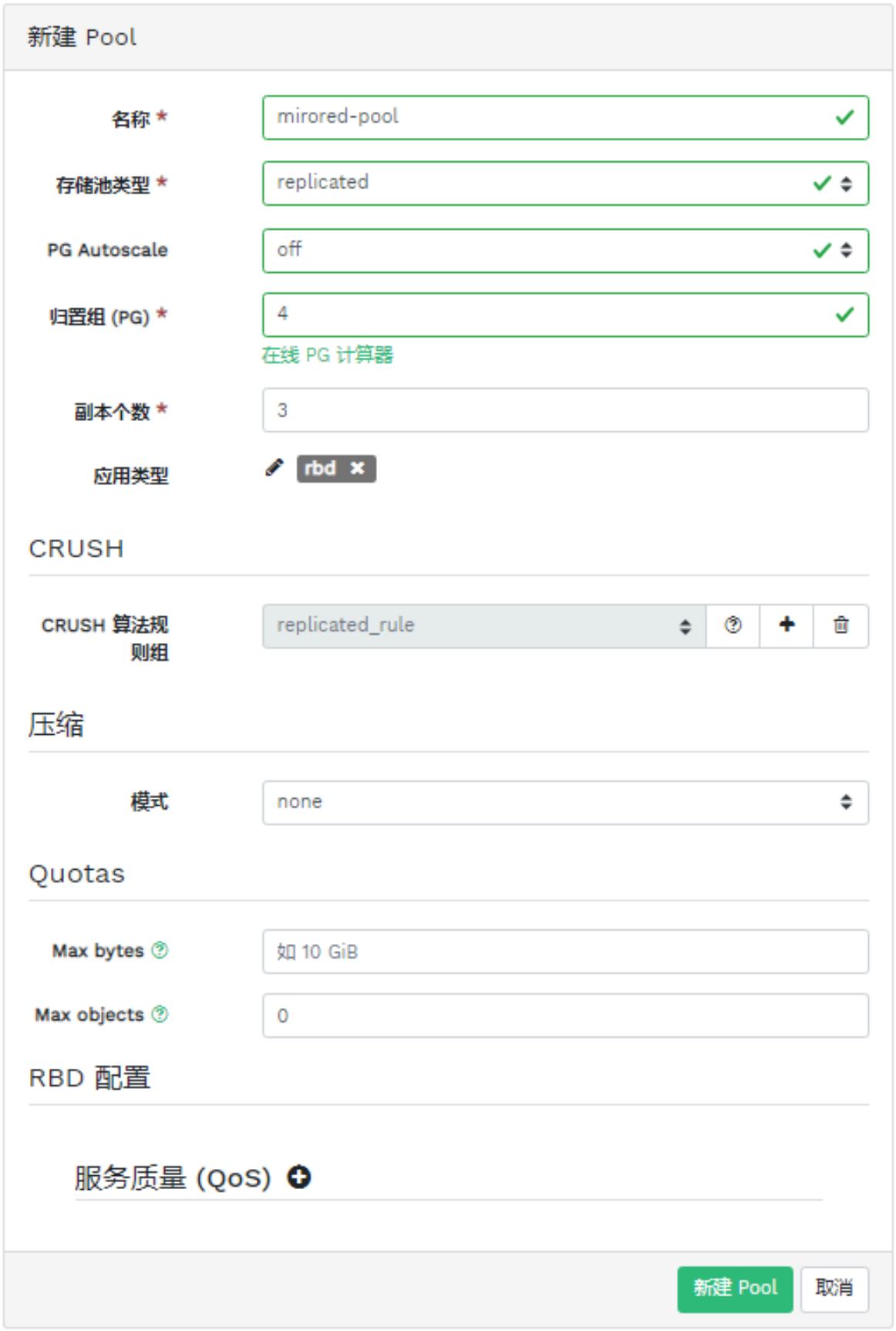
要使用 **rbd** 命令删除镜像对等 Ceph 集群，请指定 **mirror pool peer remove** 命令、存储池名称和对等 UUID（可通过 **rbd mirror pool info** 命令获得）：

```
cephuser@adm > rbd mirror pool peer remove pool-name peer-uuid
```

6.6.6 在 Ceph Dashboard 中配置存储池复制

`rbd-mirror` 守护进程需要拥有主集群的访问权限，才能镜像 RBD 映像。继续之前，请确保已按照第 6.6.4 节 “引导对等”中的步骤操作。

1. 在**主**和**次**两个集群上，创建名称相同的存储池并为其指定 `rbd` 应用。有关创建新存储池的更多详细信息，请参见第 5.1 节 “添加新存储池”。


新建 Pool
名称 *
mirored-pool
存储池类型 *
replicated
PG Autoscale
off
归置组 (PG) *
4
在线 PG 计算器
副本个数 *
3
应用类型
rbd
CRUSH
CRUSH 算法规则组
replicated_rule
压缩
模式
none
Quotas
Max bytes
如 10 GiB
Max objects
0
RBD 配置
服务质量 (QoS)
新建 Pool
取消

2. 在**主**和**次**两个集群的仪表盘上，导航到块设备 › 镜像。在存储池表右侧点击要复制的存储池的名称，然后点击编辑模式，并选择复制模式。在此示例中，我们将使用**存储池**复制模式，也就是说将复制给定存储池中的所有映像。点击更新进行确认。

图 6.8： 配置复制模式

重要：主集群上的错误或警报

更新复制模式后，右侧相应的列中会显示错误或警告标志。这是因为尚未为存储池指定用于复制的同伴用户。对于**主**集群，请省略此标志，因为我们只需为**次**集群指定同伴用户。

3. 在**次**集群的仪表盘上，导航到块设备 › 镜像。通过选择添加对等添加存储池镜像对等。提供**主**集群的详细信息：

图 6.9： 添加对等身份凭证

集群名称

用于识别主集群的任意唯一字符串，如“primary”。该集群名称不得与实际环境中次集群的名称相同。

CephX ID

您创建作为镜像同伴的 Ceph 用户 ID。此示例中为“rbd-mirror-peer”。

Monitor 地址

主集群 Ceph Monitor 节点的 IP 地址的逗号分隔列表。

CephX 密钥

与同伴用户 ID 相关的密钥。您可以通过在主集群上运行以下示例命令来获得该密钥：

```
cephuser@adm > ceph auth print_key pool-mirror-peer-name
```

点击提交进行确认。

存储池

编辑模式 | 10

名称	模式	前导符	# 本地	# 远程	健康状况
example_rbd_pool	存储池	292255	2	2	OK
mirrored-pool	存储池	292255	0	0	OK
pool3	映像	292255	2	2	OK
pool4	存储池	292255	1	1	OK

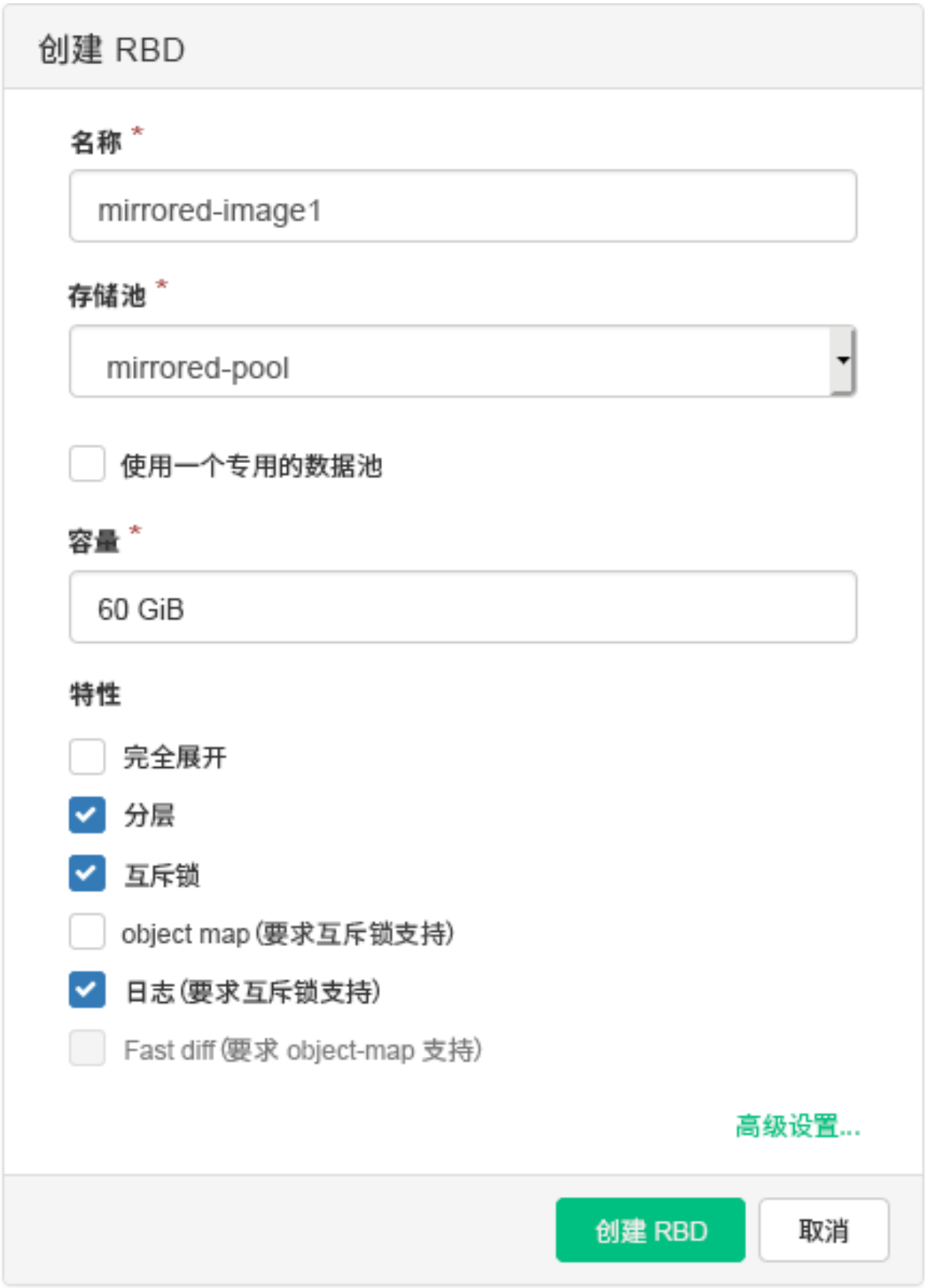
选中 1 个/共 4 个

图 6.10： 副本存储池列表

6.6.7 确认 RBD 映像复制是否有效

如果 `rbd-mirror` 守护进程正在运行，在 Ceph Dashboard 上配置 RBD 映像复制后，就需要确认复制实际上是否有效：

1. 在**主**集群的 Ceph Dashboard 上创建 RBD 映像，将您已创建用于复制目的的存储池设为其父存储池。为映像启用互斥锁和日志特性。有关如何创建 RBD 映像的详细信息，请参见第 6.3 节 “创建 RBD”。


创建 RBD
名称 *
mirrored-image1
存储池 *
mirrored-pool
使用一个专用的数据池
容量 *
60 GiB
特性
完全展开
分层
互斥锁
object map (要求互斥锁支持)
日志 (要求互斥锁支持)
Fast diff (要求 object-map 支持)
高级设置...
创建 RBD
取消

2. 创建要复制的映像后，打开**次**集群的 Ceph Dashboard，并导航到块设备 › 镜像。右侧的存储池表将在 # 远程所示数量的映像中反映该变化，并会同步 # 本地所示数量的映像。

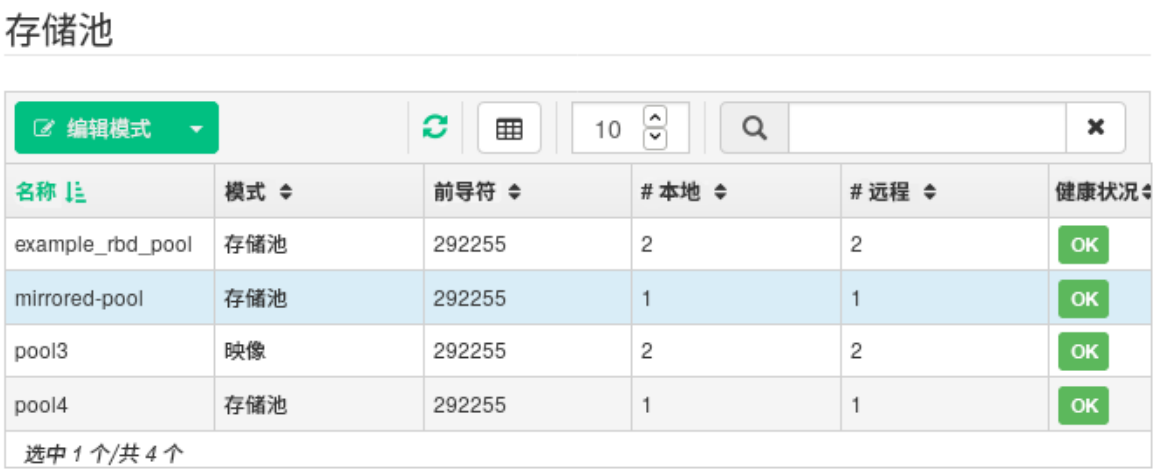


图 6.12： **同步的新 RBD 映像**

提示：复制进度

页面底部的映像表会显示 RBD 映像的复制状态。问题选项卡包含可能的问题，正在同步选项卡显示映像复制的进度，已就绪选项卡列出所有已成功复制的映像。

图 6.13： **RBD 映像的复制状态**

3. 在**主**集群上，将数据写入 RBD 映像。在**次**集群的 Ceph Dashboard 上，导航到块设备 › 映像，监视相应映像大小的增长是否与主集群上写入的数据一致。

6.7 管理 iSCSI 网关

提示：iSCSI 网关的更多信息

有关 iSCSI 网关的更多一般信息，请参见第 22 章 “Ceph iSCSI 网关”。

要列出所有可用网关和映射的映像，请点击主菜单中的块设备 › iSCSI。概述选项卡即会打开，列出当前配置的 iSCSI 网关和映射的 RBD 映像。

网关表列出每个网关的状态、iSCSI 目标数量及会话数量。映像表列出每个映射的映像的名称、相关存储池名称、后备存储类型及其他统计数据详细信息。

目标选项卡列出当前配置的 iSCSI 目标。

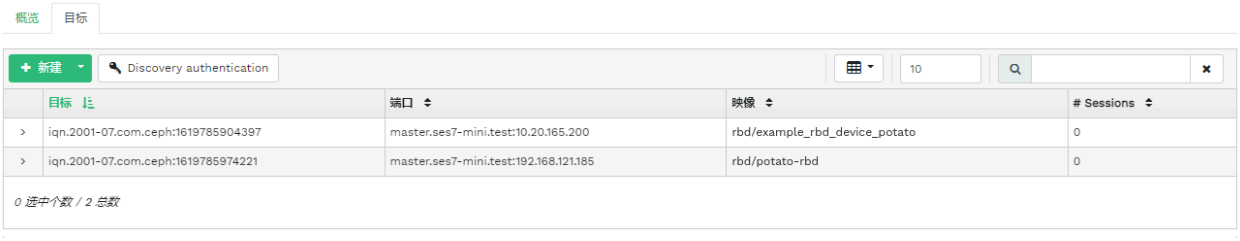


图 6.14： ISCSI 目标列表

要查看有关目标的更多详细信息，请点击表中该目标对应行上的下拉箭头。一个树状结构纲要即会打开，列出磁盘、端口、发起程序和组。点击某一项可将其展开并查看详细内容，右侧的表中也可能会显示其相关配置。

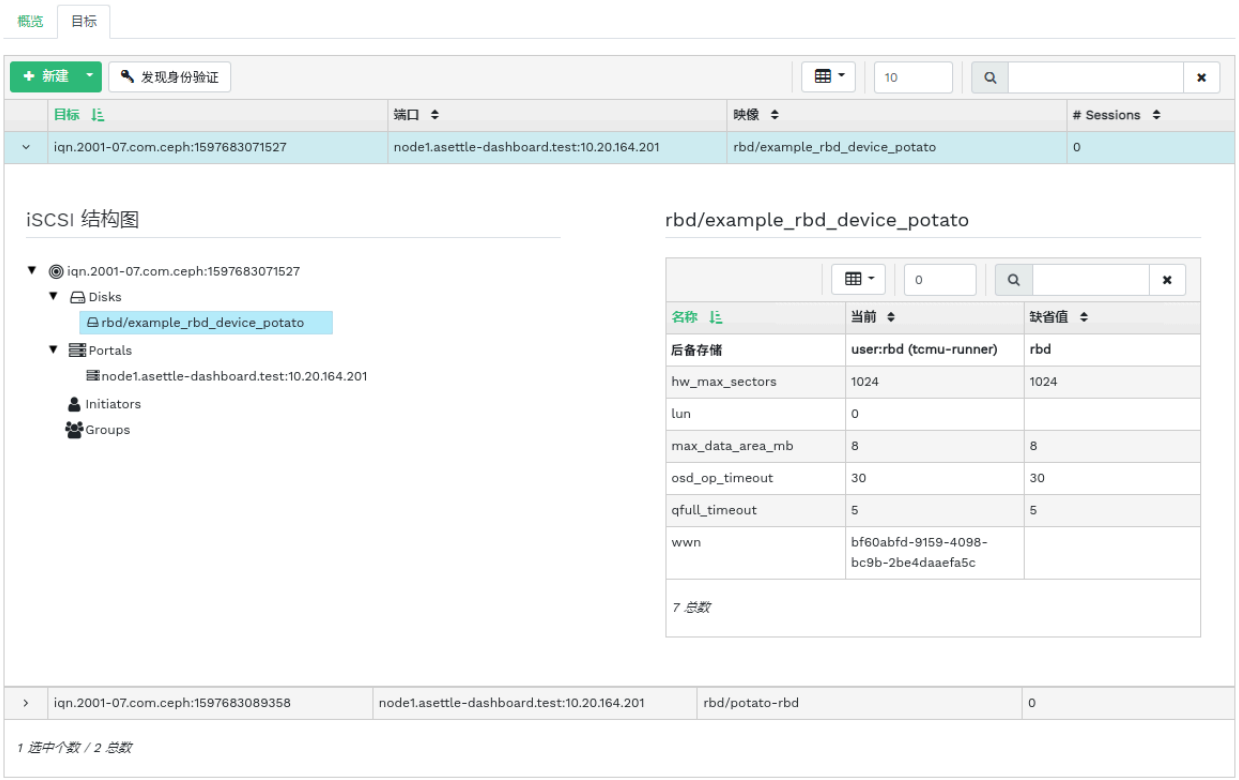


图 6.15： ISCSI 目标详细信息

6.7.1 添加 iSCSI 目标

要添加新的 iSCSI 目标，请点击目标表左上方的创建，并输入所需信息。

新建 Target
目标限定名 *
iqn.2001-07.com.ceph:1620200955217
端口 *
master.ses7-mini.test:10.20.165.200
添加端口
映像
rbd/potato_rbd
lun: 0
Backstore: rbd.
添加映像
ACL 身份验证
用户
密码
互认证用户
互认证密码
新建 Target
取消

图 6.16： 添加新目标

1. 输入新网关的目标地址。
2. 点击添加端口并从列表中选择一个或多个 iSCSI 门户。
3. 点击添加映像并为网关选择一个或多个 RBD 映像。
4. 如果您需要使用身份验证才能访问网关，请激活 ACL 身份验证复选框并输入身份凭证。选中相互身份验证和发现身份验证之后，您可看到更多高级身份验证选项。
5. 点击创建目标进行确认。

6.7.2 编辑 iSCSI 目标

要编辑某个现有 iSCSI 目标，请点击目标表中对应的行，然后点击表左上方的编辑。

然后，您便可以修改 iSCSI 目标、添加或删除端口，以及添加或删除相关 RBD 映像。您还可以调整网关的身份验证信息。

6.7.3 删除 iSCSI 目标

要删除 iSCSI 目标，请选择表中对应的行并点击编辑按钮旁的下拉箭头，然后选择删除。激活是的，我确定，然后点击删除 iSCSI 目标进行确认。

6.8 RBD 服务质量 (QoS)

提示：详细信息

有关 RBD QoS 配置选项的更多一般信息和说明，请参见第 20.6 节 “QoS 设置”。

QoS 选项可在不同级别配置。

- 全局
- 基于每个存储池
- 基于每个映像

全局配置位于列表顶部，将用于所有新创建的 RBD 映像，以及不会覆盖存储池或 RBD 映像层级相应值的映像。全局指定的选项值可能会被基于每个存储池或映像设置的值覆盖。针对某个存储池指定的选项将应用到该存储池的所有 RBD 映像，除非被针对某个映像设置的配置选项所覆盖。对映像指定的选项将覆盖对存储池指定的选项以及全局指定的选项。

因此可以如此操作：全局定义默认值，对默认值进行调整以适合特定存储池的所有 RBD 映像，然后覆盖单个 RBD 映像的存储池配置。

6.8.1 全局配置选项

要全局配置 RADOS 块设备选项，请从主菜单中选择集群 › 配置。

1. 要列出所有可用的全局配置选项，请在级别旁从下拉菜单中选择高级。
2. 在搜索字段中过滤 `rbd_qos`，以过滤表的结果。QoS 的所有可用配置选项即会列出。
3. 要更改某个值，请点击表中对应的行，然后选择表左上方的编辑。编辑对话框包含 6 个不同的字段供您指定值。在 mgr 文本框中，RBD 配置选项值是必填的。

注意

此对话框与其他对话框不同，不允许您使用方便的单位指定值。您只能使用字节或 IOPS 单位设置这些值，具体取决于您要编辑的选项。

6.8.2 针对新存储池配置选项

要创建新存储池并对其配置 RBD 配置选项，请点击存储池 › 创建。选择副本作为存储池类型。然后，您需要为存储池添加 `rbd` 应用标记，这样才能配置 RBD QoS 选项。

注意

对于纠删码存储池是无法配置 RBD QoS 配置选项的。要为纠删码存储池配置 RBD QoS 选项，您需要编辑 RBD 映像的副本元数据池。该配置随后将应用于该映像的纠删码数据池。

6.8.3 针对现有存储池配置选项

要对某个现有存储池配置 RBD QoS 选项，请点击存储池，然后点击表中该存储池对应的行，并选择表左上方的编辑。

对话框中应该就会显示 RBD 配置部分，下面是服务质量 (QoS) 部分。

注意

如果 RBD 配置和服务质量 (QoS) 部分均未显示，则可能是因为您正在编辑无法用于设置 RBD 配置选项的**纠删码**池，或该存储池未配置为供 RBD 映像使用。在下一个案例中，将为存储池指定 rbd 应用标记，相应的配置部分将会显示。

6.8.4 配置选项

点击服务质量 + 展开配置选项。所有可用选项的列表即会显示。文本框中已显示配置选项的单位。对于任何每秒字节数 (BPS) 选项，您可以选择使用“1M”或“5G”之类的缩写。系统会自动将其转换为对应的“1 MB/s”和“5 GB/s”。

点击每个文本框右侧的重设置按钮，将删除对该存储池设置的所有值。此操作不会删除全局配置或对 RBD 映像配置的选项的配置值。

6.8.5 使用新 RBD 映像创建 RBD QoS 选项

要创建 RBD 映像并对该映像设置 RBD QoS 选项，请选择设备块 › 映像，然后点击创建。点击高级... 可展开高级配置部分。点击服务质量 + 可打开所有可用配置选项。

6.8.6 编辑现有映像的 RBD QoS 选项

要编辑现有映像的 RBD QoS 选项，请选择块设备 › 映像，然后点击表中该存储池对应的行，最后点击编辑。编辑对话框随即显示。点击高级... 可展开高级配置部分。点击服务质量 + 可打开所有可用配置选项。

6.8.7 在复制或克隆映像时更改配置选项

如果克隆或复制了 RBD 映像，对该特定映像设置的值默认也将被复制。如果您想在复制或克隆期间更改这些值，可通过在复制/克隆对话框中指定更新的配置值来进行更改，此过程与创建或编辑 RBD 映像时的方式相同。执行此操作只能设置（或重设置）所复制或克隆的 RBD 映像的值，无法更改来源 RBD 映像配置，也无法更改全局配置。

如果您选择在复制/克隆时重设置选项值，系统将不会对该映像设置该选项的值。这意味着系统将使用为父存储池指定的该选项的任何值（如果为父存储池配置了相应值）。如果没有为父存储池配置相应值，则将使用全局默认值。

7 管理 NFS Ganesha

重要

NFS Ganesha 支持 NFS 4.1 和更高版本，不支持 NFS 3 版本。

提示：NFS Ganesha 的更多信息

有关 NFS Ganesha 的更多一般信息，请参见第 25 章 “NFS Ganesha”。

要列出所有可用的 NFS 导出项，请点击主菜单中的 NFS。

该列表会显示每个导出项的目录、守护进程主机名、存储后端类型以及访问类型。

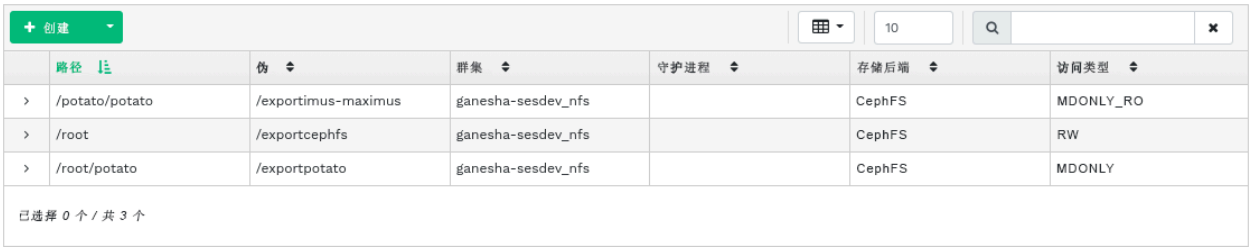


图 7.1： NFS 导出项列表

要查看某个 NFS 导出的更详细的信息，请点击表中对应的行。

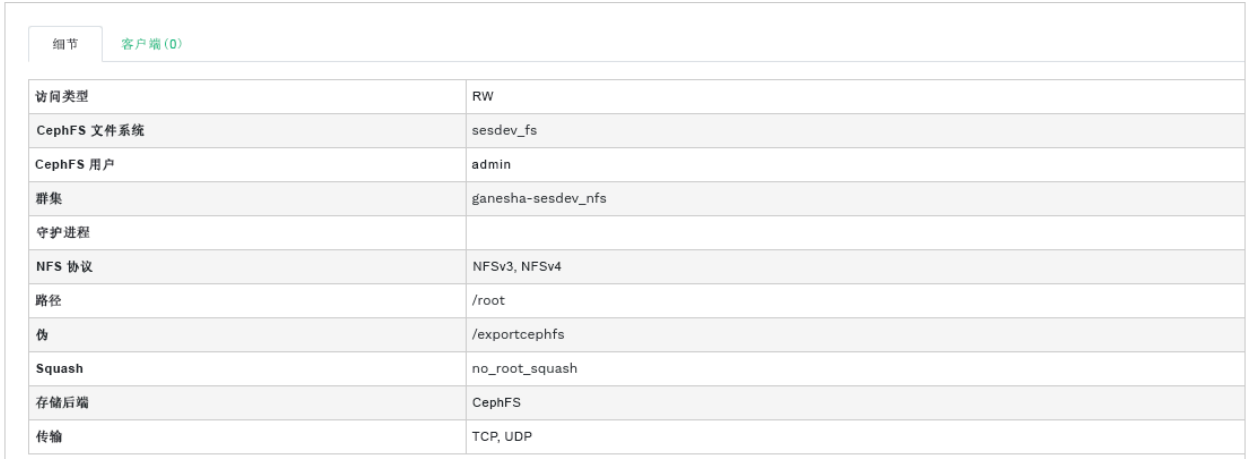


图 7.2： NFS 导出项详细信息

7.1 创建 NFS 导出项

要添加新的 NFS 导出项，请点击导出项表格左上方的创建并输入所需的信息。

图 7.3： 添加新的 NFS 导出项

1. 选择一个或多个将运行导出项的 NFS Ganesha 守护进程。
2. 选择存储后端。

   重要

   目前，仅支持由 CephFS 提供支持的 NFS 导出项。

3. 选择用户 ID 及其他后端相关选项。
4. 输入 NFS 导出项的目录路径。如果该目录在服务器上不存在，系统将创建该目录。
5. 指定其他 NFS 相关选项，例如支持的 NFS 协议版本、伪、访问类型、匿名访问或传输协议。
6. 如果您需要设置限制，仅允许特定的客户端访问，请点击添加客户端并添加它们的 IP 地址以及访问类型和匿名访问选项。
7. 点击创建 NFS 导出项进行确认。

7.2 删除 NFS 导出项

要删除导出项，请选择并高亮显示表中该导出项对应的行。点击编辑按钮旁的下拉箭头，然后选择删除。激活是的，我确定复选框，然后点击删除 NFS 导出项进行确认。

7.3 编辑 NFS 导出项

要编辑现有导出项，请选择并高亮显示表中该导出项对应的行，然后点击导出项表格左上方的编辑。

然后，您便可以调整 NFS 导出项的所有详细信息。

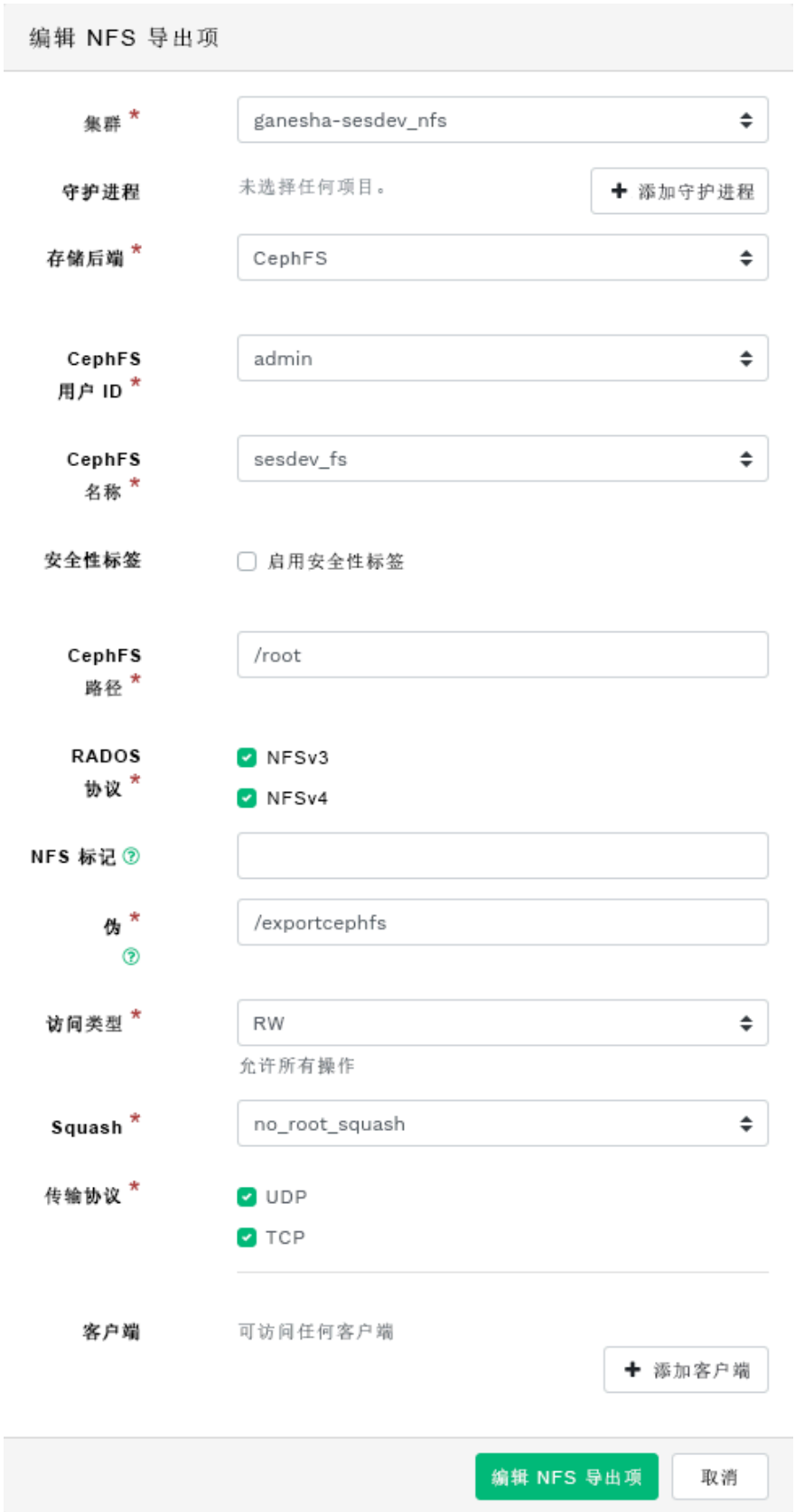


图 7.4： 编辑 NFS 导出项

8 管理 CephFS

提示：详细信息

要查看有关 CephFS 的详细信息，请参见第 23 章 “集群化文件系统”。

8.1 查看 CephFS 概览

在主菜单中点击文件系统可查看配置的文件系统的概览。主表会显示每个文件系统的名称、创建日期，以及文件系统是否启用。

点击表中某个文件系统对应的行，可查看其排位以及添加到其中的存储池的详细信息。

图 8.1： CEPHFS 详细信息

您可在屏幕底部查看实时收集的相关 MDS Inode 及客户端请求数量统计数字。

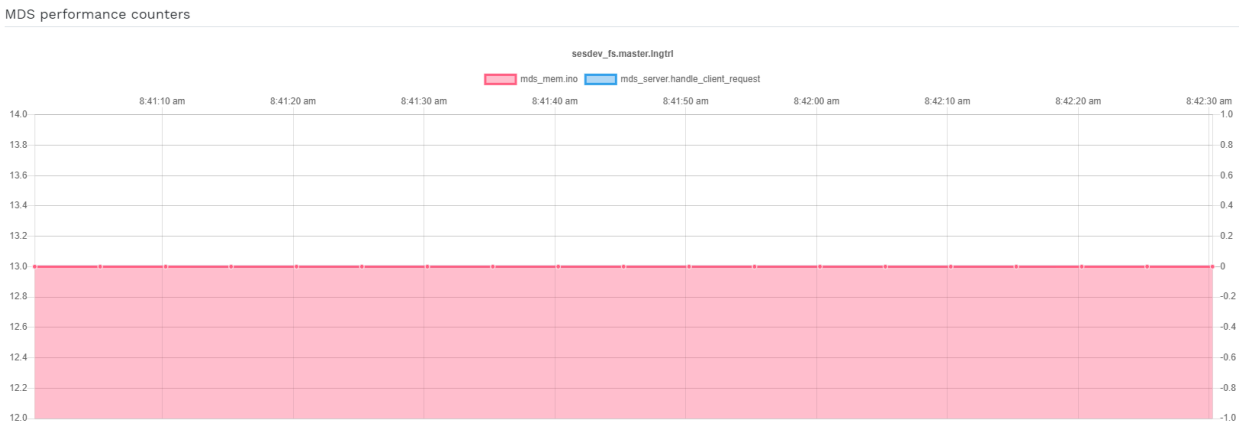


图 8.2： CEPHFS 详细信息

9 管理对象网关

重要

开始之前，在尝试访问 Ceph Dashboard 上的对象网关前端时，您可能会收到以下通知：

```
Information
No RGW credentials found, please consult the documentation on how to
 enable RGW for the dashboard.
Please consult the documentation on how to configure and enable the Object
 Gateway management functionality.
```

这是因为 cephadm 没有为 Ceph Dashboard 自动配置对象网关。如果您收到此通知，请按照第 10.4 节 “启用对象网关管理前端”中的说明手动启用 Ceph Dashboard 的对象网关前端。

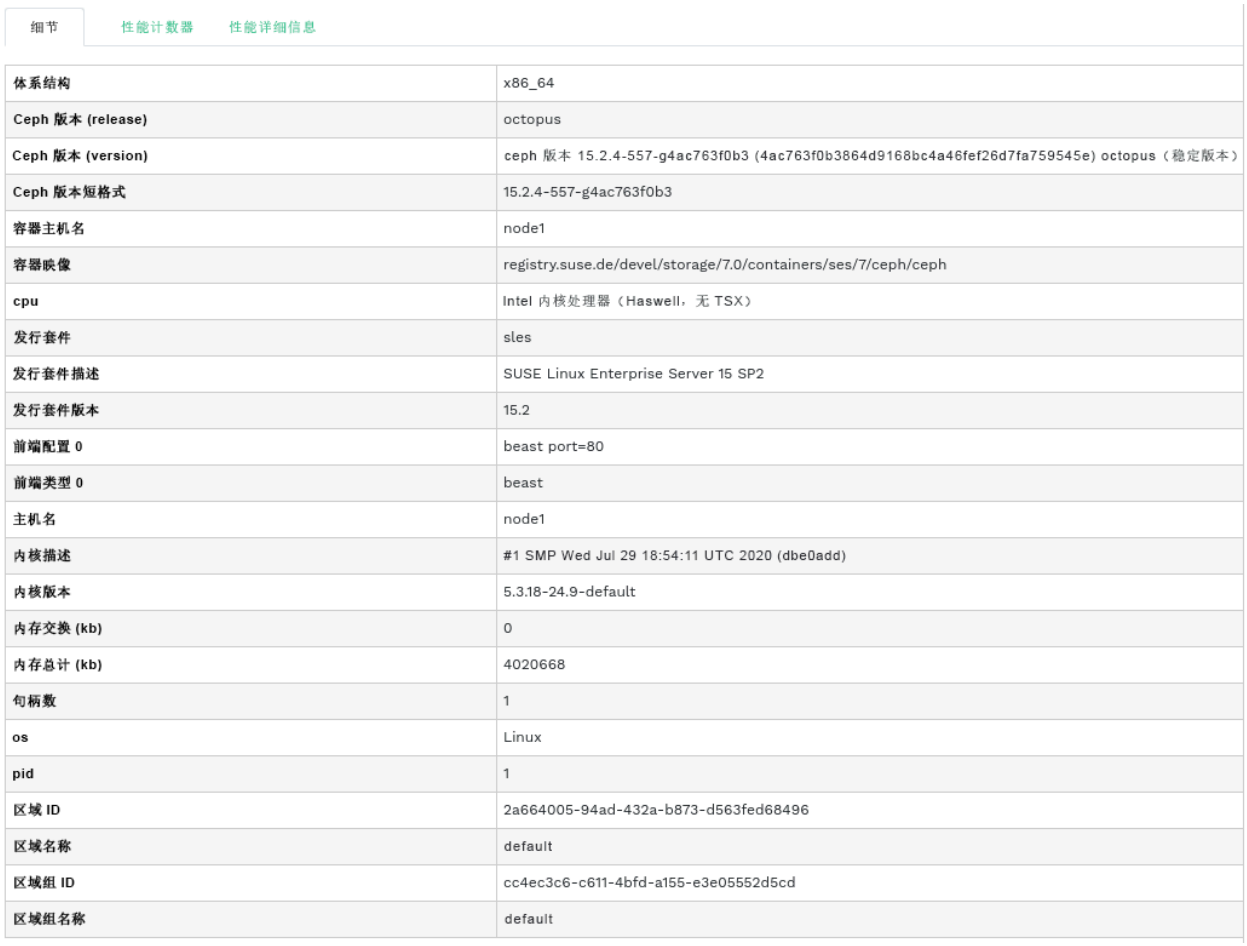
提示：对象网关的更多信息

有关对象网关的更多一般信息，请参见第 21 章 “Ceph 对象网关”。

9.1 查看对象网关

要查看配置的对象网关列表，请点击对象网关 › 守护进程。该列表包含网关的 ID、运行网关守护进程的集群节点的主机名以及网关的版本号。

点击网关名称旁的下拉箭头可查看有关该网关的详细信息。性能计数器选项卡显示有关读/写操作及缓存统计数据的详细信息。



细节　性能计数器　性能详细信息

体系结构	x86_64
Ceph 版本 (release)	octopus
Ceph 版本 (version)	ceph 版本 15.2.4-557-g4ac763f0b3 (4ac763f0b3864d9168bc4a46fef26d7fa759545e) octopus（稳定版本）
Ceph 版本短格式	15.2.4-557-g4ac763f0b3
容器主机名	node1
容器映像	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph
cpu	Intel 内核处理器（Haswell，无 TSX）
发行套件	sles
发行套件描述	SUSE Linux Enterprise Server 15 SP2
发行套件版本	15.2
前端配置 0	beast port=80
前端类型 0	beast
主机名	node1
内核描述	#1 SMP Wed Jul 29 18:54:11 UTC 2020 (dbe0add)
内核版本	5.3.18-24.9-default
内存交换 (kb)	0
内存总计 (kb)	4020668
句柄数	1
os	Linux
pid	1
区域 ID	2a664005-94ad-432a-b873-d563fed68496
区域名称	default
区域组 ID	cc4ec3c6-c611-4bfd-a155-e3e05552d5cd
区域组名称	default

图 9.1： 网关的详细信息

9.2 管理对象网关用户

点击对象网关 › 用户可查看现有对象网关用户的列表。

点击用户名旁的下拉箭头可查看有关用户帐户的详细信息，例如状态信息或用户和存储桶配额详细信息。

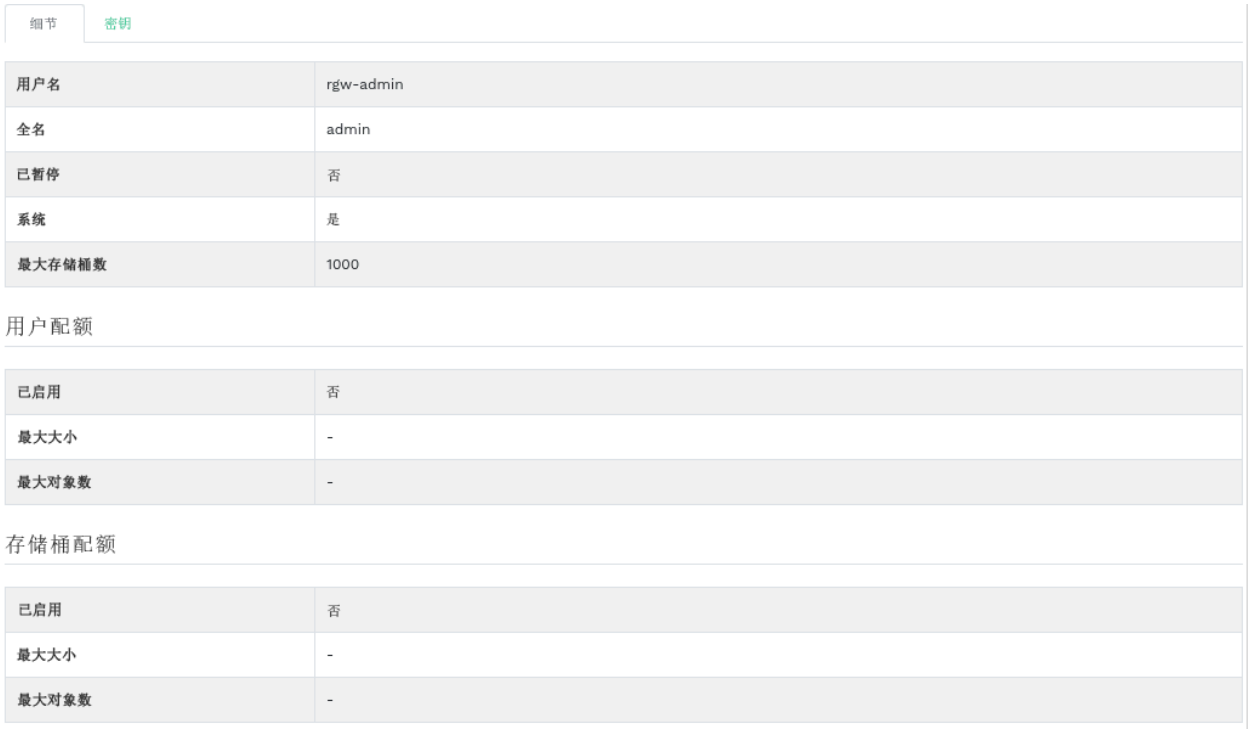


图 9.2： 网关用户

9.2.1 添加新网关用户

要添加新网关用户，请点击表标题左上方的创建。填写其身份凭证、有关 S3 密钥及用户和存储桶配额的详细信息，然后点击创建用户进行确认。

创建用户
用户名 *
example_rgw_user
全名 *
Example User
电子邮件地址
example@user.com
最大存储桶数
自定义
1000
已暂停
S3 密钥
自动生成密钥
用户配额
已启用
存储桶配额
已启用
大小不限
对象数不限
创建用户
取消

图 9.3： 添加新网关用户

9.2.2 删除网关用户

要删除网关用户，请选择并高亮显示该用户。点击编辑旁的下拉按钮，然后从列表中选择删除，即可删除用户帐户。激活是的，我确定复选框，然后点击删除用户进行确认。

9.2.3 编辑网关用户详细信息

要更改网关用户详细信息，请选择并高亮显示该用户。点击表标题左上方的编辑。

修改基本或附加用户信息，例如其用户权限、密钥、子用户和配额信息。点击编辑用户进行确认。

密钥选项卡包含只读的网关用户列表以及这些用户的访问密钥和秘密密钥。要查看这些密钥，请点击列表中的用户名，然后选择表标题左上方的显示。在 S3 密钥对话框中，点击眼睛图标可显示密钥，点击剪贴板图标可将相关密钥复制到剪贴板。

9.3 管理对象网关存储桶

对象网关 (OGW) 存储桶实现了 OpenStack Swift 容器的功能。对象网关存储桶可充当存储数据对象的容器。

点击对象网关 › 存储桶可查看对象网关存储桶列表。

9.3.1 添加新的存储桶

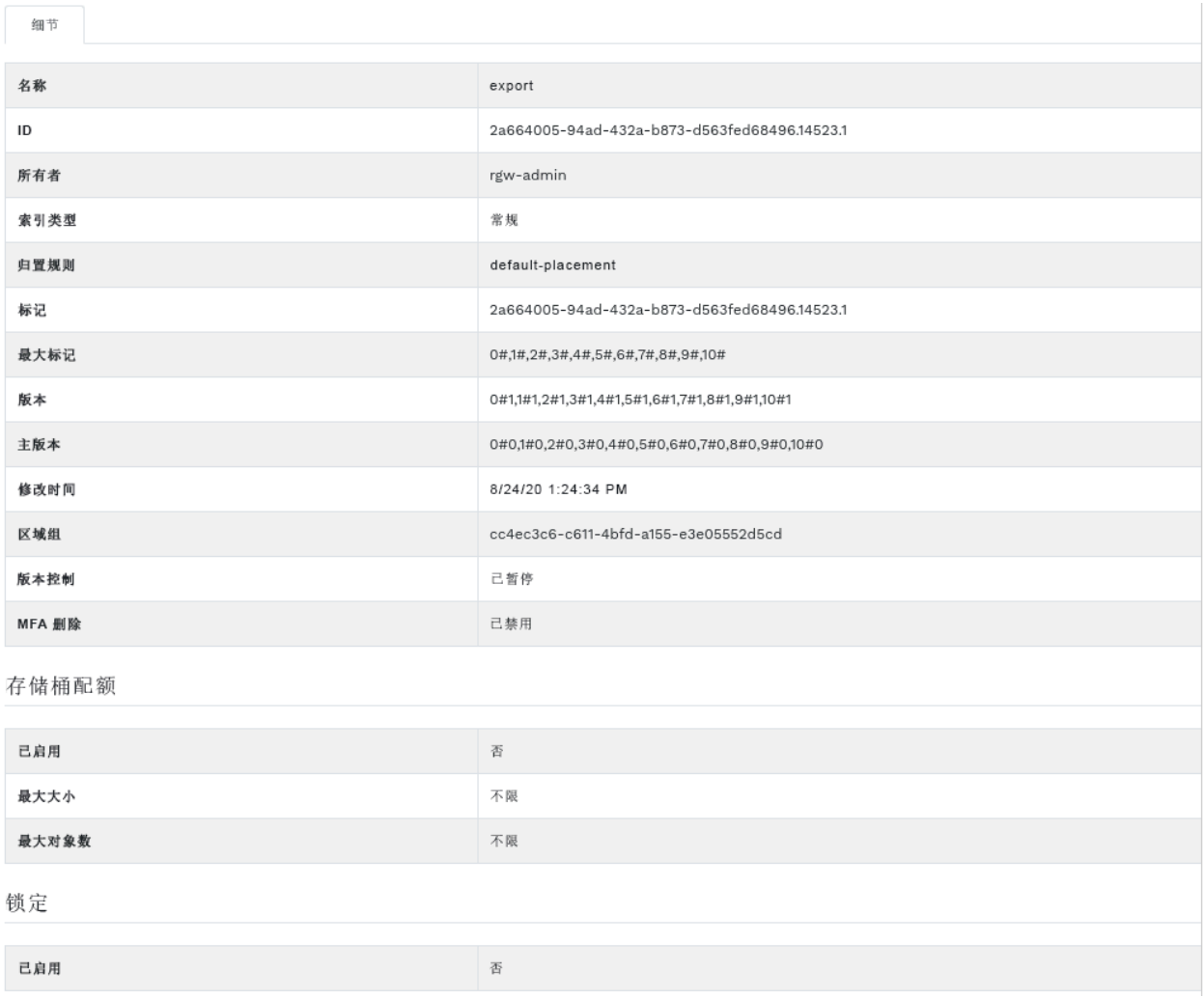
要添加新的对象网关存储桶，请点击表标题左上方的创建。输入存储桶的名称，选择所有者，并设置归置目标。点击创建存储桶进行确认。

注意

您也可以在此阶段通过选择已启用来启用锁定；但此功能在存储桶创建之后亦可配置。有关更多信息，请参见第 9.3.3 节 “编辑存储桶”。

9.3.2 查看存储桶详细信息

要查看有关对象网关存储桶的详细信息，请点击存储桶名称旁的下拉箭头。



细节

名称	export
ID	2a664005-94ad-432a-b873-d563fed68496.14523.1
所有者	rgw-admin
索引类型	常规
归置规则	default-placement
标记	2a664005-94ad-432a-b873-d563fed68496.14523.1
最大标记	0#,1#,2#,3#,4#,5#,6#,7#,8#,9#,10#
版本	0#1,1#1,2#1,3#1,4#1,5#1,6#1,7#1,8#1,9#1,10#1
主版本	0#0,1#0,2#0,3#0,4#0,5#0,6#0,7#0,8#0,9#0,10#0
修改时间	8/24/20 1:24:34 PM
区域组	cc4ec3c6-c611-4bfd-a155-e3e05552d5cd
版本控制	已暂停
MFA 删除	已禁用

存储桶配额

已启用	否
最大大小	不限
最大对象数	不限

锁定

已启用	否

图 9.4： 网关存储桶详细信息

提示：存储桶配额

您可以在详细信息表下方查看有关存储桶配额和锁定设置的详细信息。

9.3.3 编辑存储桶

选择并高亮显示存储桶，然后点击表标题左上方的编辑。

您可以更新存储桶的所有者，或者启用版本控制、多重身份验证或锁定功能。点击编辑存储桶确认任何更改。

图 9.5： 编辑存储桶详细信息

9.3.4 删除存储桶

要删除对象网关存储桶，请选择并高亮显示该存储桶。点击编辑旁的下拉按钮，然后从列表中选择删除，即可删除存储桶。激活是的，我确定复选框，然后点击删除存储桶进行确认。

10 手动配置

本节为更喜欢在命令行上手动配置仪表盘设置的用户提供了较深入的信息。

10.1 配置 TLS/SSL 支持

所有连至仪表盘的 HTTP 连接默认都使用 TLS/SSL 来保障安全。安全连接需要 SSL 证书。您可以使用自我签名证书，也可以生成证书并让知名证书颁发机构 (CA) 对其签名。

提示：禁用 SSL

您可能会出于某种原因需要禁用 SSL 支持。例如，如果仪表盘是在不支持 SSL 的代理之后运行。

禁用 SSL 要慎重，因为禁用 SSL 后，**用户名和密码**将以**未加密**的形式发送到仪表盘。

要禁用 SSL，请运行以下命令：

```
cephuser@adm > ceph config set mgr mgr/dashboard/ssl false
```

提示：重启动 Ceph Manager 进程

更改 SSL 证书和密钥后，您需要手动重启动 Ceph Manager 进程。您可以通过运行以下命令

```
cephuser@adm > ceph mgr fail ACTIVE-MANAGER-NAME
```

或禁用后再重新启用仪表盘扩展模块的方式来执行此操作，如此还会触发 Manager 的自我重新生成：

```
cephuser@adm > ceph mgr module disable dashboard
cephuser@adm > ceph mgr module enable dashboard
```

10.1.1 创建自我签名证书

为了安全通讯而创建自我签名证书的过程非常简单。采用这种方式可迅速让仪表盘运行起来。

注意：Web 浏览器控诉

大多数 Web 浏览器都将对自我签名证书发起控诉，并需要在建立与仪表盘的安全连接前进行明确确认。

要生成并安装自我签名证书，请使用以下内置命令：

```
cephuser@adm > ceph dashboard create-self-signed-cert
```

10.1.2 使用 CA 签名的证书

为了正确保障连至仪表盘的连接安全，以及消除 Web 浏览器对自我签名证书的控诉，我们建议使用 CA 签名的证书。

您可以使用如下命令生成证书密钥对：

```
# openssl req -new -nodes -x509 \
  -subj "/O=IT/CN=ceph-mgr-dashboard" -days 3650 \
  -keyout dashboard.key -out dashboard.crt -extensions v3_ca
```

以上命令会输出 `dashboard.key` 和 `dashboard.crt` 文件。获得 CA 签名的 `dashboard.crt` 文件后，请运行以下命令为所有 Ceph Manager 实例启用该证书：

```
cephuser@adm > ceph dashboard set-ssl-certificate -i dashboard.crt
cephuser@adm > ceph dashboard set-ssl-certificate-key -i dashboard.key
```

提示：每个 Manager 实例使用不同的证书

如果您需要为每个 Ceph Manager 实例使用不同的证书，请按如下所示修改命令，在其中指定实例的名称。将 `NAME` 替换为 Ceph Manager 实例的名称（通常是相关的主机名）：

```
cephuser@adm > ceph dashboard set-ssl-certificate NAME -i dashboard.crt
cephuser@adm > ceph dashboard set-ssl-certificate-key NAME -i
 dashboard.key
```

10.2 更改主机名和端口号

Ceph Dashboard 与特定 TCP/IP 地址和 TCP 端口绑定。默认情况下，托管仪表盘的当前活跃 Ceph Manager 会与 TCP 端口 8443 绑定（如果禁用 SSL，则与 8080 端口绑定）。

注意

如果在运行 Ceph Manager（以及 Ceph Dashboard）的主机上启用了防火墙，则您可能需要更改配置以启用对这些端口的访问。有关 Ceph 防火墙设置的详细信息，请参见《Troubleshooting Guide》, 第 13 章 “Hints and tips”, 第 13.7 节 “Firewall settings for Ceph”。

Ceph Dashboard 默认与“::”绑定，该符号表示所有可用的 IPv4 和 IPv6 地址。您可以使用以下命令更改 Web 应用的 IP 地址和端口号，以使它们适用于所有 Ceph Manager 实例：

```
cephuser@adm > ceph config set mgr mgr/dashboard/server_addr IP_ADDRESS
cephuser@adm > ceph config set mgr mgr/dashboard/server_port PORT_NUMBER
```

提示：分别配置 Ceph Manager 实例

因为每个 `ceph-mgr` 守护进程都托管着各自的仪表盘实例，您可能需要分别配置每个 Ceph Manager 实例。使用以下命令（将 `NAME` 替换为 `ceph-mgr` 实例的 ID）更改特定 Manager 实例的 IP 地址和端口号：

```
cephuser@adm > ceph config set mgr mgr/dashboard/NAME/
server_addr IP_ADDRESS
cephuser@adm > ceph config set mgr mgr/dashboard/NAME/
server_port PORT_NUMBER
```

提示：列出配置的端点

`ceph mgr services` 命令会显示当前配置的所有端点。查找 `dashboard` 键可获取用于访问仪表盘的 URL。

10.3 调整用户名和密码

如果您不想使用默认管理员帐户，可创建其他用户帐户并将其与至少一个角色相关联。我们提供了一组预定义的系统角色供您使用。有关详细信息，请参见第 11 章 “在命令行上管理用户和角色”。

要创建具有管理员权限的用户，请使用以下命令：

```
cephuser@adm > ceph dashboard ac-user-create USER_NAME PASSWORD administrator
```

10.4 启用对象网关管理前端

要使用仪表盘的对象网关管理功能，您需要提供启用了 `system` 标志的用户的登录身份凭证：

1. 如果您没有带 `system` 标志的用户，请创建一个：

   ```
   cephuser@adm > radosgw-admin user create --uid=USER_ID --display-
   name=DISPLAY_NAME --system
   ```

   记下命令输出中的 `access_key` 和 `secret_key` 密钥。

2. 您还可以使用 **`radosgw-admin`** 命令来获取现有用户的身份凭证：

   ```
   cephuser@adm > radosgw-admin user info --uid=USER_ID
   ```

3. 在单独的文件中向仪表盘提供收到的身份凭证：

   ```
   cephuser@adm > ceph dashboard set-rgw-api-access-key ACCESS_KEY_FILE
   cephuser@adm > ceph dashboard set-rgw-api-secret-key SECRET_KEY_FILE
   ```

注意

默认情况下，SUSE Linux Enterprise Server 15 SP3 中会启用防火墙。有关防火墙配置的详细信息，请参见《Troubleshooting Guide》, 第 13 章 “Hints and tips”, 第 13.7 节 “Firewall settings for Ceph”。

请注意以下几点事项：

- 对象网关的主机名和端口号是自动确定的。
- 如果使用多个区域，对象网关会自动确定主区域组和主区域内的主机。此方式足以满足大部分设置的需求，但在某些情况下，您可能需要手动设置主机名和端口：

```
cephuser@adm > ceph dashboard set-rgw-api-host HOST
cephuser@adm > ceph dashboard set-rgw-api-port PORT
```

- 下面是您可能需要进行的额外设置：

```
cephuser@adm > ceph dashboard set-rgw-api-scheme SCHEME  # http or https
cephuser@adm > ceph dashboard set-rgw-api-admin-resource ADMIN_RESOURCE
cephuser@adm > ceph dashboard set-rgw-api-user-id USER_ID
```

- 如果您在对象网关设置中使用的是自我签名证书（第 10.1 节 “配置 TLS/SSL 支持”），请在仪表盘中禁用证书验证，以免发生因证书是由未知 CA 签名或与主机名不符而导致连接被拒。

```
cephuser@adm > ceph dashboard set-rgw-api-ssl-verify False
```

- 如果对象网关处理请求所花的时间过长，导致仪表盘超时，则可调整超时值（默认为 45 秒）：

```
cephuser@adm > ceph dashboard set-rest-requests-timeout SECONDS
```

10.5 启用 iSCSI 管理

Ceph Dashboard 使用 Ceph iSCSI 网关的 `rbd-target-api` 服务所提供的 REST API 管理 iSCSI 目标。确保已在 iSCSI 网关上安装并启用它。

注意

Ceph Dashboard 的 iSCSI 管理功能取决于 `ceph-iscsi` 项目的最新版本 3。请确保您的操作系统提供了正确的版本，否则 Ceph Dashboard 将无法启用管理功能。

如果 `ceph-iscsi` REST API 以 HTTPS 模式配置并使用自我签名证书，请配置仪表盘以避免在访问 ceph-iscsi API 时发生 SSL 证书验证。

禁用 API SSL 验证：

```
cephuser@adm > ceph dashboard set-iscsi-api-ssl-verification false
```

定义可用的 iSCSI 网关：

```
cephuser@adm > ceph dashboard iscsi-gateway-list
cephuser@adm > ceph dashboard iscsi-gateway-
add scheme://username:password@host[:port]
cephuser@adm > ceph dashboard iscsi-gateway-rm gateway_name
```

10.6 启用单点登录

单点登录 (SSO) 是一种允许用户以单个 ID 和密码同时登录多个应用的访问控制方法。

Ceph Dashboard 支持通过 SAML 2.0 协议对用户进行外部身份验证。由于**授权**仍由仪表盘执行，因此您需要先创建用户帐户并将其与所需角色相关联。不过，**身份验证**过程可由现有**身份提供者** (IdP) 执行。

要配置单点登录，请使用以下命令：

```
cephuser@adm > ceph dashboard sso setup saml2 CEPH_DASHBOARD_BASE_URL \
 IDP_METADATA IDP_USERNAME_ATTRIBUTE \
 IDP_ENTITY_ID SP_X_509_CERT \
 SP_PRIVATE_KEY
```

参数：

`CEPH_DASHBOARD_BASE_URL`

可用于访问 Ceph Dashboard 的基本 URL（例如“https://cephdashboard.local”）。

`IDP_METADATA`

IdP 元数据 XML 的 URL、文件路径或内容（例如“https://myidp/metadata”）。

`IDP_USERNAME_ATTRIBUTE`

可选。将用于从身份验证响应中获取用户名的属性。默认为“uid”。

`IDP_ENTITY_ID`

可选。当 IdP 元数据上存在多个实体 ID 时，请使用该参数。

`SP_X_509_CERT` / `SP_PRIVATE_KEY`

可选。Ceph Dashboard（服务提供程序）进行签名和加密时将使用的证书的文件路径或内容。活动的 Ceph Manager 实例需要能访问这些文件路径。

注意：SAML 请求

SAML 请求的发出者值将采用以下格式：

```
CEPH_DASHBOARD_BASE_URL/auth/saml2/metadata
```

要显示当前 SAML 2.0 配置，请运行以下命令：

```
cephuser@adm > ceph dashboard sso show saml2
```

要禁用单点登录，请运行以下命令：

```
cephuser@adm > ceph dashboard sso disable
```

要检查是否启用了 SSO，请运行以下命令：

```
cephuser@adm > ceph dashboard sso status
```

要启用 SSO，请运行以下命令：

```
cephuser@adm > ceph dashboard sso enable saml2
```

11 在命令行上管理用户和角色

本节说明如何管理 Ceph Dashboard 使用的用户帐户。该工具可帮助您创建或修改用户帐户，以及设置正确的用户角色和权限。

11.1 管理密码策略

默认情况下会启用密码策略功能，包括以下检查：

- 密码长度是否超过 **N** 个字符？
- 旧密码和新密码是否相同？

可以完全打开或关闭密码策略功能：

```
cephuser@adm > ceph dashboard set-pwd-policy-enabled true|false
```

可以打开或关闭以下各项检查：

```
cephuser@adm > ceph dashboard set-pwd-policy-check-length-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-oldpwd-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-username-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-exclusion-list-enabled true|
false
cephuser@adm > ceph dashboard set-pwd-policy-check-complexity-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-sequential-chars-
enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-repetitive-chars-
enabled true|false
```

此外，可使用以下选项配置密码策略行为。

- 最短密码长度（默认值为 8）：

  ```
  cephuser@adm > ceph dashboard set-pwd-policy-min-length N
  ```

- 最小密码复杂性（默认值为 10）：

```
cephuser@adm > ceph dashboard set-pwd-policy-min-complexity N
```

密码复杂性是通过对密码中的每个字符进行分类来计算的。

- 密码中不允许使用的逗号分隔的单词列表：

```
cephuser@adm > ceph dashboard set-pwd-policy-exclusion-list word[,...]
```

11.2 管理用户帐户

Ceph Dashboard 支持管理多个用户帐户。每个用户帐户包含用户名、密码（使用 `bcrypt` 以加密形式存储）、可选名称和可选电子邮件地址。

用户帐户存储在 Ceph Monitor 的配置数据库中，并在所有 Ceph Manager 实例之间全局共享。

使用以下命令可管理用户帐户：

显示现有用户：

```
cephuser@adm > ceph dashboard ac-user-show [USERNAME]
```

创建新用户：

```
cephuser@adm > ceph dashboard ac-user-create USERNAME -i [PASSWORD_FILE]
 [ROLENAME] [NAME] [EMAIL]
```

删除用户：

```
cephuser@adm > ceph dashboard ac-user-delete USERNAME
```

更改用户密码：

```
cephuser@adm > ceph dashboard ac-user-set-password USERNAME -
i PASSWORD_FILE
```

修改用户名和电子邮件：

```
cephuser@adm > ceph dashboard ac-user-set-info USERNAME NAME EMAIL
```

禁用用户

```
cephuser@adm > ceph dashboard ac-user-disable USERNAME
```

启用用户

```
cephuser@adm > ceph dashboard ac-user-enable USERNAME
```

11.3 用户角色和权限

本节说明您可为用户角色指定的安全作用范围、如何管理用户角色，以及如何为用户帐户指定角色。

11.3.1 定义安全作用范围

用户帐户与一组角色相关联，这些角色定义了该用户可以访问仪表盘的哪些部分。仪表盘的各个部分均会划分在某个**安全**作用范围内。安全作用范围已预先定义且固定不变。当前可用的安全作用范围如下：

hosts

包含与主机菜单项相关的所有特性。

config-opt

包含与 Ceph 配置选项管理相关的所有特性。

pool

包含与存储池管理相关的所有特性。

osd

包含与 Ceph OSD 管理相关的所有特性。

monitor

包含与 Ceph Monitor 管理相关的所有特性。

rbd-image

包含与 RADOS 块设备映像管理相关的所有特性。

rbd-mirroring

包含与 RADOS 块设备镜像管理相关的所有特性。

iscsi

包含与 iSCSI 管理相关的所有特性。

rgw

包含与对象网关管理相关的所有特性。

cephfs

包含与 CephFS 管理相关的所有特性。

manager

包含与 Ceph Manager 管理相关的所有特性。

log

包含与 Ceph 日志管理相关的所有特性。

grafana

包含与 Grafana 代理相关的所有特性。

prometheus

包含与 Prometheus 告警管理相关的所有特性。

dashboard-settings

允许更改仪表盘设置。

11.3.2 指定用户角色

角色指定了一个**安全作用范围**与一组**权限**之间的一组映射。权限分为四种类型：“read”、“create”、“update”和“delete”。

下面的示例指定了一个角色，具有该角色的用户拥有与存储池管理相关的特性的“read”和“create”权限，以及与 RBD 映像管理相关的特性的全部权限：

```
{
  'role': 'my_new_role',
  'description': 'My new role',
```

```
  'scopes_permissions': {
    'pool': ['read', 'create'],
    'rbd-image': ['read', 'create', 'update', 'delete']
  }
}
```

仪表盘提供了一组预定义角色，我们称之为**系统角色**。全新安装 Ceph Dashboard 后即可使用这些角色：

administrator
: 提供针对所有安全作用范围的全部权限。

read-only
: 提供针对仪表盘设置以外的其他所有安全作用范围的读取权限。

block-manager
: 提供针对“rbd-image”、“rbd-mirroring”和“iscsi”作用范围的全部权限。

rgw-manager
: 提供针对“rgw”作用范围的全部权限。

cluster-manager
: 提供针对“hosts”、“osd”、“monitor”、“manager”和“config-opt”作用范围的全部权限。

pool-manager
: 提供针对“pool”作用范围的全部权限。

cephfs-manager
: 提供针对“cephfs”作用范围的全部权限。

11.3.2.1 管理自定义角色

您可以使用以下命令创建新用户角色：

创建新角色：

```
cephuser@adm > ceph dashboard ac-role-create ROLENAME [DESCRIPTION]
```

删除角色：

```
cephuser@adm > ceph dashboard ac-role-delete ROLENAME
```

为角色添加作用范围权限：

```
cephuser@adm > ceph dashboard ac-role-add-scope-
perms ROLENAME SCOPENAME PERMISSION [PERMISSION...]
```

从角色中删除作用范围权限：

```
cephuser@adm > ceph dashboard ac-role-del-perms ROLENAME SCOPENAME
```

11.3.2.2 为用户帐户指定角色

使用以下命令可为用户指定角色：

设置用户角色：

```
cephuser@adm > ceph dashboard ac-user-set-roles USERNAME ROLENAME [ROLENAME
 ...]
```

为用户添加其他角色：

```
cephuser@adm > ceph dashboard ac-user-add-roles USERNAME ROLENAME [ROLENAME
 ...]
```

删除用户的角色：

```
cephuser@adm > ceph dashboard ac-user-del-roles USERNAME ROLENAME [ROLENAME
 ...]
```

提示：清除自定义角色

如果您创建了自定义用户角色，并打算以后使用 **`ceph.purge`** 运行程序删除 Ceph 集群，则需要先清除自定义角色。有关详细信息，请参见第 13.9 节 “删除整个 Ceph 集群”。

11.3.2.3 示例：创建用户和自定义角色

本节说明创建具有以下权限的用户帐户的过程：可管理 RBD 映像、查看和创建 Ceph 存储池，并拥有针对任何其他安全作用范围的只读访问权限。

1. 创建名为 `tux` 的新用户：

```
cephuser@adm > ceph dashboard ac-user-create tux PASSWORD
```

2. 创建角色并指定作用范围权限：

```
cephuser@adm > ceph dashboard ac-role-create rbd/pool-manager
cephuser@adm > ceph dashboard ac-role-add-scope-perms rbd/pool-manager \
 rbd-image read create update delete
cephuser@adm > ceph dashboard ac-role-add-scope-perms rbd/pool-manager pool
 read create
```

3. 将角色与 `tux` 用户相关联：

```
cephuser@adm > ceph dashboard ac-user-set-roles tux rbd/pool-manager read-
only
```

11.4 代理配置

如果您要建立固定 URL 以访问 Ceph Dashboard，或者不允许直接连接到 Manager 节点，可以设置一个代理将传入请求转发到当前活动的 `ceph-mgr` 实例。

11.4.1 使用反向代理访问仪表盘

如果您要通过反向代理配置访问仪表盘，则可能需要使用 URL 前缀来访问。要让仪表盘使用包含您前缀的超链接，您可以设置 `url_prefix` 设置：

```
cephuser@adm > ceph config set mgr mgr/dashboard/url_prefix URL_PREFIX
```

然后，您便可通过网址 `http://HOST_NAME:PORT_NUMBER/URL_PREFIX/` 来访问仪表盘。

11.4.2 禁用重定向

如果 Ceph Dashboard 位于 HAProxy 等负载平衡代理之后，请禁用重定向行为，以防止出现将内部（无法解析的）URL 发布到前端客户端的情况。使用以下命令可让仪表盘以 HTTP 错误（默认为500）响应，而不是重定向到活动仪表盘：

```
cephuser@adm > ceph config set mgr mgr/dashboard/standby_behaviour "error"
```

要将设置重置为默认的重定向行为，请使用以下命令：

```
cephuser@adm > ceph config set mgr mgr/dashboard/standby_behaviour "redirect"
```

11.4.3 配置错误状态码

如果禁用了重定向行为，则应自定义待机仪表盘的 HTTP 状态码。为此，请运行以下命令：

```
cephuser@adm > ceph config set mgr mgr/dashboard/standby_error_status_code 503
```

11.4.4 HAProxy 示例配置

以下示例配置适用于使用 HAProxy 的 TLS/SSL 传递。

注意

该配置适用于以下情况：如果仪表盘发生故障转移，前端客户端可能会收到 HTTP 重定向 (303) 响应，并将重定向到无法解析的主机。

两次 HAProxy 健康状况检查期间发生故障转移时，会发生此情况。在这种情况下，之前处于活动状态的仪表盘节点现在将以 303 进行响应，从而指向新活动节点。为避免出现该情况，您应该考虑禁用待机节点上的重定向行为。

```
  defaults
    log global
    option log-health-checks
    timeout connect 5s
    timeout client 50s
    timeout server 450s
```

```
frontend dashboard_front
  mode http
  bind *:80
  option httplog
  redirect scheme https code 301 if !{ ssl_fc }

frontend dashboard_front_ssl
  mode tcp
  bind *:443
  option tcplog
  default_backend dashboard_back_ssl

backend dashboard_back_ssl
  mode tcp
  option httpchk GET /
  http-check expect status 200
  server x HOST:PORT ssl check verify none
  server y HOST:PORT ssl check verify none
  server z HOST:PORT ssl check verify none
```

11.5 审计 API 请求

Ceph Dashboard 的 REST API 可在 Ceph 审计日志中记录 PUT、POST 和 DELETE 请求。记录功能默认处于禁用状态，但您可使用以下命令将其启用：

```
cephuser@adm > ceph dashboard set-audit-api-enabled true
```

如果记录功能处于启用状态，将会记录每个请求的以下参数：

from

请求的来源，例如“https://[::1]:44410”。

path

REST API 路径，例如 `/api/auth`。

method

“PUT”、“POST”或“DELETE”。

user

用户的名称（或“None”）。

示例日志项如下所示：

```
2019-02-06 10:33:01.302514 mgr.x [INF] [DASHBOARD] \
 from='https://[::ffff:127.0.0.1]:37022' path='/api/rgw/user/exu' method='PUT' \
 user='admin' params='{"max_buckets": "1000", "display_name": "Example User",
 "uid": "exu", "suspended": "0", "email": "user@example.com"}'
```

提示：禁用记录请求有效负载的功能

请求有效负载（参数及其值的列表）的记录功能默认处于启用状态。您可以按照如下方式禁用该功能：

```
cephuser@adm > ceph dashboard set-audit-api-log-payload false
```

11.6 在 Ceph Dashboard 中配置 NFS Ganesha

Ceph Dashboard 可以管理使用 CephFS 或对象网关作为其后备存储的 NFS Ganesha 导出项。该仪表盘用于管理存储在 CephFS 集群上的 RADOS 对象中的 NFS Ganesha 配置文件。NFS Ganesha 必须将其部分配置存储在 Ceph 集群中。

运行以下命令以配置 NFS Ganesha 配置对象的位置：

```
cephuser@adm > ceph dashboard set-ganesha-clusters-rados-pool-
namespace pool_name[/namespace]
```

现在您便可以使用 Ceph Dashboard 管理 NFS Ganesha 导出项。

11.6.1 配置多个 NFS Ganesha 集群

Ceph Dashboard 支持管理属于不同 NFS Ganesha 集群的 NFS Ganesha 导出项。建议每个 NFS Ganesha 集群将其配置对象存储在不同的 RADOS 存储池/名称空间中，以便相互隔离各自配置。

使用以下命令可指定每个 NFS Ganesha 集群的配置位置：

```
cephuser@adm > ceph dashboard set-ganesha-clusters-rados-pool-
namespace cluster_id:pool_name[/namespace](,cluster_id:pool_name[/namespace])*
```

`cluster_id` 是唯一标识 NFS Ganesha 集群的任意字符串。

使用多个 NFS Ganesha 集群配置 Ceph Dashboard 时，Web UI 会自动让您选择导出项的所属集群。

11.7 调试插件

Ceph Dashboard 插件扩展了仪表盘的功能。调试插件允许根据调试模式来自定义仪表盘的行为。可以使用以下命令启用、禁用或检查调试插件：

```
cephuser@adm > ceph dashboard debug status
Debug: 'disabled'
cephuser@adm > ceph dashboard debug enable
Debug: 'enabled'
cephuser@adm > dashboard debug disable
Debug: 'disabled'
```

默认情况下，它处于禁用状态。这是适用于生产部署的推荐设置。如果需要，可以启用调试模式，而无需重启动。

II 集群操作

12 确定集群状态 85

13 操作任务 112

14 Ceph 服务的操作 136

15 备份和恢复 141

16 监控和告警 144

12 确定集群状态

当集群正在运行时，您可以使用 **ceph** 工具来监视它。要确定集群状态，通常需要检查 Ceph OSD、Ceph Monitor、归置组和元数据服务器的状态。

提示：交互方式

要以交互模式运行 **ceph** 工具，请不带任何参数在命令行中键入 **ceph**。如果要在一行中输入多条 **ceph** 命令，则使用交互模式较为方便。例如：

```
cephuser@adm > ceph
ceph> health
ceph> status
ceph> quorum_status
ceph> mon stat
```

12.1 检查集群的状态

您可以使用 **ceph status** 或 **ceph -s** 了解集群的即时状态：

```
cephuser@adm > ceph -s
cluster:
    id:     b4b30c6e-9681-11ea-ac39-525400d7702d
    health: HEALTH_OK

  services:
    mon: 5 daemons, quorum ses-min1,ses-master,ses-min2,ses-min4,ses-min3 (age
 2m)
    mgr: ses-min1.gpijpm(active, since 3d), standbys: ses-min2.oopvyh
    mds: my_cephfs:1 {0=my_cephfs.ses-min1.oterul=up:active}
    osd: 3 osds: 3 up (since 3d), 3 in (since 11d)
    rgw: 2 daemons active (myrealm.myzone.ses-min1.kwwazo, myrealm.myzone.ses-
min2.jngabw)
```

```
task status:
  scrub status:
      mds.my_cephfs.ses-min1.oterul: idle

data:
  pools:   7 pools, 169 pgs
  objects: 250 objects, 10 KiB
  usage:   3.1 GiB used, 27 GiB / 30 GiB avail
  pgs:     169 active+clean
```

输出内容提供了以下信息：

- 集群 ID
- 集群健康状况状态
- Monitor 索引版本号和 Monitor 仲裁的状态
- OSD 索引版本号和 OSD 的状态
- Ceph Manager 的状态
- 对象网关的状态
- 归置组索引版本
- 归置组和存储池数量
- **理论上**存储的数据量和存储的对象数量
- 所存储数据的总量。

提示：Ceph 计算数据使用率的方式

`used` 值反映实际使用的原始存储量。`xxx GB / xxx GB` 值表示集群的可用容量（两者中较小的数字），以及集群的整体存储容量。理论数量反映在复制、克隆所存储数据或创建其快照前这些数据的大小。因此，实际存储的数据量通常会超出理论上的存储量，因为 Ceph 会创建数据的副本，可能还会将存储容量用于克隆和创建快照。

显示即时状态信息的其他命令如下：

- **ceph pg stat**
- **ceph osd pool stats**
- **ceph df**
- **ceph df detail**

要实时更新信息，请在 **watch** 命令中以参数的方式使用以上任意命令（包括 **ceph -s**）：

```
# watch -n 10 'ceph -s'
```

如果您看累了，请按 Ctrl – C 。

12.2 检查集群健康状况

在启动集群后到开始读取和/或写入数据期间，检查集群的健康状况：

```
cephuser@adm > ceph health
HEALTH_WARN 10 pgs degraded; 100 pgs stuck unclean; 1 mons down, quorum 0,2 \
node-1,node-2,node-3
```

提示

如果之前为您的配置或密钥环指定了非默认位置，则此时可以指定它们的位置：

```
cephuser@adm > ceph -c /path/to/conf -k /path/to/keyring health
```

Ceph 集群会返回下列健康状况代码之一：

OSD_DOWN

一个或多个 OSD 标记为已停机。OSD 守护进程可能已停止，或同伴 OSD 可能无法通过网络连接 OSD。常见原因包括守护进程已停止或已崩溃、主机已停机或网络中断。

校验主机是否运行良好，守护进程是否已启动，并且网络是否正常工作。如果守护进程已崩溃，守护进程日志文件 (/var/log/ceph/ceph-osd.*) 可能会包含调试信息。

OSD_crush type_DOWN，例如 OSD_HOST_DOWN

特定 CRUSH 子树中的所有 OSD 均标记为已停机，例如主机上的所有 OSD。

OSD_ORPHAN

在 CRUSH 索引层次结构中引用了 OSD，但它不存在。可使用以下命令从 CRUSH 层次结构中删除 OSD：

```
cephuser@adm > ceph osd crush rm osd.ID
```

OSD_OUT_OF_ORDER_FULL

以下项的使用率阈值未按升序排列：**backfillfull**（默认值为 0.90）、**nearfull**（默认值为 0.85）、**full**（默认值为 0.95）、**failsafe_full**。特别是，我们需要 **backfillfull** < **nearfull**，**nearfull** < **full** 且 **full** < **failsafe_full**。

要读取最新的值，请运行以下命令：

```
cephuser@adm > ceph health detail
HEALTH_ERR 1 full osd(s); 1 backfillfull osd(s); 1 nearfull osd(s)
osd.3 is full at 97%
osd.4 is backfill full at 91%
osd.2 is near full at 87%
```

可以使用以下命令调整阈值：

```
cephuser@adm > ceph osd set-backfillfull-ratio ratio
cephuser@adm > ceph osd set-nearfull-ratio ratio
cephuser@adm > ceph osd set-full-ratio ratio
```

OSD_FULL

一个或多个 OSD 超出了 **full** 阈值，阻止集群处理写入操作。可使用以下命令检查各存储池的用量：

```
cephuser@adm > ceph df
```

可使用以下命令查看当前定义的 **full** 比例：

```
cephuser@adm > ceph osd dump | grep full_ratio
```

恢复写入可用性的临时解决方法是稍稍提高 full 阈值：

```
cephuser@adm > ceph osd set-full-ratio ratio
```

请通过部署更多 OSD 将新的存储添加到集群，或者删除现有数据来腾出空间。

OSD_BACKFILLFULL

一个或多个 OSD 超出了 **backfillfull** 阈值，因而不允许将数据重新平衡到此设备。这是一条预警，意味着重新平衡可能无法完成，并且集群将满。可使用以下命令检查各存储池的用量：

```
cephuser@adm > ceph df
```

OSD_NEARFULL

一个或多个 OSD 超出了 **nearfull** 阈值。这是一条预警，意味着集群将满。可使用以下命令检查各存储池的用量：

```
cephuser@adm > ceph df
```

OSDMAP_FLAGS

已设置一个或多个所需的集群标志。可使用以下命令设置或清除这些标志（**full** 除外）：

```
cephuser@adm > ceph osd set flag
cephuser@adm > ceph osd unset flag
```

这些标志包括：

full

集群标记为已满，无法处理写入操作。

pauserd、pausewr

已暂停读取或写入。

noup

不允许 OSD 启动。

nodown

将会忽略 OSD 故障报告，如此 Monitor 便不会将 OSD 标记为 **down**。

noin

先前标记为 **out** 的 OSD 在启动时将不会重新标记为 **in**。

noout

停机的 OSD 在配置间隔过后将不会自动标记为 **out**。

nobackfill、norecover、norebalance

恢复或数据重新平衡进程已暂停。

noscrub、nodeep_scrub

洗刷进程已禁用（请参见第 17.6 节 “洗刷归置组”）。

notieragent

缓存分层活动已暂停。

OSD_FLAGS

一个或多个 OSD 设置了所需的每 OSD 标志。这些标志包括：

noup

不允许 OSD 启动。

nodown

将会忽略此 OSD 的故障报告。

noin

如果此 OSD 先前在发生故障后自动标记为 **out**，当它启动时将不会标记为 **in**。

noout

如果此 OSD 已停机，则在配置的间隔过后，它将不会自动标记为 **out**。

可使用以下命令来设置和清除每 OSD 标志：

```
cephuser@adm > ceph osd add-flag osd-ID
cephuser@adm > ceph osd rm-flag osd-ID
```

OLD_CRUSH_TUNABLES

CRUSH 索引目前使用的设置很旧，应予以更新。`mon_crush_min_required_version` 配置选项可确定使用时不会触发此健康状况警告的最旧可调变量（即能够连接到集群的最旧客户端版本）。

OLD_CRUSH_STRAW_CALC_VERSION

CRUSH 索引目前使用较旧的非最佳方法来计算 straw 存储桶的中间权重值。应该更新 CRUSH 索引以使用较新的方法 (`straw_calc_version`=1)。

CACHE_POOL_NO_HIT_SET

一个或多个缓存池未配置命中集来跟踪用量，这使分层代理无法识别要从缓存中刷新和赶出的冷对象。可使用以下命令对快速缓存池配置命中集：

```
cephuser@adm > ceph osd pool set poolname hit_set_type type
cephuser@adm > ceph osd pool set poolname hit_set_period period-in-seconds
cephuser@adm > ceph osd pool set poolname hit_set_count number-of-hitsets
cephuser@adm > ceph osd pool set poolname hit_set_fpp target-false-
positive-rate
```

OSD_NO_SORTBITWISE

未在运行早于 Luminous 12 版本的 OSD，但是尚未设置 `sortbitwise` 标志。您需要先设置 `sortbitwise` 标志，Luminous 12 或更新版本的 OSD 才能启动：

```
cephuser@adm > ceph osd set sortbitwise
```

POOL_FULL

一个或多个存储池已达到其配额，不再允许写入。可使用以下命令设置存储池配额和用量：

```
cephuser@adm > ceph df detail
```

您可以使用以下命令提高存储池配额

```
cephuser@adm > ceph osd pool set-quota poolname max_objects num-objects
cephuser@adm > ceph osd pool set-quota poolname max_bytes num-bytes
```

或者删除一些现有数据以减少用量。

PG_AVAILABILITY

数据可用性下降，这意味着集群无法处理针对集群中某些数据的潜在读取或写入请求。具体而言，一个或多个 PG 处于不允许处理 I/O 请求的状态。有问题的 PG 状态包括**正在互联**、**过时**、**不完整**和不**工作**（如果这些状况不迅速解决）。运行以下命令可获得有关哪些 PG 受影响的详细信息：

```
cephuser@adm > ceph health detail
```

大多数情况下，出现此情形的根本原因在于一个或多个 OSD 当前已停机。可使用以下命令查询特定的有问题 PG 的状态：

```
cephuser@adm > ceph tell pgid query
```

PG_DEGRADED

某些数据的数据冗余降低，这意味着集群没有所需数量的副本用于所有数据（对于副本存储池）或纠删码分段（对于纠删码存储池）。具体而言，一个或多个 PG 设置了 **degraded** 或 **undersized** 标志（集群中没有该归置组的足够实例），或者有一段时间未设置 **clean** 标志。运行以下命令可获得有关哪些 PG 受影响的详细信息：

```
cephuser@adm > ceph health detail
```

大多数情况下，出现此情形的根本原因在于一个或多个 OSD 当前已停机。可使用以下命令查询特定的有问题 PG 的状态：

```
cephuser@adm > ceph tell pgid query
```

PG_DEGRADED_FULL

由于集群中的可用空间不足，某些数据的数据冗余可能已降低或面临风险。具体而言，一个或多个 PG 设置了 **backfill_toofull** 或 **recovery_tooful** 标志，这意味着集群无法迁移或恢复数据，原因是一个或多个 OSD 高于 **backfillfull** 阈值。

PG_DAMAGED

数据洗刷（请参见第 17.6 节 “洗刷归置组”）进程发现集群中存在某些数据一致性问题。具体而言，一个或多个 PG 设置了 **inconsistent** 或 **snaptrim_error** 标志（表示某个较早的洗刷操作发现问题），或者设置了 **repair** 标志（表示当前正在修复此类不一致问题）。

OSD_SCRUB_ERRORS

最近的 OSD 洗刷操作发现了不一致问题。

CACHE_POOL_NEAR_FULL

缓存层池将满。在此环境中，“满”由缓存池的 **target_max_bytes** 和 **target_max_objects** 属性确定。池达到目标阈值时，如果正在从缓存刷新并赶出数据，写入池的请求可能会被阻止，出现常会导致延迟很高且性能变差的状态。可使用以下命令调整缓存池目标大小：

```
cephuser@adm > ceph osd pool set cache-pool-name target_max_bytes bytes
cephuser@adm > ceph osd pool set cache-pool-name target_max_objects objects
```

正常的快速缓存清理和逐出活动还可能因基础层可用性或性能下降或者集群的整体负载较高而受到限制。

TOO_FEW_PGS

使用中的 PG 数量低于每个 OSD 的 PG 数的可配置阈值 mon_pg_warn_min_per_osd。这可能导致集群中各 OSD 间的数据分布和平衡未达到最佳，以致降低整体性能。

TOO_MANY_PGS

使用中的 PG 数量高于每个 OSD 的 PG 数的可配置阈值 mon_pg_warn_max_per_osd。这可能导致 OSD 守护进程的内存用量较高，集群状态更改（例如 OSD 重启动、添加或删除）之后互联速度降低，并且 Ceph Manager 和 Ceph Monitor 上的负载较高。

虽然不能减少现有存储池的 pg_num 值，但可以减少 pgp_num 值。这样可有效地在同组 OSD 上共置一些 PG，从而减轻上述的一些负面影响。可使用以下命令调整 pgp_num 值：

```
cephuser@adm > ceph osd pool set pool pgp_num value
```

SMALLER_PGP_NUM

一个或多个存储池的 pgp_num 值小于 pg_num。这通常表示 PG 计数有所提高，但未同时提升归置行为。使用以下命令设置 pgp_num，使其与触发数据迁移的 pg_num 相匹配，通常便可解决此问题：

```
cephuser@adm > ceph osd pool set pool pgp_num pg_num_value
```

MANY_OBJECTS_PER_PG

一个或多个存储池的每 PG 平均对象数大大高于集群的整体平均值。该特定阈值通过 mon_pg_warn_max_object_skew 配置值控制。这通常表示包含集群中大部分数据的存储池具有的 PG 太少，以及/或者不包含这么多数据的其他存储池具有的 PG 太多。可通过调整 Monitor 上的 mon_pg_warn_max_object_skew 配置选项提高阈值，来消除该健康状况警告。

POOL_APP_NOT_ENABLED

存在包含一个或多个对象但尚未标记为供特定应用使用的存储池。将存储池标记为供某个应用使用即可消除此警告。例如，如果存储池由 RBD 使用：

```
cephuser@adm > rbd pool init pool_name
```

如果存储池正由自定义应用“foo”使用，您还可以使用低级别命令标记它：

```
cephuser@adm > ceph osd pool application enable foo
```

POOL_FULL

一个或多个存储池已达到（或几乎要达到）其配额。触发此错误状况的阈值通过 `mon_pool_quota_crit_threshold` 配置选项控制。可使用以下命令上调、下调（或删除）存储池配额：

```
cephuser@adm > ceph osd pool set-quota pool max_bytes bytes
cephuser@adm > ceph osd pool set-quota pool max_objects objects
```

将配额值设置为 0 将禁用配额。

POOL_NEAR_FULL

一个或多个存储池接近其配额。触发此警告状况的阈值通过 `mon_pool_quota_warn_threshold` 配置选项控制。可使用以下命令上调、下调（或删除）存储池配额：

```
cephuser@adm > ceph osd osd pool set-quota pool max_bytes bytes
cephuser@adm > ceph osd osd pool set-quota pool max_objects objects
```

将配额值设置为 0 将禁用配额。

OBJECT_MISPLACED

集群中的一个或多个对象未存储在集群希望存储的节点上。这表示集群最近的某项更改导致的数据迁移尚未完成。误放的数据本质上不属于危险状况。数据一致性方面永远不会有风险，仅当所需位置放置了对象所需份数的新副本之后，系统才会删除对象的旧副本。

OBJECT_UNFOUND

找不到集群中的一个或多个对象。具体而言，OSD 知道对象的新副本或更新副本应该存在，但在当前启用的 OSD 上却找不到该版对象的副本。系统将阻止对“未找到”对象的读取或写入请求。从理论上讲，可以将具有未找到对象最近副本的已停用 OSD 重新启用。可通过负责处理未找到对象的 PG 的互联状态识别候选 OSD：

```
cephuser@adm > ceph tell pgid query
```

REQUEST_SLOW

正花费很长的时间处理一个或多个 OSD 请求。这可能表示负载极重、存储设备速度缓慢或有软件错误。可以从 OSD 主机执行以下命令来查询有问题的 OSD 上的请求队列：

```
cephuser@adm > cephadm enter --name osd.ID -- ceph daemon osd.ID ops
```

可以查看近期最慢的请求摘要：

```
cephuser@adm > cephadm enter --name osd.ID -- ceph daemon osd.ID
 dump_historic_ops
```

可使用以下命令查找 OSD 的位置：

```
cephuser@adm > ceph osd find osd.id
```

REQUEST_STUCK

一个或多个 OSD 请求已被阻止一段相当长的时间，例如 4096 秒。这表示集群已有很长一段时间处于非健康状况（例如，没有足够的运行中 OSD 或非活跃 PG），或者 OSD 存在某种内部问题。

PG_NOT_SCRUBBED

最近未洗刷（请参见第 17.6 节 “洗刷归置组”）一个或多个 PG。通常每 `mon_scrub_interval` 秒洗刷一次 PG，当 `mon_warn_not_scrubbed` 这类间隔已过但未进行洗刷时，就会触发此警告。如果 PG 未标记为清理，系统将不会洗刷它们。如果 PG 放置错误或已降级，就会出现这种情况（请参见上文中的 PG_AVAILABILITY 和 PG_DEGRADED）。您可以使用以下命令手动对标记为清理的 PG 启动洗刷：

```
cephuser@adm > ceph pg scrub pgid
```

PG_NOT_DEEP_SCRUBBED

最近未深层洗刷（请参见第 17.6 节 “洗刷归置组”）一个或多个 PG。通常每 `osd_deep_scrub_interval` 秒洗刷一次 PG，当 `mon_warn_not_deep_scrubbed` 秒已过但未进行洗刷时，就会触发此警告。如果 PG 未标记为清理，系统将不会（深层）洗刷它们。如果 PG 放置错误或已降级，就会出现这种情况（请参见上文中的 PG_AVAILABILITY 和 PG_DEGRADED）。您可以使用以下命令手动对标记为清理的 PG 启动洗刷：

```
cephuser@adm > ceph pg deep-scrub pgid
```

提示

如果之前为您的配置或密钥环指定了非默认位置，则此时可以指定它们的位置：

```
# ceph -c /path/to/conf -k /path/to/keyring health
```

12.3 检查集群的使用率统计数据

要查看集群的数据使用率以及数据在多个存储池之间的分布，请使用 **ceph df** 命令。要获取更多详细信息，请使用 **ceph df detail**。

```
cephuser@adm > ceph df
--- RAW STORAGE ---
CLASS  SIZE    AVAIL   USED     RAW USED  %RAW USED
hdd    30 GiB  27 GiB  121 MiB   3.1 GiB      10.40
TOTAL  30 GiB  27 GiB  121 MiB   3.1 GiB      10.40

--- POOLS ---
POOL                   ID  STORED   OBJECTS  USED     %USED  MAX AVAIL
device_health_metrics   1      0 B        0      0 B      0    8.5 GiB
cephfs.my_cephfs.meta   2  1.0 MiB       22  4.5 MiB   0.02    8.5 GiB
cephfs.my_cephfs.data   3      0 B        0      0 B      0    8.5 GiB
.rgw.root               4  1.9 KiB       13  2.2 MiB      0    8.5 GiB
myzone.rgw.log          5  3.4 KiB      207    6 MiB   0.02    8.5 GiB
myzone.rgw.control      6      0 B        8      0 B      0    8.5 GiB
myzone.rgw.meta         7      0 B        0      0 B      0    8.5 GiB
```

输出中的 RAW STORAGE 段落提供集群用于数据的存储空间容量概览。

- CLASS：设备的存储类别。有关设备类型的更多详细信息，请参见第 17.1.1 节 “设备类型”。
- SIZE：集群的整体存储容量。
- AVAIL：集群中可以使用的可用空间容量。
- USED：单纯为块设备中保存的数据对象分配的空间（所有 OSD 上的累计空间）。
- RAW USED：“USED”空间与块设备上为实现 Ceph 而分配/预留的空间（例如 BlueStore 的 BlueFS 部分）之和。
- % RAW USED：已用的原始存储量百分比。将此数字与 full ratio 和 near full ratio 搭配使用，可确保您不会用完集群的容量。有关其他详细信息，请参见第 12.8 节 “储存容量”。

注意：集群填充程度

当原始存储填满级别接近 100% 时，您需要向集群添加新存储空间。较高的用量可能导致单个 OSD 填满，集群健康状况出现问题。

使用命令 **`ceph osd df tree`** 可列出所有 OSD 的填充程度。

输出内容的 POOLS 段落提供了存储池列表和每个存储池的理论用量。此段落的输出**不**反映副本、克隆数据或快照。例如，如果您存储含 1MB 数据的对象，理论用量将是 1MB，但是根据副本、克隆数据或快照数量，实际用量可能是 2MB 或更多。

- POOL：存储池的名称。
- ID：存储池 ID。
- STORED：用户存储的数据量。
- OBJECTS：每个存储池的理论已存储对象数。
- USED：所有 OSD 节点单纯为数据分配的空间容量，以 kB 为单位。
- %USED：每个存储池的理论已用存储百分比。
- MAX AVAIL：给定存储池中的最大可用空间。

注意

POOLS 段落中的数字是理论上的。它们不包括副本、快照或克隆数量。因此，USED 与 %USED 数量之和不会加总到输出内容 RAW STORAGE 段落中的 RAW USED 和 %RAW USED 数量中。

12.4 检查 OSD 状态

可通过执行以下命令来检查 OSD，以确保它们已启动且正在运行：

```
cephuser@adm > ceph osd stat
```

或

```
cephuser@adm > ceph osd dump
```

还可以根据 OSD 在 CRUSH 索引中的位置查看 OSD。

ceph osd tree 将列显 CRUSH 树及主机、它的 OSD、OSD 是否已启动及其权重：

```
   cephuser@adm > ceph osd tree
ID  CLASS  WEIGHT   TYPE NAME              STATUS  REWEIGHT  PRI-AFF
-1      3  0.02939  root default
-3      3  0.00980    rack mainrack
-2      3  0.00980            host osd-host
0       1  0.00980                    osd.0   up   1.00000   1.00000
1       1  0.00980                    osd.1   up   1.00000   1.00000
2       1  0.00980                    osd.2   up   1.00000   1.00000
```

12.5 检查填满的 OSD

Ceph 可阻止您向填满的 OSD 写入数据，以防丢失数据。在正常运行的集群中，当集群接近其填满比例时，您会收到警告。**mon osd full ratio** 默认设为容量的 0.95 (95%)，达到该比例后，集群会阻止客户端写入数据。**mon osd nearfull ratio** 默认设为容量的 0.85 (85%)，达到该比例时，集群会生成健康状况警告。

可通过 **ceph health** 命令报告填满的 OSD 节点：

```
cephuser@adm > ceph health
  HEALTH_WARN 1 nearfull osds
  osd.2 is near full at 85%
```

或

```
cephuser@adm > ceph health
  HEALTH_ERR 1 nearfull osds, 1 full osds
  osd.2 is near full at 85%
  osd.3 is full at 97%
```

处理填满的集群的最佳方法是添加新的 OSD 主机/磁盘，以便让集群将数据重新分布到新的可用存储空间。

提示：防止 OSD 填满

OSD 变满（即用完 100% 的磁盘空间）之后，往往会迅速崩溃而不发出警告。管理 OSD 节点时需记住下面几点提示。

- 每个 OSD 的磁盘空间（通常挂载在 `/var/lib/ceph/osd/osd-{1,2..}` 下）需放置在专用的底层磁盘或分区上。
- 检查 Ceph 配置文件，确保 Ceph 不会将其日志文件存储在专供 OSD 使用的磁盘/分区上。
- 确保没有其他进程写入专供 OSD 使用的磁盘/分区。

12.6 检查 Monitor 状态

启动集群后，请在第一次读取和/或写入数据之前检查 Ceph Monitor 的仲裁状态。如果集群已在处理请求，请定期检查 Ceph Monitor 的状态，确保其正在运行。

要显示 Monitor 索引，请执行以下命令：

```
cephuser@adm > ceph mon stat
```

或

```
cephuser@adm > ceph mon dump
```

要检查 Monitor 集群的仲裁状态，请执行以下命令：

```
cephuser@adm > ceph quorum_status
```

Ceph 将返回仲裁状态。例如，由三个 Monitor 组成的 Ceph 集群可能返回以下内容：

```
{ "election_epoch": 10,
  "quorum": [
        0,
        1,
        2],
  "monmap": { "epoch": 1,
      "fsid": "444b489c-4f16-4b75-83f0-cb8097468898",
```

```
      "modified": "2011-12-12 13:28:27.505520",
      "created": "2011-12-12 13:28:27.505520",
      "mons": [
            { "rank": 0,
              "name": "a",
              "addr": "192.168.1.10:6789\/0"},
            { "rank": 1,
              "name": "b",
              "addr": "192.168.1.11:6789\/0"},
            { "rank": 2,
              "name": "c",
              "addr": "192.168.1.12:6789\/0"}
           ]
    }
}
```

12.7 检查归置组状态

归置组会将对象映射到 OSD。监视归置组时，您希望它们处于 `active` 和 `clean` 状态。有关详细内容，请参见第 12.9 节 “监控 OSD 和归置组”。

12.8 储存容量

作为防止数据丢失的安全措施，当 Ceph 存储集群接近其容量上限时，Ceph 将阻止您向 Ceph OSD 写入或从中读取数据。因此，让生产集群接近其填满比例不是一种好的做法，因为这样会牺牲高可用性。默认的填满比例设置为 0.95，即容量的 95%。对于所含 OSD 数量较少的测试集群而言，如此设置是非常激进的。

提示：增加存储容量

在监视集群时，请注意与 `nearfull` 比例有关的警告。出现该警告表示，如果一个或多个 OSD 发生故障，某些 OSD 的故障可能会导致服务暂时中断。请考虑添加更多 OSD 以增加存储容量。

测试集群的一种常见情境是系统管理员从 Ceph 存储集群中删除 Ceph OSD，等待集群重新达到平衡。然后再删除另一个 Ceph OSD，以此类推，直到集群最终达到填满比例并锁死。我们建议即使使用测试集群时也进行一定的容量规划。通过规划，您可以预估维持高可用性所需的备用容量。从理论上讲，您需要规划能够应对一系列 Ceph OSD 发生故障的情况的方案，使集群无需立即替换这些 Ceph OSD 也可恢复到 `active + clean` 状态。您可以运行 `active + degraded` 状态的集群，但这不适合正常运行状况。

下图展示了一个包含 33 个 Ceph 节点的简化 Ceph 存储集群，其中每个主机有一个 Ceph OSD，每个 Ceph OSD 从 3 TB 驱动器读取以及向其中写入数据。此示例集群实际的容量上限为 99 TB。`mon osd full ratio` 选项设置为 0.95。如果集群的剩余容量降至 5 TB，集群将不允许客户端读取和写入数据。因此，存储集群的运行容量为 95 TB，而不是 99 TB。

图 12.1： CEPH 集群

在这样的集群中，有一个或两个 OSD 发生故障属于正常现象。一种不常发生但合乎常理的情况是机柜的路由器或电源发生故障，导致多个 OSD（例如 OSD 7-12）同时停用。在这种情况下，您仍然应该设法使集群保持正常运行并达到 `active + clean` 状态，即使这意味着需要立即添加一些主机及额外的 OSD。如果容量使用率过高，您可能不会丢失数据。但是，如果集群的容量使用率超过填满比例，您虽然解决了故障域内发生的中断问题，却可能会损失数据可用性。因此，我们建议至少进行大致的容量规划。

针对您的集群确定以下两个数值：

1. OSD 的数量。
2. 集群的总容量。

如果您将集群的总容量除以集群中的 OSD 数量，将得到集群内单个 OSD 的平均容量。将该数值与您预期正常运行期间将同时发生故障的 OSD 数量（一个相对较小的数值）相乘。最后，将集群容量与填满比例相乘得到运行容量上限。然后，减去您预期将发生故障的 OSD 中的数据量，即可得到一个合理的填满比例。使用更高的 OSD 故障数（一整个机柜的 OSD）重复上述过程，即可得到一个合理的接近填满比例数值。

以下设置仅在创建集群时才会应用，随后会存储在 OSD 索引中：

```
[global]
 mon osd full ratio = .80
 mon osd backfillfull ratio = .75
 mon osd nearfull ratio = .70
```

提示

仅在创建集群时才会应用这些设置。此后，需要使用 **`ceph osd set-nearfull-ratio`** 和 **`ceph osd set-full-ratio`** 命令在 OSD 索引中更改这些设置。

mon osd full ratio

在将 OSD 视为满之前使用的磁盘空间百分比。默认值为 0.95

mon osd backfillfull ratio

在将 OSD 视为过满而无法回填之前使用的磁盘空间百分比。默认值为 0.90

mon osd nearfull ratio

在将 OSD 视为将满之前使用的磁盘空间百分比。默认值为 0.85

提示：检查 OSD 权重

如果某些 OSD 将满，但其他 OSD 的容量充足，则表示将满 OSD 的 CRUSH 权重可能有问题。

12.9 监控 OSD 和归置组

高可用性和高可靠性要求采用容错方法来管理硬件和软件问题。Ceph 没有单一故障点，可以在“已降级”模式下处理数据请求。Ceph 的数据归置引入了一个间接层，以确保数据不会与特定 OSD 地址直接绑定。这表示跟踪系统故障原因需要找到属于问题根源的归置组和底层 OSD。

提示：发生故障时的访问

如果集群的某个部分发生故障，集群可能会阻止您访问特定对象，但这并不意味着您无法访问其他对象。遇到故障时，请执行相关步骤来监视 OSD 和归置组。然后开始进行查错。

Ceph 一般情况下会进行自我修复。但如果问题仍然存在，监视 OSD 和归置组将有助于您找到问题所在。

12.9.1 监视 OSD

OSD 可能处于**在集群内**（“in”）状态，也可能处于**在集群外**（“out”）状态。同时，它也可能处于**启用并运行**（“up”）或**停用且未运行**（“down”）状态。如果某个 OSD 处于“up”状态，则它可能在集群内（您可以读取和写入数据），也可能在集群外。如果该 OSD 之前在集群内，最近已移出集群，则 Ceph 会将归置组迁移到其他 OSD。如果某个 OSD 在集群外，CRUSH 将不会为其指定归置组。如果某个 OSD 处于“down”状态，则它应该也处于“out”状态。

注意：不健康状态

如果某个 OSD 处于“down”和“in”状态，则表示存在问题，并且集群将处于不健康状态。

如果您执行 `ceph health`、`ceph -s` 或 `ceph -w` 等命令，可能会注意到集群并非始终回显 `HEALTH OK`。对于 OSD，您应当预期集群在以下情况下**不会**回显 `HEALTH OK`：

- 您尚未启动集群（它不会响应）。
- 您已启动或重启动集群，但它尚未准备就绪，因为系统正在创建归置组，并且 OSD 正在互联。
- 您已添加或删除某个 OSD。
- 您已修改集群索引。

监视 OSD 的一个重要目的是确保当集群已启用且在运行时，集群中的所有 OSD 也已启用且在运行。要确定是否所有 OSD 都在运行，请执行以下命令：

```
# ceph osd stat
x osds: y up, z in; epoch: eNNNN
```

结果应显示 OSD 总数 (x)、处于“up”状态的 OSD 数量 (y)、处于“in”状态的 OSD 数量 (z)，以及索引版本号 (eNNNN)。如果在集群内（“in”）的 OSD数量大于处于“up”状态的 OSD 数量，请执行以下命令确定未在运行的 `ceph-osd` 守护进程：

```
# ceph osd tree
#ID CLASS WEIGHT  TYPE NAME             STATUS REWEIGHT PRI-AFF
-1       2.00000 pool openstack
-3       2.00000 rack dell-2950-rack-A
-2       2.00000 host dell-2950-A1
0   ssd 1.00000      osd.0                up  1.00000 1.00000
1   ssd 1.00000      osd.1              down  1.00000 1.00000
```

例如，如果 ID 为 1 的 OSD 处于停用状态，请将其启动：

```
cephuser@osd > sudo systemctl start ceph-CLUSTER_ID@osd.0.service
```

有关与已停止或不会重启动的 OSD 相关的问题，请参见《Troubleshooting Guide》, 第 4 章 “Troubleshooting OSDs”, 第 4.3 节 “OSDs not running”。

12.9.2 指定归置组集

CRUSH 向 OSD 指定归置组时，会查看存储池的副本数量，然后再为 OSD 指定归置组，以便将每个归置组副本都指定给不同的 OSD。例如，如果存储池需要三个归置组副本，CRUSH 可能会将这三个副本分别指定给 `osd.1`、`osd.2` 和 `osd.3`。CRUSH 实际上会寻找一种伪随机归置方法，这种方法会将您在 CRUSH 索引中设置的故障域纳入考量，因此在大型集群中，您很少会看

到归置组被指定给最邻近的 OSD 的情况。我们将应包含特定归置组的副本的 OSD 集称为**在任集**。在某些情况下，在任集中的 OSD 会处于停用状态，或者无法处理要访问归置组中对象的请求。当以下其中一种情况发生时，可能会出现这些情况：

- 您添加或删除了某个 OSD。CRUSH 随后会将归置组重新指定给其他 OSD，因而更改了**在任集**的组成部分，导致系统通过“回填”过程迁移数据。
- 某个 OSD 之前处于“down”状态、之前进行了重启动，而现在正在恢复。
- **在任集**中的某个 OSD 处于“down”状态，或者无法处理请求，并且另一个 OSD 已暂代其职。

  Ceph 使用**启用集**来处理客户端请求，启用集是实际处理请求的 OSD 集。在大多数情况下，**启用集**和**在任集**几乎完全相同。当两者不同时，可能表示 Ceph 正在迁移数据、某个 OSD 正在恢复，或者集群存在问题（例如，在此类情况下，Ceph 通常会回显 HEALTH WARN 状态及“stuck stale”消息）。

要检索归置组列表，请运行以下命令：

```
cephuser@adm > ceph pg dump
```

要查看哪些 OSD 在给定归置组的**在任集**或**启用集**内，请运行以下命令：

```
cephuser@adm > ceph pg map PG_NUM
osdmap eNNN pg RAW_PG_NUM (PG_NUM) -> up [0,1,2] acting [0,1,2]
```

结果应该会显示 OSD 索引版本号 (eNNN)、归置组数量 (PG_NUM)、**启用集**（“up”）中的 OSD，以及**在任集**（“acting”）中的 OSD：

提示：集群问题指示符

如果**启用集**与**在任集**不一致，则可能表示集群正在自我重新平衡，或者集群可能存在问题。

12.9.3 正在互联

归置组必须处于 active 及 clean 状态，您才能将数据写入其中。为了让 Ceph 确定某个归置组的当前状态，该归置组的主 OSD（**在任集**中的第一个 OSD）会与第二个和第三个 OSD 建立互联，以便就归置组的当前状态达成一致（假设存储池包含三个归置组副本）。

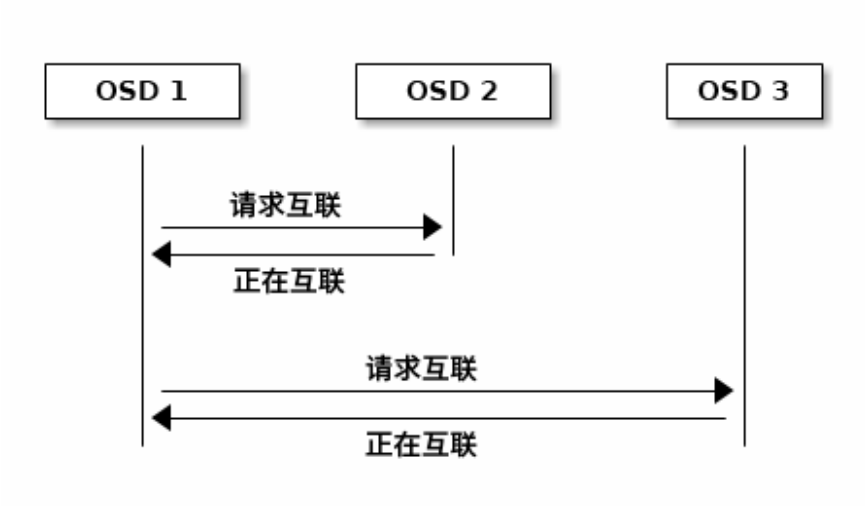


图 12.2： 互联纲要

12.9.4 监控归置组状态

如果您执行 `ceph health`、`ceph -s` 或 `ceph -w` 等命令，可能会注意到集群并非始终回显 `HEALTH OK` 消息。检查 OSD 是否正在运行之后，还应检查归置组状态。

在一些与归置组互联相关的情况下，集群预期将**不会**回显 `HEALTH OK`：

- 您已创建存储池，并且归置组尚未互联。
- 归置组正在恢复。
- 您已向集群添加了 OSD，或已从集群中删除了 OSD。
- 您已修改 CRUSH 索引，并且您的归置组正在迁移。
- 在不同的归置组副本中存在数据不一致的情况。
- Ceph 正在洗刷归置组的副本。
- Ceph 的存储容量不足，无法完成回填操作。

如果上述其中一种情况导致 Ceph 回显 `HEALTH WARN`，请不要惊慌。集群在许多情况下都会自行恢复。在有些情况下，您可能需要采取措施。监视归置组的一个重要目的是确保当集群已启用并运行时，所有归置组都处于“active”状态并且最好处于“clean”状态。要查看所有归置组的状态，请运行以下命令：

```
cephuser@adm > ceph pg stat
x pgs: y active+clean; z bytes data, aa MB used, bb GB / cc GB avail
```

结果应该会显示归置组总数 (x)、处于特定状态（例如“active+clean”）的归置组数量 (y)，以及存储的数据量 (z)。

除了归置组状态以外，Ceph 还会回显使用的存储容量 (aa)、剩余的存储容量 (bb)，以及归置组的总存储容量。在以下情况下，这些数值可能非常重要：

- 已达到 `near full ratio` 或 `full ratio`。
- 由于您的 CRUSH 配置中存在错误，您的数据未在集群中分布。

提示：归置组 ID

归置组 ID 由存储池编号（并非存储池名称）加一个句点 (.)和归置组 ID（一个十六进制数）组成。您可以在 **`ceph osd lspools`** 的输出中查看存储池编号及其名称。例如，默认存储池 `rbd` 与存储池编号 0 对应。完全限定的归置组 ID 的格式如下：

```
POOL_NUM.PG_ID
```

通常显示如下：

```
0.1f
```

要检索归置组列表，请运行以下命令：

```
cephuser@adm > ceph pg dump
```

您还可以将输出内容设置为 JSON 格式，并将其保存到文件中：

```
cephuser@adm > ceph pg dump -o FILE_NAME --format=json
```

要查询特定归置组，请运行以下命令：

```
cephuser@adm > ceph pg POOL_NUM.PG_ID query
```

以下列表详细说明了常见的归置组状态。

CREATING（正在创建）

当您创建存储池时，Ceph 会创建您指定数量的归置组。Ceph 会在创建一个或多个归置组时回显“creating”。创建归置组之后，属于归置组**在任集**的各 OSD 将会互联。完成互联过程时，归置组状态应该为“active+clean”，这表示 Ceph 客户端可以开始向归置组写入数据。

图 12.3： 归置组状态

PEERING（正在互联）

当 Ceph 在对归置组执行互联操作时，会在存储归置组副本的各 OSD 之间就该归置组中的对象和元数据的状态达成一致。当 Ceph 完成互联过程时，便表示存储归置组的各 OSD 之间就归置组的当前状态达成一致。不过，完成互联过程并**不**表示每个副本都有最新的内容。

注意：权威历史

在**在任集**的所有 OSD 都持续进行写入操作之前，Ceph 将**不会**向客户端确认写入操作。这样做可确保在上次成功完成互联操作之后，至少有一个**在任集**成员将拥有每个确认的写入操作的记录。

通过准确记录每个确认的写入操作，Ceph 可以构建并扩展一个新的权威归置组历史，即一个完整且完全有序的在任集，如果执行该在任集，会将 OSD 的归置组副本更新到最新状态。

ACTIVE（活动）

当 Ceph 完成互联过程时，归置组可能会变为 `active` 状态。`active` 状态表示通常可在主归置组和副本中使用归置组中的数据进行读取和写入操作。

CLEAN（正常）

如果归置组处于 `clean` 状态，则表示主 OSD 和副本 OSD 已成功互联，并且该归置组没有流浪副本。Ceph 已将归置组中的所有对象复制正确的次数。

DEGRADED（已降级）

当客户端将对象写入主 OSD 时，该主 OSD 负责将副本写入副本 OSD。主 OSD 将对象写入存储空间之后，归置组将保持“degraded”状态，直到主 OSD 收到了副本 OSD 发送的 Ceph 已成功创建副本对象的确认消息。

归置组有可能处于“active+degraded”状态，这是因为即使 OSD 尚未存储所有对象，它也可能处于“active”状态。如果某个 OSD 变成停用状态，Ceph 会将指定给该 OSD 的每个归置组都标记为“degraded”。当该 OSD 恢复启用状态后，各 OSD 必须再次互联。不过，如果某个已降级归置组处于“active”状态，客户端仍然可以将新对象写入该归置组。

如果某个 OSD 处于“down”状态，并且持续保持“degraded”状况，Ceph 可能会将该停用的 OSD 标记为“out”（表示移出集群），并将停用（“down”）的 OSD 的数据重新映射到另一个 OSD。从将 OSD 标记为“down”到将其标记为“out”相隔的时间通过 `mon osd down out interval` 选项控制，该选项默认设置为 600 秒。

归置组也可能处于“degraded”状态，当 Ceph 找不到应在归置组中的一个或多个对象时便会发生此情况。虽然您无法读取未找到的对象或向其写入数据，却仍然可以访问“degraded”状态的归置组中的所有其他对象。

RECOVERING（正在恢复）

Ceph 设计用于在发生硬件和软件问题时进行大规模容错。当 OSD 变成“down”状态时，其内容可能落后于归置组中其他副本的当前状态。当 OSD 恢复“up”状态时，必须更新归置组的内容以反映最新状态。在此期间，OSD 可能会显现出“recovering”状态。

恢复并非总是无足轻重的，因为硬件故障可能会导致多个 OSD 发生级联故障。例如，一个机架或机柜的网络交换机可能会发生故障，这可能会导致一些主机的 OSD 落后于集群的当前状态。解决故障之后，必须恢复每个 OSD。

Ceph 提供了一些设置，用来平衡新服务请求与恢复数据对象并将归置组恢复到最新状态的需求之间的资源争用。`osd recovery delay start` 设置允许 OSD 在启动恢复过程之前重启动、重新互联，甚至处理一些重放请求。`osd recovery thread timeout` 用于设置线程超时，因为有可能会有多个 OSD 交错发生故障、重启动以及重新互联。`osd recovery max active` 设置用于限制 OSD 将同时处理的恢复请求数，以防止 OSD 无法处理请求。`osd recovery max chunk` 设置用于限制恢复的数据块大小，以防出现网络拥塞。

BACK FILLING（正在回填）

当新 OSD 加入集群时，CRUSH 会将集群中 OSD 的归置组重新指定给新添加的 OSD。强制新 OSD 立即接受重新指定的归置组可能会使新 OSD 过载。向 OSD 回填归置组可让此过程在后台开始。回填完成后，新 OSD 将在准备就绪时开始处理请求。

在执行回填操作期间，系统可能会显示以下其中一种状态：“backfill_wait”表示回填操作待处理，但尚未执行；“backfill”表示正在进行回填操作；“backfill_too_full”表示已请求进行回填操作，但由于存储容量不足而无法完成。如果某个归置组无法回填，则可能会被视为“incomplete”。

Ceph 提供了一些设置来管理与向某个 OSD（尤其是新 OSD）重新指定归置组有关的负载。默认情况下，`osd max backfills` 将向或从一个 OSD 同时进行的最大回填数设置为 10。`backfill full ratio` 允许 OSD 在接近其填满比例（默认为 90%）时拒绝回填请求，并使用 **`ceph osd set-backfillfull-ratio`** 命令进行更改。如果某个 OSD 拒绝回填请求，`osd backfill retry interval` 可让 OSD 重试请求（默认在 10 秒后）。OSD 还可以设置 `osd backfill scan min` 和 `osd backfill scan max`，以管理扫描间隔（默认值分别为 64 和 512）。

REMAPPED（已重新映射）

当用于处理归置组的**在任集**发生变化时，数据会从旧**在任集**迁移到新**在任集**。新主 OSD 可能需要一段时间才能处理请求。因此，新主 OSD 可能会要求旧主 OSD 继续处理请求，直到归置组迁移完成。数据迁移完成时，映射将使用新**在任集**的主 OSD。

STALE（过时）

尽管 Ceph 使用检测信号来确保主机和守护进程正在运行，但 `ceph-osd` 守护进程也可能会卡住，无法及时报告统计数据（例如，当发生暂时的网络故障时）。默认情况下，OSD 守护进程每半秒钟 (0.5) 报告一次其归置组、引导及故障统计数据，这个频率高于检测信号阈值。如果某个归置组**在任集**的主 OSD 未能向 Monitor 报告，或者其他 OSD 已将该主 OSD 报告为“down”，则 Monitor 会将该归置组标记为“stale”。

当您启动集群后，集群常常会在互联过程完成之前显示为“stale”状态。集群运行一段时间之后，如果归置组显示为“stale”状态，则表示这些归置组的主 OSD 处于停用状态，或者未向 Monitor 报告归置组统计数据。

12.9.5 查找对象位置

要在 Ceph 对象存储中存储对象数据，Ceph 客户端需要设置对象名称并指定相关的存储池。Ceph 客户端会检索最新的集群索引，并且 CRUSH 算法会计算如何将对象映射到归置组，然后计算如何以动态方式将该归置组指定给 OSD。要查找对象位置，您只需知道对象名称和存储池名称。例如：

```
cephuser@adm > ceph osd map POOL_NAME OBJECT_NAME [NAMESPACE]
```

例 12.1： 查找对象

作为示范，我们来创建一个对象。在命令行上使用 **`rados put`** 命令指定对象名称“test-object-1”、包含一些对象数据的示例文件“testfile.txt”的路径，以及存储池名称“data”。

```
cephuser@adm > rados put test-object-1 testfile.txt --pool=data
```

要确认 Ceph 对象存储是否已存储对象，请运行以下命令：

```
cephuser@adm > rados -p data ls
```

现在，我们来确定对象位置。Ceph 将会输出对象的位置：

```
cephuser@adm > ceph osd map data test-object-1
osdmap e537 pool 'data' (0) object 'test-object-1' -> pg 0.d1743484 \
(0.4) -> up ([1,0], p0) acting ([1,0], p0)
```

要移除示例对象，只需使用 **`rados rm`** 命令将其删除：

```
cephuser@adm > rados rm test-object-1 --pool=data
```

13 操作任务

13.1 修改集群配置

要修改现有 Ceph 集群的配置，请执行以下步骤：

1. 将集群的当前配置导出到文件：

```
cephuser@adm > ceph orch ls --export --format yaml > cluster.yaml
```

2. 编辑包含配置的文件并更新相关行。可在《部署指南》, 第 8 章 “使用 cephadm 部署其余核心服务”和第 13.4.3 节 “使用 DriveGroups 规范添加 OSD。”中找到规范示例。

3. 应用新配置：

```
cephuser@adm > ceph orch apply -i cluster.yaml
```

13.2 添加节点

要向 Ceph 集群添加新节点，请执行以下步骤：

1. 在新主机上安装 SUSE Linux Enterprise Server 和 SUSE Enterprise Storage。有关更多信息，请参考《部署指南》, 第 5 章 “安装和配置 SUSE Linux Enterprise Server”。

2. 将主机配置为已存在 Salt 主控端的 Salt 受控端。有关更多信息，请参考《部署指南》, 第 6 章 “部署 Salt”。

3. 将新主机添加到 `ceph-salt` 并让 cephadm 能识别它，例如：

```
root@master # ceph-salt config /ceph_cluster/minions add ses-
min5.example.com
root@master # ceph-salt config /ceph_cluster/roles/cephadm add ses-
min5.example.com
```

有关更多信息，请参考《部署指南》, 第 7 章 “使用 `ceph-salt` 部署引导集群”, 第 7.2.2 节 “添加 Salt 受控端”。

4. 确认节点是否已添加到 `ceph-salt`：

```
root@master # ceph-salt config /ceph_cluster/minions ls
o- minions ..................................................... [Minions: 5]
[...]
  o- ses-min5.example.com ..................................... [no roles]
```

5. 将配置应用于新集群主机：

```
root@master # ceph-salt apply ses-min5.example.com
```

6. 确认新添加的主机现在是否属于 cephadm 环境：

```
cephuser@adm > ceph orch host ls
HOST                   ADDR                   LABELS   STATUS
[...]
ses-min5.example.com   ses-min5.example.com
```

13.3 删除节点

提示：删除 OSD

如果您要删除的节点运行有 OSD，请先从中删除 OSD，然后检查该节点上是否未运行任何 OSD。有关删除 OSD 的更多详细信息，请参见第 13.4.4 节 “删除 OSD”。

要从集群中删除节点，请执行以下操作：

1. 对于除 `node-exporter` 和 `crash` 之外的所有 Ceph 服务类型，请从集群归置规范文件（例如 `cluster.yml`）中删除节点的主机名。有关更多详细信息，请参见《部署指南》，第 8 章 “使用 cephadm 部署其余核心服务”，第 8.2 节 “服务和归置规范”。例如，如果您要删除名为 `ses-min2` 的主机，请从所有 `placement:` 部分中删除所有出现的 `- ses-min2`：

   更新

```
service_type: rgw
service_id: EXAMPLE_NFS
```

```
placement:
  hosts:
  - ses-min2
  - ses-min3
```

更改为

```
service_type: rgw
service_id: EXAMPLE_NFS
placement:
  hosts:
  - ses-min3
```

将您的更改应用于配置文件：

```
cephuser@adm > ceph orch apply -i rgw-example.yaml
```

2. 从 cephadm 的环境中删除节点：

```
cephuser@adm > ceph orch host rm ses-min2
```

3. 如果节点正在运行 `crash.osd.1` 和 `crash.osd.2` 服务，请通过在主机上运行以下命令将其删除：

```
root@minion > cephadm rm-daemon --fsid CLUSTER_ID --name SERVICE_NAME
```

例如：

```
root@minion > cephadm rm-daemon --fsid b4b30c6e... --name crash.osd.1
root@minion > cephadm rm-daemon --fsid b4b30c6e... --name crash.osd.2
```

4. 从您要删除的受控端中删除所有角色：

```
cephuser@adm > ceph-salt config /ceph_cluster/roles/tuned/throughput remove
 ses-min2
cephuser@adm > ceph-salt config /ceph_cluster/roles/tuned/latency remove
 ses-min2
cephuser@adm > ceph-salt config /ceph_cluster/roles/cephadm remove ses-min2
cephuser@adm > ceph-salt config /ceph_cluster/roles/admin remove ses-min2
```

如果您要删除的受控端是引导受控端，则还需要删除引导角色：

```
cephuser@adm > ceph-salt config /ceph_cluster/roles/bootstrap reset
```

5. 删除单个主机上的所有 OSD 后，从 CRUSH 索引中删除该主机：

```
cephuser@adm > ceph osd crush remove bucket-name
```

注意

存储桶名称应该与主机名相同。

6. 现在您便可以从集群中删除受控端：

```
cephuser@adm > ceph-salt config /ceph_cluster/minions remove ses-min2
```

重要

如果发生故障，并且您尝试删除的受控端处于永久断电状态，则需要从 Salt 主控端中删除节点：

```
root@master # salt-key -d minion_id
```

然后，手动将节点从 `pillar_root/ceph-salt.sls` 中删除。它通常位于 `/srv/pillar/ceph-salt.sls` 中。

13.4 OSD 管理

本节介绍如何在 Ceph 集群中添加、擦除或删除 OSD。

13.4.1 列出磁盘设备

要识别所有集群节点上已使用和未使用的磁盘设备，请通过运行以下命令列出这些设备：

```
cephuser@adm > ceph orch device ls
```

```
HOST       PATH      TYPE SIZE  DEVICE  AVAIL REJECT REASONS
ses-master /dev/vda  hdd  42.0G         False locked
ses-min1   /dev/vda  hdd  42.0G         False locked
ses-min1   /dev/vdb  hdd  8192M  387836 False locked, LVM detected, Insufficient
 space (<5GB) on vgs
ses-min2   /dev/vdc  hdd  8192M  450575 True
```

13.4.2 擦除磁盘设备

要重新使用磁盘设备，需要先将其擦除（或**清除**）：

```
ceph orch device zap HOST_NAME DISK_DEVICE
```

例如：

```
cephuser@adm > ceph orch device zap ses-min2 /dev/vdc
```

注意

如果您之前在未设置 `unmanaged` 标志的情况下使用 DriveGroups 或 `--all-available-devices` 选项部署 OSD，cephadm 将在您擦除这些 OSD 后再自动部署它们。

13.4.3 使用 DriveGroups 规范添加 OSD。

DriveGroups 用于指定 Ceph 集群中 OSD 的布局。它们在单个 YAML 文件中定义。在本节中，我们将使用 `drive_groups.yml` 作为示例。

管理员应手动指定一组相关的 OSD（部署在 HDD 和 SDD 组合上的混合 OSD）或一组使用相同部署选项的 OSD（例如，对象存储、加密选项相同的各独立 OSD）。为了避免明确列出设备，DriveGroups 会使用与 **`ceph-volume`** 库存报告中的几个所选字段对应的过滤项列表。cephadm 将提供一些代码，用于将这些 DriveGroups 转换为供用户检查的实际设备列表。

将 OSD 规范应用于集群的命令是：

```
cephuser@adm > ceph orch apply osd -i drive_groups.yml
```

要查看操作预览并测试您的应用，可以将 `--dry-run` 选项与 **`ceph orch apply osd`** 命令一起使用。例如：

```
cephuser@adm > ceph orch apply osd -i drive_groups.yml --dry-run
...
+---------+------+------+----------+----+-----+
|SERVICE  |NAME  |HOST  |DATA      |DB  |WAL  |
+---------+------+------+----------+----+-----+
|osd      |test  |mgr0  |/dev/sda  |-   |-    |
|osd      |test  |mgr0  |/dev/sdb  |-   |-    |
+---------+------+------+----------+----+-----+
```

如果 `--dry-run` 输出符合您的预期，则只需重新运行命令而无需使用 `--dry-run` 选项。

13.4.3.1 非受管 OSD

将符合 DriveGroups 规范的所有可用正常磁盘设备添加到集群后，它们将自动用作 OSD。此行为称为**受管**模式。

要禁用**受管**模式，请将 `unmanaged: true` 行添加到相关规范，例如：

```
service_type: osd
service_id: example_drvgrp_name
placement:
 hosts:
 - ses-min2
 - ses-min3
encrypted: true
unmanaged: true
```

提示

要将已部署的 OSD 从**受管**模式更改为**非受管**模式，请于第 13.1 节 “修改集群配置”中所述的过程期间在适当位置添加 `unmanaged: true` 行。

13.4.3.2 DriveGroups 规范

下面是 DriveGroups 规范文件的示例：

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  drive_spec: DEVICE_SPECIFICATION
db_devices:
  drive_spec: DEVICE_SPECIFICATION
wal_devices:
  drive_spec: DEVICE_SPECIFICATION
block_wal_size: '5G'  # (optional, unit suffixes permitted)
block_db_size: '5G'   # (optional, unit suffixes permitted)
encrypted: true       # 'True' or 'False' (defaults to 'False')
```

注意

之前在 DeepSea 中名为“加密”的选项已被重命名为“已加密”。在 SUSE Enterprise Storage 7 中应用 DriveGroups 时，请确保在服务规范中使用此新术语，否则 **`ceph orch apply`** 操作将失败。

13.4.3.3 匹配磁盘设备

您可以使用以下过滤器描述规格：

- 按磁盘型号：

  ```
  model: DISK_MODEL_STRING
  ```

- 按磁盘供应商：

  ```
  vendor: DISK_VENDOR_STRING
  ```

提示

始终以小写形式输入 `DISK_VENDOR_STRING`。

要获取有关磁盘型号和供应商的详细信息，请检查以下命令的输出：

```
cephuser@adm > ceph orch device ls
HOST     PATH     TYPE  SIZE DEVICE_ID                  MODEL
 VENDOR
ses-min1 /dev/sdb ssd  29.8G SATA_SSD_AF34075704240015  SATA SSD
 ATA
ses-min2 /dev/sda ssd   223G Micron_5200_MTFDDAK240TDN  Micron_5200_MTFD
 ATA
[...]
```

- 磁盘是否为旋转硬盘。SSD 和 NVMe 驱动器不属于旋转硬盘。

```
rotational: 0
```

- 部署将**所有**可用驱动器都用于 OSD 的节点：

```
data_devices:
  all: true
```

- 还可通过限制匹配磁盘的数量来过滤：

```
limit: 10
```

13.4.3.4 按大小过滤设备

您可以按磁盘设备大小对其过滤，可以按确切大小也可以按大小范围来过滤。`size:` 参数接受使用下列格式的参数：

- '10G' - 包括大小为该值的磁盘。
- '10G:40G' - 包括大小在该范围内的磁盘。

- ':10G' - 包括大小小于或等于 10 GB 的磁盘。
- '40G:' - 包括大小等于或大于 40 GB 的磁盘。

例 13.1： 按磁盘大小匹配

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  size: '40TB:'
db_devices:
  size: ':2TB'
```

注意：需要引号

如果使用“:”分隔符，您需要用引号括住大小，否则“:”符号将被解释为新的配置哈希。

提示：单位缩写

您可以指定以兆字节 (M) 或太字节 (T) 为单位的大小，而不能指定以千兆字节 (G) 为单位的大小。

13.4.3.5 DriveGroups 示例

本节包含其他 OSD 设置的示例。

例 13.2： 简单设置

> 下面的示例说明了使用相同设置的两个节点：
>
> - 20 个 HDD

- 供应商：Intel
- 型号：SSD-123-foo
- 大小：4 TB

- 2 个 SSD
  - 供应商：Micron
  - 型号：MC-55-44-ZX
  - 大小：512 GB

对应的 `drive_groups.yml` 文件如下所示：

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  model: SSD-123-foo
db_devices:
  model: MC-55-44-XZ
```

这样的配置简单有效。但问题是管理员将来可能要添加来自不同供应商的磁盘，而这样的磁盘不在可添加范围内。您可以通过减少针对驱动器核心属性的过滤器来予以改进。

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  rotational: 1
db_devices:
  rotational: 0
```

在上面的示例中，我们强制将所有旋转设备声明为“data devices”，所有非旋转设备将被用作“shared devices”（wal、db）。

如果您知道大小超过 2 TB 的驱动器将始终充当较慢的数据设备，则可以按大小过滤：

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  size: '2TB:'
db_devices:
  size: ':2TB'
```

例 13.3：高级设置

下面的示例说明了两种不同的设置：20 个 HDD 将共享 2 个 SSD，而 10 个 SSD 将共享 2 个 NVMe。

- 20 个 HDD
  - 供应商：Intel
  - 型号：SSD-123-foo
  - 大小：4 TB
- 12 个 SSD
  - 供应商：Micron
  - 型号：MC-55-44-ZX
  - 大小：512 GB
- 2 个 NVMe
  - 供应商：Samsung
  - 型号：NVME-QQQQ-987
  - 大小：256 GB

这样的设置可使用如下两个布局定义：

```
service_type: osd
service_id: example_drvgrp_name
```

```
placement:
  host_pattern: '*'
data_devices:
  rotational: 0
db_devices:
  model: MC-55-44-XZ
```

```
service_type: osd
service_id: example_drvgrp_name2
placement:
  host_pattern: '*'
data_devices:
  model: MC-55-44-XZ
db_devices:
  vendor: samsung
  size: 256GB
```

例 13.4： 包含非一致节点的高级设置

上述示例假设所有节点的驱动器都相同，但情况并非总是如此：

节点 1-5：

- 20 个 HDD
  - 供应商：Intel
  - 型号：SSD-123-foo
  - 大小：4 TB
- 2 个 SSD
  - 供应商：Micron
  - 型号：MC-55-44-ZX
  - 大小：512 GB

节点 6-10：

- 5 个 NVMe

- 供应商：Intel
- 型号：SSD-123-foo
- 大小：4 TB

- 20 个 SSD
  - 供应商：Micron
  - 型号：MC-55-44-ZX
  - 大小：512 GB

您可以在布局中使用“target”键来定位特定节点。Salt 定位标记可让事情变得简单：

```
service_type: osd
service_id: example_drvgrp_one2five
placement:
  host_pattern: 'node[1-5]'
data_devices:
  rotational: 1
db_devices:
  rotational: 0
```

后接

```
service_type: osd
service_id: example_drvgrp_rest
placement:
  host_pattern: 'node[6-10]'
data_devices:
  model: MC-55-44-XZ
db_devices:
  model: SSD-123-foo
```

例 13.5： 专家设置

上述所有案例都假设 WAL 和 DB 使用相同设备，但也有可能会在专用设备上部署 WAL：

- 20 个 HDD

- 供应商：Intel
- 型号：SSD-123-foo
- 大小：4 TB

- 2 个 SSD
  - 供应商：Micron
  - 型号：MC-55-44-ZX
  - 大小：512 GB
- 2 个 NVMe
  - 供应商：Samsung
  - 型号：NVME-QQQQ-987
  - 大小：256 GB

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  model: MC-55-44-XZ
db_devices:
  model: SSD-123-foo
wal_devices:
  model: NVME-QQQQ-987
```

例 13.6：复杂（和不太可能的）设置

在以下设置中，我们尝试定义：

- 由 1 个 NVMe 支持 20 个 HDD
- 由 1 个 SSD (db) 和 1 个 NVMe (wal) 支持 2 个 HDD
- 由 1 个 NVMe 支持 8 个 SSD

- 2 个独立 SSD（加密）
- 1 个 HDD 作为备用，不应部署

共使用如下数量的驱动器：

- 23 个 HDD
  - 供应商：Intel
  - 型号：SSD-123-foo
  - 大小：4 TB
- 10 个 SSD
  - 供应商：Micron
  - 型号：MC-55-44-ZX
  - 大小：512 GB
- 1 个 NVMe
  - 供应商：Samsung
  - 型号：NVME-QQQQ-987
  - 大小：256 GB

DriveGroups 定义如下所示：

```
service_type: osd
service_id: example_drvgrp_hdd_nvme
placement:
  host_pattern: '*'
data_devices:
  rotational: 0
db_devices:
  model: NVME-QQQQ-987
```

```
service_type: osd
```

```
service_id: example_drvgrp_hdd_ssd_nvme
placement:
  host_pattern: '*'
data_devices:
  rotational: 0
db_devices:
  model: MC-55-44-XZ
wal_devices:
  model: NVME-QQQQ-987
```

```
service_type: osd
service_id: example_drvgrp_ssd_nvme
placement:
  host_pattern: '*'
data_devices:
  model: SSD-123-foo
db_devices:
  model: NVME-QQQQ-987
```

```
service_type: osd
service_id: example_drvgrp_standalone_encrypted
placement:
  host_pattern: '*'
data_devices:
  model: SSD-123-foo
encrypted: True
```

在文件的整个分析过程中，将始终保留一个 HDD。

13.4.4 删除 OSD

在从集群中删除 OSD 节点之前，请确认该集群中的可用磁盘空间是否比您要删除的 OSD 磁盘大。请注意，删除 OSD 会导致整个集群进行重新平衡。

1. 通过获取其 ID 来识别要删除的 OSD：

```
cephuser@adm > ceph orch ps --daemon_type osd
```

```
NAME   HOST            STATUS        REFRESHED  AGE  VERSION
osd.0  target-ses-090  running (3h)  7m ago     3h   15.2.7.689 ...
osd.1  target-ses-090  running (3h)  7m ago     3h   15.2.7.689 ...
osd.2  target-ses-090  running (3h)  7m ago     3h   15.2.7.689 ...
osd.3  target-ses-090  running (3h)  7m ago     3h   15.2.7.689 ...
```

2. 从集群中删除一个或多个 OSD：

```
cephuser@adm > ceph orch osd rm OSD1_ID OSD2_ID ...
```

例如：

```
cephuser@adm > ceph orch osd rm 1 2
```

3. 您可以查询删除操作的状态：

```
cephuser@adm > ceph orch osd rm status
OSD_ID  HOST         STATE                    PG_COUNT  REPLACE  FORCE
 STARTED_AT
2       cephadm-dev  done, waiting for purge  0         True     False
 2020-07-17 13:01:43.147684
3       cephadm-dev  draining                 17        False    True
 2020-07-17 13:01:45.162158
4       cephadm-dev  started                  42        False    True
 2020-07-17 13:01:45.162158
```

13.4.4.1 停止 OSD 删除

安排 OSD 删除后，您可以视需要停止删除。以下命令将重置 OSD 的初始状态并将其从队列中删除：

```
cephuser@adm > ceph orch osd rm stop OSD_SERVICE_ID
```

13.4.5 替换 OSD

有多种原因会导致您可能需要替换 OSD 磁盘。例如：

- 根据 SMART 信息，OSD 磁盘已发生故障或很快将发生故障，并且将不再能用来安全地存储数据。
- 您需要升级 OSD 磁盘，例如增加其大小。
- 您需要更改 OSD 磁盘布局。
- 您打算从非 LVM 布局转为基于 LVM 的布局。

要在保留其 ID 的情况下替换 OSD，请运行：

```
cephuser@adm > ceph orch osd rm OSD_SERVICE_ID --replace
```

例如：

```
cephuser@adm > ceph orch osd rm 4 --replace
```

替换 OSD 与删除 OSD 基本相同（有关更多详细信息，请参见第 13.4.4 节 “删除 OSD”），只不过 OSD 不是从 CRUSH 层次结构中永久删除，而是被指定了一个 `destroyed` 标志。

`destroyed` 标志用于确定将在下一次 OSD 部署期间重复使用的 OSD ID。新添加的符合 DriveGroups 规范的磁盘（有关更多详细信息，请参见第 13.4.3 节 “使用 DriveGroups 规范添加 OSD。”）将被指定其所替换磁盘的 OSD ID。

提示

追加 `--dry-run` 选项不会执行实际替换，而会预览通常会发生的步骤。

注意

如果发生故障后替换 OSD，我们强烈建议对归置组触发一次深层洗刷。有关详细信息，请参见第 17.6 节 “洗刷归置组”。

运行以下命令可启动深层洗刷：

```
cephuser@adm > ceph osd deep-scrub osd.OSD_NUMBER
```

重要：共享设备故障

如果 DB/WAL 的共享设备发生故障，您需要对共享该故障磁盘的所有 OSD 执行替换过程。

13.5 将 Salt 主控端移至新节点

如果需要用新 Salt 主控端主机替换 Salt 主控端主机，请执行以下步骤：

1. 导出集群配置并备份导出的 JSON 文件。有关详细信息，请参见《部署指南》，第 7 章 “使用 `ceph-salt` 部署引导集群”，第 7.2.14 节 “导出集群配置”。

2. 如果旧的 Salt 主控端也是集群中的唯一管理节点，则需要将 `/etc/ceph/ceph.client.admin.keyring` 和 `/etc/ceph/ceph.conf` 手动移至新的 Salt 主控端。

3. 停止并禁用旧的 Salt 主控端节点上的 Salt 主控端 `systemd` 服务：

```
root@master # systemctl stop salt-master.service
root@master # systemctl disable salt-master.service
```

4. 如果旧的 Salt 主控端节点不再位于集群中，还要停止并禁用 Salt 受控端 `systemd` 服务：

```
root@master # systemctl stop salt-minion.service
root@master # systemctl disable salt-minion.service
```

   #### 警告

   如果旧的 Salt 主控端节点上有任何正在运行的 Ceph 守护进程（MON、MGR、OSD、MDS、网关、监控），请不要停止或禁用 `salt-minion.service`。

5. 按照《部署指南》，第 5 章 “安装和配置 SUSE Linux Enterprise Server”中所述的过程在新的 Salt 主控端上安装 SUSE Linux Enterprise Server 15 SP3。

提示：转换 Salt 受控端

为便于将 Salt 受控端转换为新的 Salt 主控端，请从每个 Salt 受控端中删除原来的 Salt 主控端的公共密钥：

```
root@minion > rm /etc/salt/pki/minion/minion_master.pub
root@minion > systemctl restart salt-minion.service
```

6. 在新的 Salt 主控端上安装 `salt-master` 软件包和 `salt-minion` 软件包（如果适用）。

7. 在新的 Salt 主控端节点上安装 `ceph-salt`：

```
root@master # zypper install ceph-salt
root@master # systemctl restart salt-master.service
root@master # salt '*' saltutil.sync_all
```

重要

确保在继续之前运行所有三个命令。这些命令是幂等的；是否重复并不重要。

8. 在集群中包含新的 Salt 主控端，如《部署指南》，第 7 章 “使用 ceph-salt 部署引导集群”，第 7.1 节 “安装 ceph-salt”、《部署指南》，第 7 章 “使用 ceph-salt 部署引导集群”，第 7.2.2 节 “添加 Salt 受控端”和《部署指南》，第 7 章 “使用 ceph-salt 部署引导集群”，第 7.2.4 节 “指定管理节点”中所述。

9. 导入备份的集群配置并应用该配置：

```
root@master # ceph-salt import CLUSTER_CONFIG.json
root@master # ceph-salt apply
```

重要

导入之前，请在导出的 `CLUSTER_CONFIG.json` 文件中重命名 Salt 主控端的 `minion id`。

13.6 更新集群节点

定期应用滚动更新，以使 Ceph 集群节点保持最新。

13.6.1 软件源

使用最新的软件包增补集群之前，请确认集群的所有节点均可访问相关的软件源。有关所需软件源的完整列表，请参见《部署指南》, 第 10 章 “从 SUSE Enterprise Storage 6 升级到版本 7.1”, 第 10.1.5.1 节 “软件源”。

13.6.2 软件源暂存

如果您使用向集群节点提供软件源的暂存工具（例如 SUSE Manager、订阅管理工具或 RMT），请确认 SUSE Linux Enterprise Server 和 SUSE Enterprise Storage 的“更新”软件源的阶段都是在同一时刻创建的。

强烈建议您使用暂存工具来应用补丁级别为 `frozen` 或 `staged` 的补丁。这样可确保加入集群的新节点具有与已在集群中运行的节点相同的补丁级别。通过这种方法，您无需向集群的所有节点都应用最新补丁，新节点也能加入集群。

13.6.3 Ceph 服务停机时间

集群节点可能会在更新期间重引导，具体视配置而定。如果对象网关、Samba 网关、NFS Ganesha 或 iSCSI 等服务存在单一故障点，客户端计算机可能会暂时与相应节点正在重引导的服务断开连接。

13.6.4 运行更新

要将所有集群节点上的软件包更新到最新版本，请运行以下命令：

```
root@master # ceph-salt update
```

13.7 更新 Ceph

您可以指示 cephadm 将 Ceph 从一个 Bug 修复版本更新到另一个版本。Ceph 服务的自动更新遵循建议的顺序进行，即从 Ceph Manager、Ceph Monitor 开始更新，然后继续更新 Ceph OSD、元数据服务器和对象网关等其他服务。只有当 Ceph 指示集群将仍然可用之后，才会重启动每个守护进程。

注意

以下更新过程使用 **`ceph orch upgrade`** 命令。请注意，以下说明详细介绍了如何使用某个产品版本（例如维护更新）更新您的 Ceph 集群，但**未**提供如何将集群从一个产品版本升级到另一个产品版本的说明。

13.7.1 启动更新

开始更新之前，请确认所有节点当前都处于联机状态，并且集群是健康的：

```
cephuser@adm > cephadm shell -- ceph -s
```

更新到某个特定的 Ceph 版本：

```
cephuser@adm > ceph orch upgrade start --image REGISTRY_URL
```

例如：

```
cephuser@adm > ceph orch upgrade start --image registry.suse.com/ses/7.1/ceph/
ceph:latest
```

升级主机上的软件包：

```
cephuser@adm > ceph-salt update
```

13.7.2 监控更新

运行以下命令可确定是否正在进行更新：

```
cephuser@adm > ceph orch upgrade status
```

如果正在进行更新，您将在 Ceph 状态输出中看到一个进度条：

```
cephuser@adm > ceph -s
[...]
  progress:
    Upgrade to registry.suse.com/ses/7.1/ceph/ceph:latest (00h 20m 12s)
      [=======.....................] (time remaining: 01h 43m 31s)
```

您还可以查看 cephadm 日志：

```
cephuser@adm > ceph -W cephadm
```

13.7.3 取消更新

您可以随时停止更新过程：

```
cephuser@adm > ceph orch upgrade stop
```

13.8 停止或重引导集群

在某些情况下，可能需要停止或重引导整个集群。建议您仔细检查运行中服务的依赖项。下列步骤概要说明如何停止和启动集群：

1. 告知 Ceph 集群不要将 OSD 标记为 out：

   ```
   cephuser@adm > ceph osd set noout
   ```

2. 按下面的顺序停止守护进程和节点：

   1. 存储客户端
   2. 网关，例如 NFS Ganesha 或对象网关
   3. 元数据服务器

4. Ceph OSD
5. Ceph Manager
6. Ceph Monitor

3. 根据需要执行维护任务。
4. 以与关闭过程相反的顺序启动节点和服务器：
    1. Ceph Monitor
    2. Ceph Manager
    3. Ceph OSD
    4. 元数据服务器
    5. 网关，例如 NFS Ganesha 或对象网关
    6. 存储客户端
5. 删除 noout 标志：

```
cephuser@adm > ceph osd unset noout
```

13.9 删除整个 Ceph 集群

`ceph-salt purge` 命令可删除整个 Ceph 集群。如果部署了更多的 Ceph 集群，则将清除 **`ceph -s`** 报告的集群。这样，您便可以在测试不同的设置时清理集群环境。

为防止意外删除，编制流程会检查是否解除了安全措施。您可以通过运行以下命令来解除安全措施并删除 Ceph 集群：

```
root@master # ceph-salt disengage-safety
root@master # ceph-salt purge
```

14 Ceph 服务的操作

您可以在守护进程、节点或集群级别操作 Ceph 服务。根据您需要的方法，使用 cephadm 或 **`systemctl`** 命令。

14.1 操作单个服务

如果您需要操作单个服务，请先标识该服务：

```
cephuser@adm > ceph orch ps
NAME                                HOST        STATUS         REFRESHED  [...]
mds.my_cephfs.ses-min1.oterul       ses-min1    running (5d)   8m ago
mgr.ses-min1.gpijpm                 ses-min1    running (5d)   8m ago
mgr.ses-min2.oopvyh                 ses-min2    running (5d)   8m ago
mon.ses-min1                        ses-min1    running (5d)   8m ago
mon.ses-min2                        ses-min2    running (5d)   8m ago
mon.ses-min4                        ses-min4    running (5d)   7m ago
osd.0                               ses-min2    running (61m)  8m ago
osd.1                               ses-min3    running (61m)  7m ago
osd.2                               ses-min4    running (61m)  7m ago
rgw.myrealm.myzone.ses-min1.kwwazo  ses-min1    running (5d)   8m ago
rgw.myrealm.myzone.ses-min2.jngabw  ses-min2    error          8m ago
```

要标识特定节点上的服务，请运行：

```
ceph orch ps NODE_HOST_NAME
```

例如：

```
cephuser@adm > ceph orch ps ses-min2
NAME                                HOST      STATUS         REFRESHED
mgr.ses-min2.oopvyh                 ses-min2  running (5d)   3m ago
mon.ses-min2                        ses-min2  running (5d)   3m ago
osd.0                               ses-min2  running (67m)  3m ago
```

提示

ceph orch ps 命令支持多种输出格式。要更改格式，请追加 `--format FORMAT` 选项，其中 `FORMAT` 是 `json`、`json-pretty` 或 `yaml` 其中之一。例如：

```
cephuser@adm > ceph orch ps --format yaml
```

知道服务名称后，您就可以启动、重启动或停止该服务：

```
ceph orch daemon COMMAND SERVICE_NAME
```

例如，要重启动 ID 为 0 的 OSD 服务，请运行：

```
cephuser@adm > ceph orch daemon restart osd.0
```

14.2 操作服务类型

如果您需要操作整个 Ceph 集群中特定类型的服务，请使用以下命令：

```
ceph orch COMMAND SERVICE_TYPE
```

将 `COMMAND` 替换为 `start`、`stop` 或 `restart`。

例如，以下命令会重启动集群中的所有 MON，无论它们实际是在哪个节点上运行：

```
cephuser@adm > ceph orch restart mon
```

14.3 操作单个节点上的服务

通过使用 **systemctl** 命令，您可以操作单个节点上与 Ceph 有关的 `systemd` 服务和目标。

14.3.1 标识服务和目标

在操作与 Ceph 有关的 `systemd` 服务和目标之前，您需要标识其单元文件的文件名。服务的文件名具有以下格式：

```
ceph-FSID@SERVICE_TYPE.ID.service
```

例如：

```
ceph-b4b30c6e-9681-11ea-ac39-525400d7702d@mon.doc-ses-min1.service
```

```
ceph-b4b30c6e-9681-11ea-ac39-525400d7702d@rgw.myrealm.myzone.doc-ses-
min1.kwwazo.service
```

FSID

Ceph 集群的唯一 ID。您可以在 **`ceph fsid`** 命令的输出中找到该 ID。

SERVICE_TYPE

服务的类型，例如 `osd`、`mon` 或 `rgw`。

ID

服务的标识字符串。对于 OSD，它是服务的 ID 号。对于其他服务，它可以是节点的主机名，也可以是与服务类型相关的其他字符串。

提示

`SERVICE_TYPE.ID` 部分与 **`ceph orch ps`** 命令输出中的 `NAME` 列的内容相同。

14.3.2 操作一个节点上所有服务

通过使用 Ceph 的 `systemd` 目标，您可以同时操作某个节点上的**所有**服务，或者同时操作**属于**由其 `FSID` 标识的集群的所有服务。

例如，要停止某个节点上的所有 Ceph 服务，而不考虑服务所属的集群，请运行：

```
root@minion > systemctl stop ceph.target
```

要重启动属于 ID 为 `b4b30c6e-9681-11ea-ac39-525400d7702d` 的 Ceph 集群的所有服务，请运行：

```
root@minion > systemctl restart ceph-b4b30c6e-9681-11ea-ac39-525400d7702d.target
```

14.3.3 操作一个节点上的单个服务

标识特定服务的名称后，请如下所示操作该服务：

```
systemctl COMMAND SERVICE_NAME
```

例如，要重启动 ID 为 b4b30c6e-9681-11ea-ac39-525400d7702d 的集群上 ID 为 1 的单个 OSD 服务，请运行：

```
# systemctl restart ceph-b4b30c6e-9681-11ea-ac39-525400d7702d@osd.1.service
```

14.3.4 查询服务状态

可以查询 systemd 来了解服务的状态。例如：

```
# systemctl status ceph-b4b30c6e-9681-11ea-ac39-525400d7702d@osd.0.service
```

14.4 关闭并重启动整个 Ceph 集群

如果发生计划停电，可能需要关闭并重启动集群。要停止与 Ceph 有关的所有服务然后再重启动而不会出现问题，请按照以下步骤进行操作。

过程 14.1： 关闭整个 CEPH 集群

1. 关闭或断开访问集群的任何客户端。
2. 要阻止 CRUSH 自动重新平衡集群，请将集群设置为 noout：

   ```
   cephuser@adm > ceph osd set noout
   ```

3. 停止所有集群节点上的所有 Ceph 服务：

   ```
   root@master # ceph-salt stop
   ```

4. 关闭所有集群节点的电源：

   ```
   root@master # salt -G 'ceph-salt:member' cmd.run "shutdown -h"
   ```

过程 14.2： **启动整个 CEPH 集群**

1. 打开管理节点的电源。
2. 打开 Ceph Monitor 节点的电源。
3. 打开 Ceph OSD 节点的电源。
4. 取消设置之前设置的 noout 标志：

```
root@master # ceph osd unset noout
```

5. 打开所有已配置网关的电源。
6. 打开集群客户端的电源或连接集群客户端。

15 备份和恢复

本章说明您应当备份 Ceph 集群的哪些部分才能恢复集群功能。

15.1 备份集群配置和数据

15.1.1 备份 `ceph-salt` 配置

导出集群配置。有关详细信息，请参见《部署指南》, 第 7 章 “使用 `ceph-salt` 部署引导集群”, 第 7.2.14 节 “导出集群配置”。

15.1.2 备份 Ceph 配置

备份 `/etc/ceph` 目录。该目录包含至关重要的集群配置。例如，当您需要替换管理节点时，就需要备份 `/etc/ceph`。

15.1.3 备份 Salt 配置

您需要备份 `/etc/salt/` 目录。该目录包含 Salt 配置文件，例如 Salt 主控端密钥和已接受的客户端密钥。

从严格意义上来说，备份管理节点并不需要备份 Salt 文件，但这些文件能够简化 Salt 集群的重新部署。如果不备份这些文件，就需要在新管理节点上重新注册 Salt 受控端。

注意：Salt 主控端私用密钥的安全性

务必将 Salt 主控端私用密钥的备份存储在安全位置。Salt 主控端密钥可用于操纵所有集群节点。

15.1.4 备份自定义配置

- Prometheus 数据和自定义。
- Grafana 自定义。
- 手动更改 iSCSI 配置。
- Ceph 密钥。
- CRUSH 索引和 CRUSH 规则。通过运行以下命令将包含 CRUSH 规则的反编译 CRUSH 索引保存到 `crushmap-backup.txt` 中：

```
cephuser@adm > ceph osd getcrushmap | crushtool -d - -o crushmap-backup.txt
```

- Samba 网关配置。如果您使用的是单个网关，请备份 `/etc/samba/smb.conf`。如果您使用的是 HA 设置，还需要备份 CTDB 和 Pacemaker 配置文件。有关 Samba 网关所使用配置的详细信息，请参见第 24 章 “通过 Samba 导出 Ceph 数据”。
- NFS Ganesha 配置。仅当使用 HA 设置时需要备份。有关 NFS Ganesha 所使用配置的详细信息，请参见第 25 章 “NFS Ganesha”。

15.2 恢复 Ceph 节点

从备份中恢复节点的过程就是重新安装节点，替换其配置文件，然后重新编制集群，以便重新添加替换节点。

如果您需要重新部署管理节点，请参见第 13.5 节 “将 Salt 主控端移至新节点”。

对于受控端，通常更容易简化重构建和重新部署。

1. 重新安装节点。有关详细信息，请参见《部署指南》, 第 5 章 “安装和配置 SUSE Linux Enterprise Server”
2. 安装 Salt。有关详细信息，请参见《部署指南》, 第 6 章 “部署 Salt”
3. 从备份恢复 `/etc/salt` 目录后，启用并重启动适用的 Salt 服务，例如：

```
root@master # systemctl enable salt-master
root@master # systemctl start salt-master
```

```
root@master # systemctl enable salt-minion
root@master # systemctl start salt-minion
```

4. 从所有受控端中删除旧 Salt 主控端节点的公共主控端密钥。

```
root@master # rm /etc/salt/pki/minion/minion_master.pub
root@master # systemctl restart salt-minion
```

5. 恢复管理节点的所有本地内容。

6. 从之前导出的 JSON 文件导入集群配置。有关更多详细信息，请参见《部署指南》, 第 7 章 “使用 ceph-salt 部署引导集群”, 第 7.2.14 节 “导出集群配置”。

7. 应用导入的集群配置：

```
root@master # ceph-salt apply
```

16 监控和告警

在 SUSE Enterprise Storage 7.1 中，cephadm 会部署一个监控和告警堆栈。用户需要在 YAML 配置文件中定义要使用 cephadm 部署的服务（例如 Prometheus、告警管理器和 Grafana），也可以使用 CLI 来部署这些服务。当部署多个相同类型的服务时，会部署高可用性设置。但 Node Exporter 不适用于此规则。

使用 cephadm 可以部署以下监控服务：

- **Prometheus** 是监控和警告工具箱。它会收集 Prometheus 导出程序提供的数据，并在达到预定义阈值时触发预配置的告警。
- **Alertmanager** 用于处理 Prometheus 服务器发送的警告。它将负责删除重复信息、分组并将告警路由到正确的接收器。默认情况下，Ceph Dashboard 将自动配置为接收器。
- **Grafana** 是虚拟化和警告软件。此监控堆栈不使用 Grafana 的告警功能。而是使用告警管理器进行告警。
- **Node Exporter** 是 Prometheus 的导出程序，用于提供其所安装到的节点的相关数据。建议您在所有节点上安装 Node Exporter。

Prometheus Manager 模块提供有 Prometheus 导出程序，可从 `ceph-mgr` 中的收集点传递 Ceph 性能计数器。

Prometheus 配置（包括**抓取**目标（度量提供守护进程））由 cephadm 自动设置。cephadm 还会部署默认告警列表，例如 `health error`、`10% OSDs down` 或 `pgs inactive`。

默认情况下，传入 Grafana 的流量会使用 TLS 加密。您可以提供自己的 TLS 证书，也可以使用自我签名证书。如果在部署 Grafana 之前未配置自定义证书，则将自动为 Grafana 创建和配置自我签名证书。

可以通过执行以下步骤来为 Grafana 配置自定义证书：

1. 配置证书文件：

```
cephuser@adm >  ceph config-key set mgr/cephadm/grafana_key -i $PWD/key.pem
cephuser@adm >  ceph config-key set mgr/cephadm/grafana_crt -i $PWD/
certificate.pem
```

2. 重启动 Ceph Manager 服务：

```
cephuser@adm > ceph orch restart mgr
```

3. 重新配置 Grafana 服务，以反映新证书路径并为 Ceph Dashboard 设置正确的 URL：

```
cephuser@adm > ceph orch reconfig grafana
```

Alertmanager 处理 Prometheus 服务器发送的警告。它负责重复信息删除、分组，并将它们路由到正确的接收器。可以使用告警管理器消除告警，但也可以使用 Ceph Dashboard 管理静默功能。

建议您在所有节点上部署 `Node exporter`。可以使用含有 `node-exporter` 服务类型的 `monitoring.yaml` 文件执行此操作。有关部署服务的详细信息，请参见《部署指南》，第 8 章 “使用 cephadm 部署其余核心服务”，第 8.3.8 节 “部署监控堆栈”。

16.1 配置自定义或本地映像

提示

本节说明如何更改部署或更新服务时使用的容器映像的配置。不包括部署或重新部署服务所需的命令。

部署监控堆栈的推荐方法是按《部署指南》，第 8 章 “使用 cephadm 部署其余核心服务”，第 8.3.8 节 “部署监控堆栈”中所述应用其规范。

要部署自定义或本地容器映像，需要在 cephadm 中设置映像。为此，您需要运行以下命令：

```
cephuser@adm > ceph config set mgr mgr/cephadm/OPTION_NAME VALUE
```

其中，`OPTION_NAME` 为以下任何名称：

- container_image_prometheus
- container_image_node_exporter
- container_image_alertmanager
- container_image_grafana

如果未设置任何选项或已删除该设置，则将使用以下映像作为 VALUE：

- registry.suse.com/ses/7.1/ceph/prometheus-server:2.32.1
- registry.suse.com/ses/7.1/ceph/prometheus-node-exporter:1.1.2
- registry.suse.com/ses/7.1/ceph/prometheus-alertmanager:0.21.0
- registry.suse.com/ses/7.1/ceph/grafana:7.5.12

例如：

```
cephuser@adm > ceph config set mgr mgr/cephadm/container_image_prometheus prom/
prometheus:v1.4.1
```

注意

设置自定义映像将覆盖默认值（但并不会重写）。当有可用更新时，默认值将发生更改。设置自定义映像后，您将无法自动更新设置了自定义映像的组件。您需要手动更新配置（映像名称和标记）才能安装更新。

如果您选择遵循这些建议，则可以重置之前设置的自定义映像。之后，将再次使用默认值。使用 **ceph config rm** 重置配置选项：

```
cephuser@adm > ceph config rm mgr mgr/cephadm/OPTION_NAME
```

例如：

```
cephuser@adm > ceph config rm mgr mgr/cephadm/container_image_prometheus
```

16.2 更新监控服务

如第 16.1 节 “配置自定义或本地映像”中所述，cephadm 随附有推荐且经过测试的容器映像的 URL，默认会使用这些 URL。

更新 Ceph 软件包后，可能会随附这些 URL 的新版本。这只会更新容器映像的提取位置，而不会更新任何服务。

更新新容器映像的 URL（可以按第 16.1 节 “配置自定义或本地映像”中所述执行手动更新，也可以通过更新 Ceph 软件包进行自动更新）后，便可以更新监控服务。

要执行此操作，请使用 **ceph orch reconfig**，如下所示：

```
cephuser@adm > ceph orch reconfig node-exporter
cephuser@adm > ceph orch reconfig prometheus
cephuser@adm > ceph orch reconfig alertmanager
cephuser@adm > ceph orch reconfig grafana
```

当前没有可更新所有监控服务的单一命令。这些服务的更新顺序并不重要。

注意

如果您使用自定义容器映像，则在更新 Ceph 软件包时为监控服务指定的 URL 不会自动更改。如果您已指定自定义容器映像，则需要手动指定新容器映像的 URL。使用本地容器注册表时，可能会出现这种情况。

您可以在第 16.1 节 “配置自定义或本地映像”一节中找到要使用的推荐容器映像的 URL。

16.3 禁用监控

要禁用监控堆栈，请运行以下命令：

```
cephuser@adm > ceph orch rm grafana
cephuser@adm > ceph orch rm prometheus --force   # this will delete metrics data
 collected so far
cephuser@adm > ceph orch rm node-exporter
cephuser@adm > ceph orch rm alertmanager
cephuser@adm > ceph mgr module disable prometheus
```

16.4 配置 Grafana

Ceph Dashboard 后端要求 Grafana URL 能够在前端均衡加载 Grafana 仪表盘之前确认其是否存在。由于 Grafana 在 Ceph Dashboard 中的实施方式性质，意味着需要两个有效连接才能在 Ceph Dashboard 中看到 Grafana 图形：

- 后端（Ceph MGR 模块）需要确认请求的图形是否存在。如果此请求成功，便会告知前端它可以安全地访问 Grafana。
- 然后，前端会使用 `iframe` 直接从用户的浏览器请求 Grafana 图形。可以直接访问 Grafana 实例，而无需再经由 Ceph Dashboard。

现在，您的环境可能会使用户的浏览器难以直接访问 Ceph Dashboard 中配置的 URL。为解决此问题，可以配置一个单独的 URL，专用于告知前端（用户的浏览器）应该使用哪个 URL 来访问 Grafana。

要更改返回给前端的 URL，请发出以下命令：

```
cephuser@adm > ceph dashboard set-grafana-frontend-api-url GRAFANA-SERVER-URL
```

如果没有为该选项设置任何值，它将回退到 `GRAFANA_API_URL` 选项的值，该选项值自动设置并由 cephadm 定期更新。如果设置了值，它将指示浏览器使用此 URL 来访问 Grafana。

16.5 配置 Prometheus Manager 模块

Prometheus Manager 模块是一个 Ceph 内部模块，它扩展了 Ceph 的功能。该模块会从 Ceph 读取有关其状态和健康状况的（元）数据，并以可使用的格式向 Prometheus 提供（抓取的）数据。

注意

需要重启动 Prometheus Manager 模块才能应用配置更改。

16.5.1 配置网络接口

默认情况下，Prometheus Manager 模块接受主机上所有 IPv4 和 IPv6 地址的端口 9283 上的 HTTP 请求。端口和侦听地址都可使用 `ceph config-key set` 进行配置，并使用键 `mgr/prometheus/server_addr` 和 `mgr/prometheus/server_port`。此端口已在 Prometheus 的注册表中注册。

要更新 `server_addr`，请执行以下命令：

```
cephuser@adm > ceph config set mgr mgr/prometheus/server_addr 0.0.0.0
```

要更新 `server_port`，请执行以下命令：

```
cephuser@adm > ceph config set mgr mgr/prometheus/server_port 9283
```

16.5.2 配置 `scrape_interval`

Prometheus Manager 模块默认配置了15 秒的抓取间隔。我们建议使用的抓取间隔不少于 10 秒。要在 Prometheus 模块中设置其他抓取间隔，请将 `scrape_interval` 设置为所需值：

重要

要想正常运作而不会导致任何问题，此模块的 `scrape_interval` 应始终按照 Prometheus 抓取间隔进行设置。

```
cephuser@adm > ceph config set mgr mgr/prometheus/scrape_interval 15
```

16.5.3 配置缓存

在大型集群（超过 1000 个 OSD）上，提取度量的时间可能会变得非常重要。如果没有缓存，Prometheus Manager 模块可能会使管理器过载，并导致 Ceph Manager 实例无响应或崩溃。因此，系统默认会启用缓存，并且不能禁用缓存，但这意味着缓存可能会过时。当从 Ceph 提取度量的时间超过所配置的 `scrape_interval` 时，缓存将被视为过时。

如果发生此情况，系统将记录一条警报，并且模块将：

- 以 503 HTTP 状态码（服务不可用）响应。
- 返回缓存的内容，即使该内容可能已过时。

可使用 **`ceph config set`** 命令配置此行为。

要告知模块以可能过时的数据进行响应，请将其设置为 `return`：

```
cephuser@adm > ceph config set mgr mgr/prometheus/stale_cache_strategy return
```

要告知模块以 `service unavailable` 进行响应，请将其设置为 `fail`：

```
cephuser@adm > ceph config set mgr mgr/prometheus/stale_cache_strategy fail
```

16.5.4 启用 RBD 映像监控

Prometheus Manager 模块可以通过启用动态 OSD 性能计数器来选择性地收集 RBD 每个映像的 IO 统计数据。如此将为在 `mgr/prometheus/rbd_stats_pools` 配置参数中指定的存储池中的所有映像收集统计数据。

参数是以逗号或空格分隔的 `pool[/namespace]` 条目列表。如果未指定名称空间，则将收集存储池中所有名称空间的统计数据。

例如：

```
cephuser@adm > ceph config set mgr mgr/prometheus/rbd_stats_pools
 "pool1,pool2,poolN"
```

该模块会扫描指定的存储池和名称空间，列出所有可用映像，并定期刷新该列表。此间隔可通过 `mgr/prometheus/rbd_stats_pools_refresh_interval` 参数进行配置（以秒为单位），默认值为 300 秒（5 分钟）。

例如，如果您将同步间隔更改为 10 分钟：

```
cephuser@adm > ceph config set mgr mgr/prometheus/
rbd_stats_pools_refresh_interval 600
```

16.6 Prometheus 安全模型

Prometheus 的安全模型假设不受信任的用户有权访问 Prometheus HTTP 端点和日志。不受信任的用户有权访问数据库中所包含的 Prometheus 收集的所有（元）数据，以及各种操作和调试信息。

不过，Prometheus 的 HTTP API 仅限只读操作。无法使用 API 更改配置，并且不会公开秘密密钥。此外，Prometheus 还有一些内置的措施可用于缓解拒绝服务攻击的影响。

16.7 Prometheus 告警管理器 SNMP 网关

如果您想通过 SNMP 陷阱接收有关 Prometheus 告警的通知，可以通过 cephadm 或 Ceph Dashboard 安装 Prometheus 告警管理器 SNMP 网关。例如，如果要通过 SNMPv2c 接收此类通知，您需要创建包含以下内容的服务和归置规范文件：

注意

有关服务和归置文件的详细信息，请参见《部署指南》, 第 8 章 “使用 cephadm 部署其余核心服务”, 第 8.2 节 “服务和归置规范”。

```
service_type: snmp-gateway
service_name: snmp-gateway
placement:
    ADD_PLACEMENT_HERE
spec:
  credentials:
    snmp_community: ADD_COMMUNITY_STRING_HERE
  snmp_destination: ADD_FQDN_HERE:ADD_PORT_HERE
  snmp_version: V2c
```

或者，您可以使用 Ceph Dashboard 部署适用于 SNMPv2c 和 SNMPv3 的 SNMP 网关服务。有关详细信息，请参考 第 4.4 节 “显示服务”。

III 在集群中存储数据

17 存储的数据管理 153

18 管理存储池 184

19 纠删码存储池 204

20 RADOS 块设备 210

17 存储的数据管理

CRUSH 算法通过计算数据存储位置来确定如何存储和检索数据。使用 CRUSH，Ceph 客户端无需通过中心服务器或中介程序，即可直接与 OSD 通讯。借助算法确定的数据存储和检索方法，Ceph 可避免单一故障点、性能瓶颈和可伸缩性物理限制。

CRUSH 需要获取集群的索引，它使用 CRUSH 索引以伪随机的方式在 OSD 中存储和检索数据，并以一致的方式在整个集群中分布数据。

CRUSH 索引包含一个 OSD 列表、一个用于将设备聚合到物理位置的“存储桶”列表，以及一个告知 CRUSH 应如何复制 Ceph 集群存储池中数据的规则列表。通过反映安装的底层物理组织，CRUSH 可对相关设备故障的潜在根源建模，从而解决故障的根源。典型的根源包括物理接近、共用电源和共用网络。通过将这些信息编码到集群索引中，CRUSH 归置策略可将对象副本分隔在不同的故障域中，同时维持所需的分布方式。例如，为了消除可能的并发故障，可能需要确保数据副本位于使用不同机架、机柜、电源、控制器和/或物理位置的设备上。

部署 Ceph 集群后，将会生成默认的 CRUSH 索引。这种模式适合 Ceph 沙箱环境。但是，在部署大规模的数据集群时，强烈建议您考虑创建自定义 CRUSH 索引，因为这样做有助于管理 Ceph 集群、提高性能并确保数据安全。

例如，如果某个 OSD 停机，而您需要使用现场支持或更换硬件，则 CRUSH 索引可帮助您定位到发生 OSD 故障的主机所在的物理数据中心、机房、设备排和机柜。

同样，CRUSH 可以帮助您更快地确定故障。例如，如果特定机柜中的所有 OSD 同时停机，故障可能是由某个网络交换机或者机柜或网络交换机的电源所致，而不是发生在 OSD 自身上。

当与故障主机关联的归置组（请参见第 17.4 节 “归置组”）处于降级状态时，自定义 CRUSH 索引还可帮助您确定 Ceph 存储数据冗余副本的物理位置。

CRUSH 索引包括三个主要部分。

- OSD 设备包括与 `ceph-osd` 守护进程对应的任何对象存储设备。
- 存储桶由存储位置（例如设备排、机架、主机等）及所指定的权重以分层方式聚合而成。
- 规则集由存储桶选择方式组成。

17.1 OSD 设备

为了将归置组映射到 OSD，CRUSH 索引需要 OSD 设备（OSD 守护进程的名称）的列表。设备列表显示在 CRUSH 索引的最前面。

```
#devices
device NUM osd.OSD_NAME class CLASS_NAME
```

例如：

```
#devices
device 0 osd.0 class hdd
device 1 osd.1 class ssd
device 2 osd.2 class nvme
device 3 osd.3 class ssd
```

一般而言，一个 OSD 守护进程映射到一个磁盘。

17.1.1 设备类型

Ceph 的优势之一是 CRUSH 索引能够灵活控制数据归置。这也是集群最难管理的环节之一。**设备类型**会自动对 CRUSH 索引执行最常见的更改，以前这些更改都需要由管理员手动完成。

17.1.1.1 CRUSH 管理问题

Ceph 集群常由多种存储设备构建而成：HDD、SSD、NVMe，甚至是以上这些类型的混合。我们将这些不同的存储设备类型称为**设备类型**，以避免与 CRUSH 桶的**类型**属性（例如 host、rack 和 row，请参见第 17.2 节 “存储桶”了解更多详细信息）产生混淆。受 SSD 支持的 Ceph OSD 比受旋转磁盘支持的 OSD 速度快得多，因此更适合特定工作负载。借助 Ceph，您可轻松为不同数据集或工作负载创建 RADOS 存储池，以及指定不同的 CRUSH 规则来控制这些存储池的数据归置。

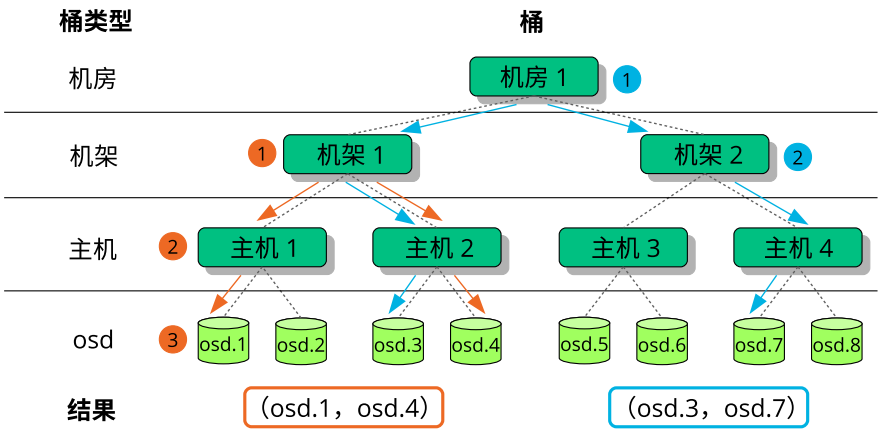


图 17.1： 具有混合设备类型的 OSD

不过，通过设置 CRUSH 规则将数据仅归置到特定类型的设备这个过程十分枯燥。规则是针对 CRUSH 层次结构运作的，但如果将某些设备混合到同一主机或机柜中（如上面的层次结构示例所示），则这些设备默认会混合在一起并显示在层次结构的同一子树中。要手动将它们分隔到单独的树中，需要针对先前版本的 SUSE Enterprise Storage 中的每个设备类型，为每个中间节点创建多个版本。

17.1.1.2 设备类型

Ceph 提供了一个较佳的解决方案，就是为每个 OSD 添加一个名为**设备类型**的属性。默认情况下，OSD 会根据 Linux 内核所公开的硬件属性，自动将自己的设备类型设置为“hdd”、“ssd”或“nvme”。**`ceph osd tree`** 命令输出的新列中会报告这些设备类型：

```
cephuser@adm > ceph osd tree
 ID CLASS WEIGHT   TYPE NAME      STATUS REWEIGHT PRI-AFF
 -1       83.17899 root default
 -4       23.86200     host cpach
 2   hdd  1.81898         osd.2      up  1.00000 1.00000
 3   hdd  1.81898         osd.3      up  1.00000 1.00000
 4   hdd  1.81898         osd.4      up  1.00000 1.00000
 5   hdd  1.81898         osd.5      up  1.00000 1.00000
 6   hdd  1.81898         osd.6      up  1.00000 1.00000
 7   hdd  1.81898         osd.7      up  1.00000 1.00000
 8   hdd  1.81898         osd.8      up  1.00000 1.00000
```

```
 15  hdd  1.81898         osd.15     up  1.00000 1.00000
 10  nvme 0.93100         osd.10     up  1.00000 1.00000
 0   ssd  0.93100         osd.0      up  1.00000 1.00000
 9   ssd  0.93100         osd.9      up  1.00000 1.00000
```

如果设备类型自动检测失败（例如由于未通过 `/sys/block` 正确向设备驱动程序公开设备的相关信息而导致失败），您可以使用命令行调整设备类型：

```
cephuser@adm > ceph osd crush rm-device-class osd.2 osd.3
done removing class of osd(s): 2,3
cephuser@adm > ceph osd crush set-device-class ssd osd.2 osd.3
set osd(s) 2,3 to class 'ssd'
```

17.1.1.3 设置 CRUSH 归置规则

CRUSH 规则可限制对特定设备类型执行归置操作。例如，您可以通过运行以下命令来创建仅将数据分布在 SSD 磁盘上的“fast”**副本**存储池：

```
cephuser@adm > ceph osd crush rule create-
replicated RULE_NAME ROOT FAILURE_DOMAIN_TYPE DEVICE_CLASS
```

例如：

```
cephuser@adm > ceph osd crush rule create-replicated fast default host ssd
```

创建名为“fast_pool”的存储池并将其分配给“fast”规则：

```
cephuser@adm > ceph osd pool create fast_pool 128 128 replicated fast
```

创建**纠删码**规则的过程略有不同。首先，创建包含所需设备类型对应属性的纠删码配置。然后，在创建纠删码存储池时使用该配置：

```
cephuser@adm > ceph osd erasure-code-profile set myprofile \
 k=4 m=2 crush-device-class=ssd crush-failure-domain=host
cephuser@adm > ceph osd pool create mypool 64 erasure myprofile
```

为应对您需要手动编辑 CRUSH 索引来自定义规则的情况，该语法已经过扩展，允许指定设备类型。例如，上述命令生成的 CRUSH 规则如下所示：

```
rule ecpool {
```

```
  id 2
  type erasure
  min_size 3
  max_size 6
  step set_chooseleaf_tries 5
  step set_choose_tries 100
  step take default class ssd
  step chooseleaf indep 0 type host
  step emit
}
```

此处的重要差别是“take”命令包含额外的“class `CLASS_NAME`”后缀。

17.1.1.4 其他命令

要列出 CRUSH 索引中使用的设备类型，请运行以下命令：

```
cephuser@adm > ceph osd crush class ls
[
  "hdd",
  "ssd"
]
```

要列出现有的 CRUSH 规则，请运行以下命令：

```
cephuser@adm > ceph osd crush rule ls
replicated_rule
fast
```

要查看名为“fast”的 CRUSH 规则的详细信息，请运行以下命令：

```
cephuser@adm > ceph osd crush rule dump fast
{
  "rule_id": 1,
  "rule_name": "fast",
  "ruleset": 1,
  "type": 1,
  "min_size": 1,
  "max_size": 10,
```

```
  "steps": [
      {
          "op": "take",
          "item": -21,
          "item_name": "default~ssd"
      },
      {
          "op": "chooseleaf_firstn",
          "num": 0,
          "type": "host"
      },
      {
          "op": "emit"
      }
  ]
}
```

要列出属于“ssd”类型的 OSD，请运行以下命令：

```
cephuser@adm > ceph osd crush class ls-osd ssd
0
1
```

17.1.1.5 从旧版 SSD 规则迁移到设备类型

在早于版本 5 的 SUSE Enterprise Storage 中，要写入适用于设备的规则，您需要手动编辑 CRUSH 索引并为每个具体的设备类型（例如 SSD）维护并行的层次结构。自 SUSE Enterprise Storage 5 起，设备类型特性透明地实现了这一目的。

您可以使用 **`crushtool`** 命令将旧版规则和层次结构转换为基于类型的新版规则。系统提供以下几种转换类型：

`crushtool --reclassify-root` `ROOT_NAME DEVICE_CLASS`

此命令会获取 `ROOT_NAME` 下的层次结构中的所有内容，并将通过

```
take ROOT_NAME
```

引用该根的所有规则调整为

```
take ROOT_NAME class DEVICE_CLASS
```

它会对存储桶重新编号，以便为指定类型的“shadow tree”使用旧 ID。因此，数据不会发生移动。

例 17.1：`crushtool --reclassify-root`

> 假设有以下现有规则：
>
> ```
> rule replicated_ruleset {
>    id 0
>    type replicated
>    min_size 1
>    max_size 10
>    step take default
>    step chooseleaf firstn 0 type rack
>    step emit
> }
> ```
>
> 如果您将根“default”重新分类为“hdd”类型，规则将变为
>
> ```
> rule replicated_ruleset {
>    id 0
>    type replicated
>    min_size 1
>    max_size 10
>    step take default class hdd
>    step chooseleaf firstn 0 type rack
>    step emit
> }
> ```

`crushtool --set-subtree-class` `BUCKET_NAME DEVICE_CLASS`

此方法会将根目录为 `BUCKET_NAME` 的子树中的每个设备标记为指定的设备类型。`--set-subtree-class` 通常与 `--reclassify-root` 选项结合使用，以确保该根中的所有设备均标为正确的类型。不过，其中一些设备可能特意使用了不同的类型，因此您不需要重新对它们进行标记。在此类情况下，请勿使用 `--set-subtree-class` 选项。请记住，这样的重新映射并不完美，因为之前的规则是跨多个类型的设备分发的，而调整后的规则将仅映射到指定设备类型的设备。

crushtool --reclassify-bucket MATCH_PATTERN DEVICE_CLASS DEFAULT_PATTERN

此方法允许将特定于类型的并行层次结构与常规层次结构合并。例如，许多用户会有类似以下的 CRUSH 索引：

例 17.2：**crushtool --reclassify-bucket**

```
host node1 {
   id -2           # do not change unnecessarily
   # weight 109.152
   alg straw
   hash 0  # rjenkins1
   item osd.0 weight 9.096
   item osd.1 weight 9.096
   item osd.2 weight 9.096
   item osd.3 weight 9.096
   item osd.4 weight 9.096
   item osd.5 weight 9.096
   [...]
}

host node1-ssd {
   id -10          # do not change unnecessarily
   # weight 2.000
   alg straw
   hash 0  # rjenkins1
   item osd.80 weight 2.000
   [...]
}

root default {
   id -1           # do not change unnecessarily
   alg straw
   hash 0  # rjenkins1
   item node1 weight 110.967
   [...]
}

root ssd {
```

```
    id -18          # do not change unnecessarily
    # weight 16.000
    alg straw
    hash 0  # rjenkins1
    item node1-ssd weight 2.000
    [...]
}
```

此函数会将与给定模式相匹配的每个存储桶重新分类。模式的格式可能为 `%suffix` 或 `prefix%`。在上面的示例中，您需要使用 `%-ssd` 模式。对于每个匹配的存储桶，与“%”通配符相匹配的名称的其余部分指定了基本存储桶。所匹配存储桶中的所有设备都会标记为指定的设备类型，随后会移至基本存储桶中。如果基本存储桶不存在（例如，如果“node12-ssd”存在，但“node12”不存在），则系统会创建基本存储桶，并将其关联到指定的默认父存储桶下。系统会为新的阴影存储桶保留旧的存储桶 ID，以防数据移动。系统会对包含引用旧存储桶的 `take` 步骤的规则进行调整。

`crushtool --reclassify-bucket` BUCKET_NAME DEVICE_CLASS BASE_BUCKET

您可以使用不包含通配符的 `--reclassify-bucket` 选项来映射单个存储桶。例如，在上面的示例中，我们希望将“ssd”存储桶映射到默认存储桶。
用于转换由上述片段组成的索引的最后一个命令将如下所示：

```
cephuser@adm > ceph osd getcrushmap -o original
cephuser@adm > crushtool -i original --reclassify \
  --set-subtree-class default hdd \
  --reclassify-root default hdd \
  --reclassify-bucket %-ssd ssd default \
  --reclassify-bucket ssd ssd default \
  -o adjusted
```

要确认转换是否正确，可以使用 `--compare` 选项。该选项会测试大量对 CRUSH 索引的输入，并比较是否会产生相同的结果。这些输入通过适用于 `--test` 的相同选项来控制。对于上面的示例，命令将如下所示：

```
cephuser@adm > crushtool -i original --compare adjusted
rule 0 had 0/10240 mismatched mappings (0)
rule 1 had 0/10240 mismatched mappings (0)
maps appear equivalent
```

提示

如果存在差异，括号中将会显示重新映射的输入比率。

如果您对调整后的 CRUSH 索引满意，便可将其应用于集群：

```
cephuser@adm > ceph osd setcrushmap -i adjusted
```

17.1.1.6 详细信息

有关 CRUSH 索引的更多详细信息，请参见第 17.5 节 “CRUSH 索引操作”。

有关 Ceph 存储池的更多一般详细信息，请参见第 18 章 “管理存储池”

有关纠删码存储池的更多详细信息，请参见第 19 章 “纠删码存储池”。

17.2 存储桶

CRUSH 索引包含 OSD 的列表，可将这些 OSD 组织成存储桶的树形结构排列形式，以便将设备聚合到物理位置。单个 OSD 构成树形结构中的树叶。

0	osd	特定的设备或 OSD（`osd.1`、`osd.2` 等）。
1	host	包含一个或多个 OSD 的主机的主机名。
2	chassis	机架中包含该 `host` 的机箱的标识符。
3	rack	计算机机柜。默认值为 `unknownrack`。
4	row	由一系列机柜组成的设备排。
5	pdu	“Power Distribution Unit”（电源分配单元）的缩写。
6	pod	“Point of Delivery”（分发点）的缩写。在此环境中为一组 PDU 或一组机架排。
7	room	包含多排机架的房间。

8	datacenter	包含一个或多个房间的物理数据中心。
9	region	全球的地理地区（例如 NAM、LAM、EMEA、APAC 等）
10	root	OSD 存储桶的树形结构根节点（通常设为 `default`）。

提示

您可以修改现有类型，以及创建自己的存储桶类型。

Ceph 的部署工具可生成 CRUSH 索引，其中包含每个主机的存储桶，以及名为“default”的根（可用于默认的 `rbd` 存储池）。剩余的存储桶类型提供了一种存储有关节点/存储桶的物理位置信息的方法，当 OSD、主机或网络硬件发生故障，并且管理员需要访问物理硬件时，这种方法可大大简化集群管理工作。

存储桶具有类型、唯一的名称（字符串）、以负整数表示的唯一 ID、相对于其项目总容量/权限的权重、存储桶算法（默认为 `straw2`）和哈希（默认为 `0`，代表 CRUSH 哈希 `rjenkins1`）。一个存储桶可以包含一个或多个项目。项目可由其他存储桶或 OSD 组成。项目可能会有一个权重来反映该项目的相对权重。

```
[bucket-type] [bucket-name] {
  id [a unique negative numeric ID]
  weight [the relative capacity/capability of the item(s)]
  alg [the bucket type: uniform | list | tree | straw2 | straw ]
  hash [the hash type: 0 by default]
  item [item-name] weight [weight]
}
```

下面的示例说明如何使用存储桶来聚合存储池，以及诸如数据中心、机房、机柜和设备排的物理位置。

```
host ceph-osd-server-1 {
        id -17
        alg straw2
        hash 0
        item osd.0 weight 0.546
        item osd.1 weight 0.546
}
```

```
row rack-1-row-1 {
        id -16
        alg straw2
        hash 0
        item ceph-osd-server-1 weight 2.00
}

rack rack-3 {
        id -15
        alg straw2
        hash 0
        item rack-3-row-1 weight 2.00
        item rack-3-row-2 weight 2.00
        item rack-3-row-3 weight 2.00
        item rack-3-row-4 weight 2.00
        item rack-3-row-5 weight 2.00
}

rack rack-2 {
        id -14
        alg straw2
        hash 0
        item rack-2-row-1 weight 2.00
        item rack-2-row-2 weight 2.00
        item rack-2-row-3 weight 2.00
        item rack-2-row-4 weight 2.00
        item rack-2-row-5 weight 2.00
}

rack rack-1 {
        id -13
        alg straw2
        hash 0
        item rack-1-row-1 weight 2.00
        item rack-1-row-2 weight 2.00
        item rack-1-row-3 weight 2.00
        item rack-1-row-4 weight 2.00
```

```
        item rack-1-row-5 weight 2.00
}

room server-room-1 {
        id -12
        alg straw2
        hash 0
        item rack-1 weight 10.00
        item rack-2 weight 10.00
        item rack-3 weight 10.00
}

datacenter dc-1 {
        id -11
        alg straw2
        hash 0
        item server-room-1 weight 30.00
        item server-room-2 weight 30.00
}

root data {
        id -10
        alg straw2
        hash 0
        item dc-1 weight 60.00
        item dc-2 weight 60.00
}
```

17.3 规则集

CRUSH 索引支持“CRUSH 规则”概念，这些规则确定存储池的数据归置。对于大型集群，您可能会创建许多存储池，其中每个存储池各自可能具有自己的 CRUSH 规则组和规则。默认 CRUSH 索引具有适用于默认根的规则。如果您想使用更多根和更多规则，则需稍后自行创建，或者在创建新存储池时让系统自动创建。

注意

大多数情况下，无需修改默认规则。创建新存储池时，该存储池的默认规则组为 0。

规则采用以下格式：

```
rule rulename {

        ruleset ruleset
        type type
        min_size min-size
        max_size max-size
        step step

}
```

ruleset

一个整数。将规则分类，使其属于一个规则组。通过在存储池中设置规则组来激活。必须指定此选项。默认值为 `0`。

type

一个字符串。描述了适用于“副本”或“纠删码”存储池的规则。必须指定此选项。默认值为 `replicated`。

min_size

一个整数。如果存储池组创建的副本数小于此数字，CRUSH 将不选择此规则。必须指定此选项。默认值为 `2`。

max_size

一个整数。如果存储池组创建的副本数大于此数字，CRUSH 将不选择此规则。必须指定此选项。默认值为 `10`。

step take `bucket`

采用以名称指定的存储桶，并开始在树中向下迭代。必须指定此选项。有关在树中迭代的说明，请参见第 17.3.1 节 “迭代节点树”。

step `targetmodenum` **type** `bucket-type`

`target` 可以是 `choose` 或 `chooseleaf`。如果设置为 `choose`，则会选择许多存储桶。`chooseleaf` 直接从存储桶集的每个存储桶的子树中选择 OSD（叶节点）。

`mode` 可以是 `firstn` 或 `indep`。请参见第 17.3.2 节 “firstn 和 indep”。

选择给定类型的存储桶的数量。其中，N 是可用选项的数量，如果 `num` > 0 且 < N，则选择该数量的存储桶；如果 `num` < 0，则表示 N - `num`；如果 `num` == 0，则选择 N 个存储桶（全部可用）。跟在 `step take` 或 `step choose` 后使用。

step emit

输出当前值并清空堆栈。通常在规则的末尾使用，但也可在同一规则中用来构成不同的树。跟在 `step choose` 后使用。

17.3.1 迭代节点树

可采用节点树的形式来查看使用存储桶定义的结构。在此树中，存储桶是节点，OSD 是叶。

CRUSH 索引中的规则定义如何从此树中选择 OSD。规则从某个节点开始，然后在树中向下迭代，以返回一组 OSD。无法定义需要选择哪个分支。CRUSH 算法可确保 OSD 集能够满足复制要求并均衡分布数据。

使用 `step take` `bucket` 时，节点树中的迭代从给定的存储桶（而不是存储桶类型）开始。如果要返回树中所有分支上的 OSD，该存储桶必须是根存储桶。否则，以下步骤只会在子树中迭代。

完成 `step take` 后，接下来会执行规则定义中的一个或多个 `step choose` 项。每个 `step choose` 项从前面选定的上层节点中选择定义数量的节点（或分支）。

最后，使用 `step emit` 返回选定的 OSD。

`step chooseleaf` 是一个便捷函数，可直接从给定存储桶的分支中选择 OSD。

图 17.2 “示例树”中提供了说明如何使用 `step` 在树中迭代的示例。在下面的规则定义中，橙色箭头和数字与 `example1a` 和 `example1b` 对应，蓝色箭头和数字与 `example2` 对应。

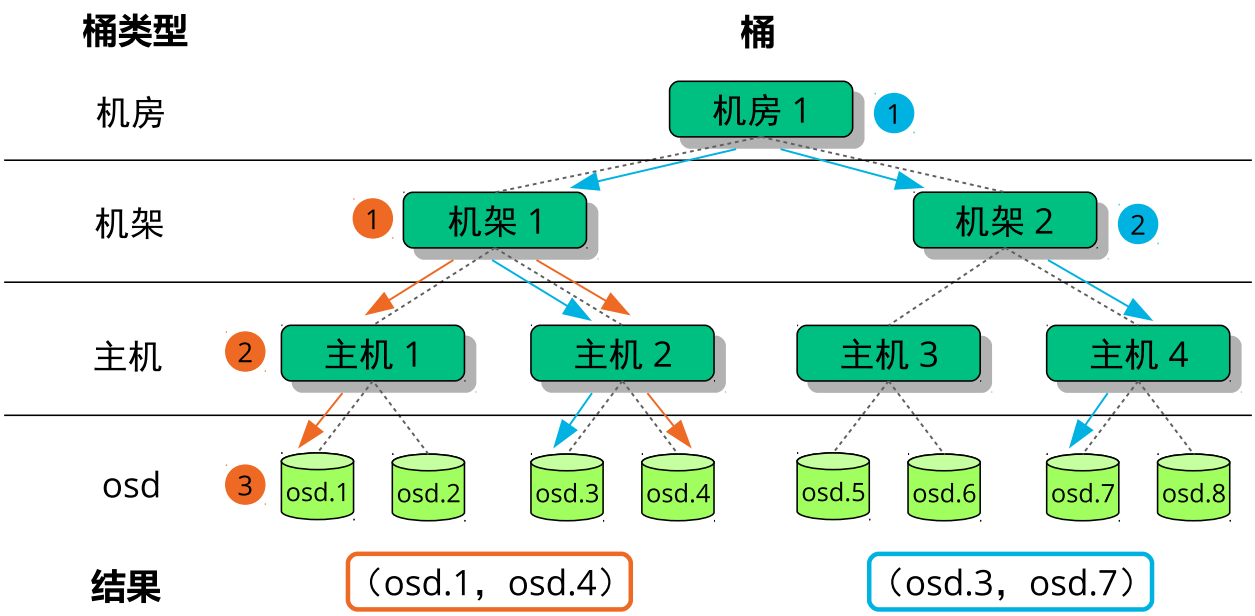


图 17.2： 示例树

```
# orange arrows
rule example1a {
        ruleset 0
        type replicated
        min_size 2
        max_size 10
        # orange (1)
        step take rack1
        # orange (2)
        step choose firstn 0 host
        # orange (3)
        step choose firstn 1 osd
        step emit
}

rule example1b {
        ruleset 0
        type replicated
        min_size 2
        max_size 10
        # orange (1)
        step take rack1
```

```
        # orange (2) + (3)
        step chooseleaf firstn 0 host
        step emit
}

# blue arrows
rule example2 {
        ruleset 0
        type replicated
        min_size 2
        max_size 10
        # blue (1)
        step take room1
        # blue (2)
        step chooseleaf firstn 0 rack
        step emit
}
```

17.3.2 firstn 和 indep

CRUSH 规则定义有故障节点或 OSD 的替换项（请参见第 17.3 节 “规则集”）。关键字 `step` 要求使用 `firstn` 或 `indep` 参数。图 17.3 “节点替换方法”提供了示例。

`firstn` 将替换节点添加到工作节点列表的末尾。如果某个节点发生故障，其后的正常节点会移位到左侧，以填充有故障节点留下的空缺。这是**副本存储池**的默认方法，也是需要采取的方法，因为次要节点已包含所有数据，因此可立即接管主要节点的职责。

`indep` 为每个工作节点选择固定的替换节点。替换有故障节点不会更改剩余节点的顺序。这对于**纠删码存储池**而言是所需的行为。在纠删码存储池中，节点上存储的数据取决于在选择节点时它所在的位置。如果节点的顺序发生变化，受影响节点上的所有数据都需要重新放置。

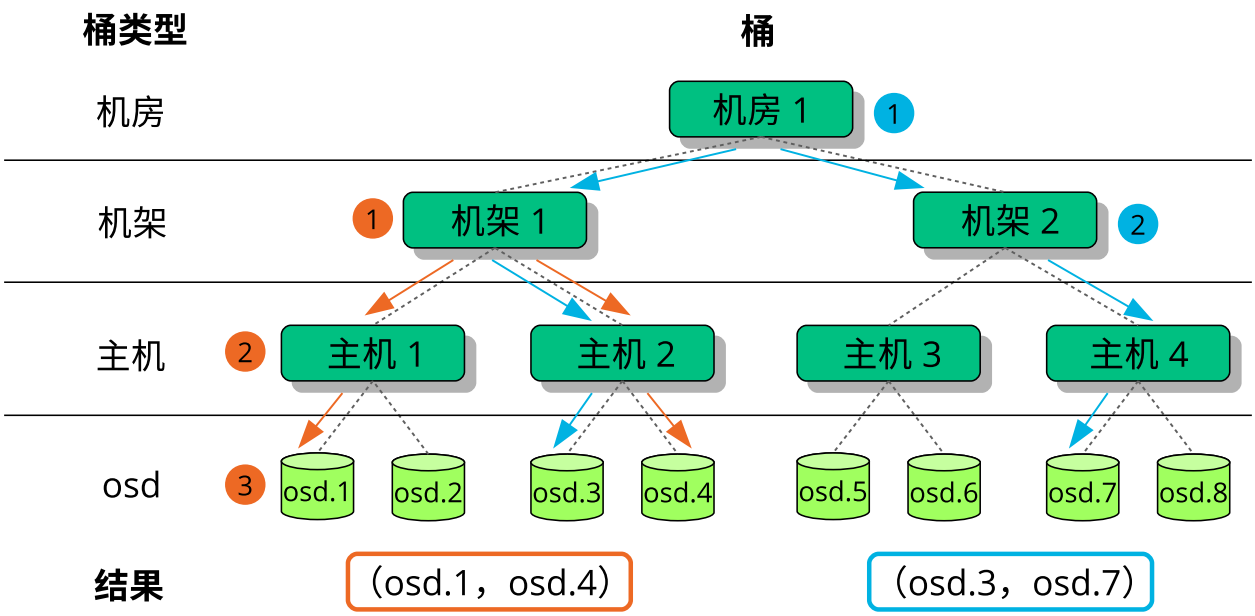


图 17.3： 节点替换方法

17.4 归置组

Ceph 会将对象映射到归置组 (PG)。归置组是指逻辑对象存储池的分片或片段，可将对象以组形式归置到 OSD 中。归置组可减少 Ceph 将数据存储到 OSD 中时每个对象的元数据量。如果归置组的数量较多（例如，每个 OSD 有 100 个归置组），将能实现较佳的平衡。

17.4.1 使用归置组

归置组 (PG) 聚合了存储池内的对象。主要原因是基于每个对象来跟踪对象归置和元数据的计算成本较高。例如，包含数百万个对象的系统无法直接跟踪各个对象的归置。

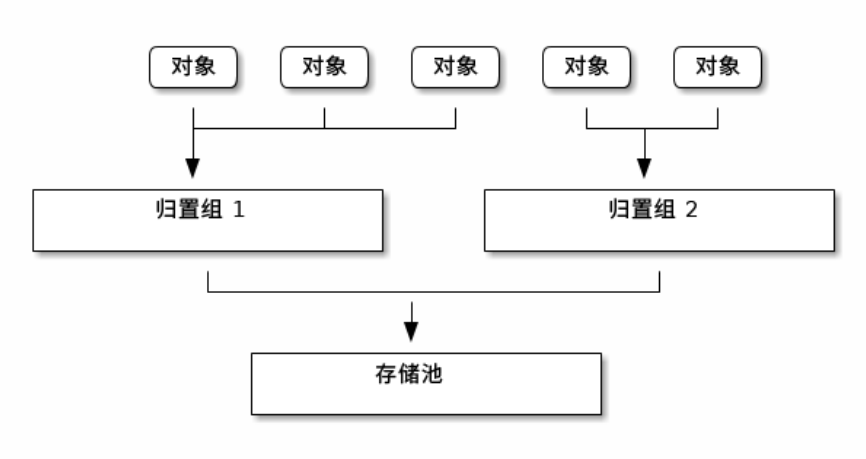


图 17.4： 存储池中的归置组

Ceph 客户端负责计算对象将属于哪个归置组。它会对对象 ID 进行哈希处理，并根据所定义存储池中的 PG 数及存储池的 ID 来应用操作。

归置组内的对象内容存储在一组 OSD 中。例如，在大小为 2 的副本存储池中，每个归置组将对象存储到两个 OSD 中：

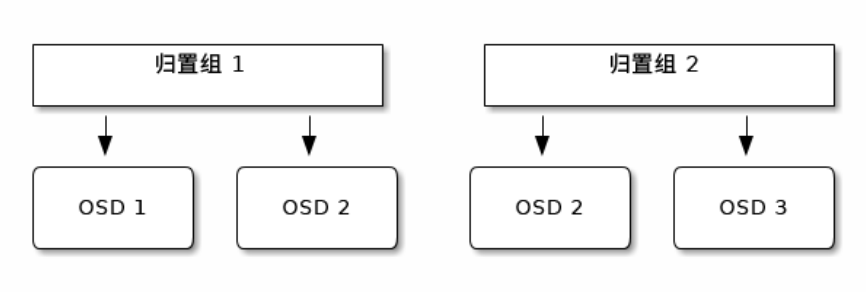


图 17.5： 归置组和 OSD

如果 OSD 2 发生故障，则系统会将另一个 OSD 指定给归置组 1，并在该 OSD 中填充 OSD 1 内所有对象的副本。如果存储池大小从 2 变为 3，则系统会向归置组另外再分配一个 OSD，并在该 OSD 中填充归置组内所有对象的副本。

归置组并不拥有 OSD，它们与同一存储池甚至其他存储池内的其他归置组共享 OSD。如果 OSD 2 发生故障，归置组 2 也将需要使用 OSD 3 来恢复对象的副本。

当归置组的数量增加时，系统会向新归置组分配 OSD。CRUSH 函数的结果也会发生更改，而且系统会将之前的归置组的部分对象复制到新归置组并将它们从旧归置组删除。

17.4.2 确定 PG_NUM 的值

注意

从 Ceph Nautilus (v14.x) 开始，您可以使用 Ceph Manager `pg_autoscaler` 模块根据需要自动扩展 PG。如果要启用此功能，请参考《Deploying and Administering SUSE Enterprise Storage with Rook》, 第 8 章 “Configuration”, 第 8.1.1.1 节 “Default PG and PGP counts”。

创建新存储池时，您仍可手动选择 PG_NUM 的值：

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

PG_NUM 无法自动计算得出。以下是一些常用的值，选择哪个值取决于集群中的 OSD 数量：

少于 5 个 OSD：

将 PG_NUM 设置为 128。

5 到 10 个 OSD：

将 PG_NUM 设置为 512。

10 到 50 个 OSD：

将 PG_NUM 设置为 1024。

随着 OSD 数量的增加，选择正确的 PG_NUM 值也变得愈加重要。PG_NUM 对集群的行为以及 OSD 发生故障时的数据持久性具有很大影响。

17.4.2.1 计算超过 50 个 OSD 的归置组

如果 OSD 数量低于 50，请使用第 17.4.2 节 “确定 PG_NUM 的值”中所述的预选值。如果 OSD 数量超过 50，建议为每个 OSD 使用约 50 到 100 个归置组，以平衡资源的使用、数据持久性和数据分布。对于单个对象存储池，您可以使用以下公式来计算基线数量：

```
total PGs = (OSDs * 100) / POOL_SIZE
```

其中，POOL_SIZE 是副本的个数（如果是副本存储池）或 **ceph osd erasure-code-profile get** 命令所返回的“k”和“m”之和（如果是纠删码存储池）。应将结果舍入到最接近的 2 的幂。对于 CRUSH 算法，建议进行舍入，以便均衡归置组之间的对象数。

例如，如果集群包含 200 个 OSD 和大小为 3 个副本的存储池，您需要按如下方式预估 PG 数：

```
          (200 * 100) / 3 = 6667
```

最接近的 2 的幂为 **8192**。

使用多个数据存储池存储对象时，您需要确保在每个存储池的归置组数与每个 OSD 的归置组数之间取得平衡。您需要采用合理的归置组总数，在不过度占用系统资源或导致互联过程过慢的前提下，确保每个 OSD 的差异保持在合理的较低水平。

例如，如果集群包含 10 个存储池，每个存储池有 512 个归置组（位于 10 个 OSD 中），则表示共有 5,120 个归置组分布于 10 个 OSD 中，即每个 OSD 有 512 个归置组。这样的设置不会使用过多资源。但是，如果创建了 1000 个存储池，且每个存储池有 512 个归置组，那么每个 OSD 需要处理约 50,000 个归置组，这样完成互联所需的资源和时间便会显著增加。

17.4.3 设置归置组数量

注意

从 Ceph Nautilus (v14.x) 开始，您可以使用 Ceph Manager pg_autoscaler 模块根据需要自动扩展 PG。如果要启用此功能，请参考《Deploying and Administering SUSE Enterprise Storage with Rook》, 第 8 章 “Configuration”, 第 8.1.1.1 节 “Default PG and PGP counts”。

如果您仍需手动指定存储池中的归置组数量，则需要在创建存储池时指定（请参见第 18.1 节 “创建存储池”）。为存储池设置归置组后，您可以运行以下命令来增加归置组的数量：

```
# ceph osd pool set POOL_NAME pg_num PG_NUM
```

增加归置组数量后，您还需要增加要归置的归置组数量 (PGP_NUM)，以便您的集群重新达到平衡。PGP_NUM 是考虑要通过 CRUSH 算法归置的归置组数。增加 PG_NUM 会分割归置组，但在增加 PGP_NUM 之前数据不会迁移到较新的归置组。PGP_NUM 应等于 PG_NUM。要增加应归置的归置组数量，请运行以下命令：

```
# ceph osd pool set POOL_NAME pgp_num PGP_NUM
```

17.4.4 获取归置组数量

要获取存储池中的归置组数量，请运行以下 **get** 命令：

```
# ceph osd pool get POOL_NAME pg_num
```

17.4.5 获取集群的 PG 统计数据

要获取集群内归置组的统计数据，请运行以下命令：

```
# ceph pg dump [--format FORMAT]
```

有效格式为“plain”（默认值）和“json”。

17.4.6 获取卡住的 PG 统计数据

要获取所有卡在指定状态的归置组的统计数据，请运行以下命令：

```
# ceph pg dump_stuck STATE \
 [--format FORMAT] [--threshold THRESHOLD]
```

可能的状态有：“inactive”（PG 由于在等待拥有最新数据的 OSD 恢复启用状态而无法处理读取或写入）、“unclean”（PG 包含未复制所需次数的对象）、“stale”（PG 处于未知状态，即托管 PG 的 OSD 未在 `mon_osd_report_timeout` 选项所指定的时间间隔内向 Monitor 集群报告相关信息）、“undersized”或“degraded”。

有效格式为“plain”（默认值）和“json”。

该阈值定义归置组至少卡住多少秒（默认为 300 秒）后系统会将其包含到返回的统计数据中。

17.4.7 搜索归置组索引

要搜索特定归置组的索引，请运行以下命令：

```
# ceph pg map PG_ID
```

Ceph 将返回归置组索引、归置组和 OSD 状态：

```
# ceph pg map 1.6c
osdmap e13 pg 1.6c (1.6c) -> up [1,0] acting [1,0]
```

17.4.8 检索归置组统计数据

要检索特定归置组的统计数据，请运行以下命令：

```
# ceph pg PG_ID query
```

17.4.9 洗刷归置组

要洗刷（第 17.6 节 “洗刷归置组”）归置组，请运行以下命令：

```
# ceph pg scrub PG_ID
```

Ceph 会检查主节点和副本节点、生成归置组内所有对象的编目，并对它们进行比较，以确保没有缺少或不匹配的对象且对象的内容均一致。假设所有副本均匹配，最后的语义整理便可确保所有与快照相关的对象元数据均一致。错误通过日志来报告。

17.4.10 设置归置组回填和恢复的优先顺序

您可能会遇到数个归置组需要恢复和/或回填，而其中一些归置组存储的数据比其他归置组更为重要的情况。例如，这些 PG 可能存储着运行中计算机所使用的映像数据，其他 PG 则可能存储的是由非活跃计算机使用的数据或相关度较低的数据。在该情况下，您可能需要优先恢复这些归置组，以便更早地恢复存储在这些归置组中的数据的性能和可用性。要将特定归置组标记为在回填或恢复过程中优先处理，请运行以下命令：

```
# ceph pg force-recovery PG_ID1 [PG_ID2 ... ]
# ceph pg force-backfill PG_ID1 [PG_ID2 ... ]
```

这将导致 Ceph 先对指定归置组执行恢复或回填，然后再处理其他归置组。这并不会中断当前正在进行的回填或恢复，而是会使指定的 PG 尽快得到处理。如果您改变了主意或将错误的归置组设为优先处理，请使用以下命令取消优先顺序：

```
# ceph pg cancel-force-recovery PG_ID1 [PG_ID2 ... ]
# ceph pg cancel-force-backfill PG_ID1 [PG_ID2 ... ]
```

`cancel-*` 命令会删除 PG 的“force”标志，使其按默认顺序接受处理。同样，这并不会影响当前正在处理的归置组，只会影响仍在排队的归置组。完成归置组恢复或回填后，系统即会自动清除“force”标志。

17.4.11 还原丢失的对象

如果集群丢失了一个或多个对象，而您已决定放弃搜索丢失的数据，则需要将未找到的对象标记为“lost”。

如果在查询过所有可能的位置后仍未找到这些对象，您可能需要放弃这些丢失的对象。这可能是由于几种故障同时发生（这种情况很少见）导致的，致使集群在写入本身恢复之前便得知写入已执行。

目前唯一支持的选项为“revert”，该选项会回滚到对象的先前版本，或在有新对象时完全忽略丢失的对象。要将“未找到”的对象标记为“lost”，请运行以下命令：

```
cephuser@adm > ceph pg PG_ID mark_unfound_lost revert|delete
```

17.4.12 启用 PG 自动扩展器

归置组 (PG) 是 Ceph 分布数据方式的详细内部实现。通过启用 PG 自动扩展功能，您可以允许集群根据集群的使用情况创建或自动调整 PG。

系统中的每个存储池都有一个 `pg_autoscale_mode` 属性，可将其设为 `off`、`on` 或 `warn`：

自动扩展器按存储池配置，可在以下三种模式下运行：

off

为此存储池禁用自动扩展功能。由管理员为每个存储池选择合适的 PG 数量。

on

为给定存储池启用 PG 计数自动调整功能。

warn

一旦应调整 PG 计数，系统将发出健康状况告警。

要为现有存储池设置自动扩展模式，请运行以下命令：

```
cephuser@adm > ceph osd pool set POOL_NAME pg_autoscale_mode mode
```

您也可以运行以下命令，以配置将应用于日后创建的所有存储池的默认 `pg_autoscale_mode`：

```
cephuser@adm > ceph config set global osd_pool_default_pg_autoscale_mode MODE
```

您可以运行以下命令来查看每个存储池、它的相关使用情况，以及对 PG 计数的任何更改建议：

```
cephuser@adm > ceph osd pool autoscale-status
```

17.5 CRUSH 索引操作

本节介绍基本的 CRUSH 索引操作方法，例如编辑 CRUSH 索引、更改 CRUSH 索引参数，以及添加/移动/删除 OSD。

17.5.1 编辑 CRUSH 索引

要编辑现有的 CRUSH 索引，请执行以下操作：

1. 获取 CRUSH 索引。要获取集群的 CRUSH 索引，请执行以下命令：

   ```
   cephuser@adm > ceph osd getcrushmap -o compiled-crushmap-filename
   ```

   Ceph 会将编译的 CRUSH 索引输出 (`-o`) 到您指定名称的文件。由于该 CRUSH 索引采用编译格式，您必须先将其反编译，然后才能对其进行编辑。

2. 反编译 CRUSH 索引。要反编译 CRUSH 索引，请执行以下命令：

```
cephuser@adm > crushtool -d compiled-crushmap-filename \
 -o decompiled-crushmap-filename
```

Ceph 将对已编译的 CRUSH 索引进行反编译 (`-d`)，并将其输出 (`-o`) 到您指定名称的文件。

3. 至少编辑“设备”、“存储桶”和“规则”中的其中一个参数。
4. 编译 CRUSH 索引。要编译 CRUSH 索引，请执行以下命令：

```
cephuser@adm > crushtool -c decompiled-crush-map-filename \
 -o compiled-crush-map-filename
```

Ceph 会将编译的 CRUSH 索引存储到您指定名称的文件。

5. 设置 CRUSH 索引。要设置集群的 CRUSH 索引，请执行以下命令：

```
cephuser@adm > ceph osd setcrushmap -i compiled-crushmap-filename
```

Ceph 将输入您所指定文件名的已编译 CRUSH 索引，作为集群的 CRUSH 索引。

提示：使用版本控制系统

请为导出并修改过的 CRUSH 索引文件使用 git 或 svn 这样的版本控制系统。这可以让可能发生的回滚变得简单。

提示：测试新 CRUSH 索引

请使用 **`crushtool --test`** 命令测试经过调整的新 CRUSH 索引，并与应用新 CRUSH 索引之前的状态进行比较。以下命令开关十分有用：`--show-statistics`、`--show-mappings`、`--show-bad-mappings`、`--show-utilization`、`--show-utilization-all`、`--show-choose-tries`

17.5.2 添加或移动 OSD

要在运行中集群的 CRUSH 索引中添加或移动 OSD，请执行以下命令：

```
cephuser@adm > ceph osd crush set id_or_name weight root=pool-name
bucket-type=bucket-name ...
```

id

一个整数。OSD 的数字 ID。必须指定此选项。

name

一个字符串。OSD 的全名。必须指定此选项。

weight

一个双精度值。OSD 的 CRUSH 权重。必须指定此选项。

root

一个键/值对。默认情况下，CRUSH 层次结构包含 default 存储池作为根。必须指定此选项。

bucket-type

键/值对。可在 CRUSH 层次结构中指定 OSD 的位置。

下面的示例将 `osd.0` 添加到层次结构，或移动之前某个位置的 OSD。

```
cephuser@adm > ceph osd crush set osd.0 1.0 root=data datacenter=dc1 room=room1
 \
row=foo rack=bar host=foo-bar-1
```

17.5.3 `ceph osd reweight` 与 `ceph osd crush reweight` 之间的差异

有两个相似的命令都可更改 Ceph OSD 的权重。它们的使用情境不同，可能会造成混淆。

17.5.3.1 `ceph osd reweight`

用法：

```
cephuser@adm > ceph osd reweight OSD_NAME NEW_WEIGHT
```

ceph osd reweight 用于对 Ceph OSD 设置覆盖权重。此值介于 0 到 1 之间，会强制 CRUSH 重新放置将以其他方式存储于此驱动器上的数据。该命令**不会**更改为 OSD 上方的存储桶指定的权重，它是一种在正常 CRUSH 分布出现问题时的纠正措施。例如，如果您的其中一个 OSD 处于 90%，而其他 OSD 处于 40%，则您可缩小此权重，以尝试对其进行补偿。

注意：OSD 权重是暂时的

请注意，**ceph osd reweight** 并非永久性设置。当某个 OSD 被标记为“out”时，它的权重会被设置为 0，当它一旦重新被标记为“in”，其权重又会更改为 1。

17.5.3.2 ceph osd crush reweight

用法：

```
cephuser@adm > ceph osd crush reweight OSD_NAME NEW_WEIGHT
```

ceph osd crush reweight 用于设置 OSD 的 **CRUSH** 权重。此权重可以是任意值（通常是以 TB 为单位的磁盘大小），用于控制系统尝试分配给 OSD 的数据量。

17.5.4 删除 OSD

要从运行中集群的 CRUSH 索引中删除 OSD，请执行以下命令：

```
cephuser@adm > ceph osd crush remove OSD_NAME
```

17.5.5 添加存储桶

要向运行中集群的 CRUSH 索引添加某个存储桶，请执行 **ceph osd crush add-bucket** 命令：

```
cephuser@adm > ceph osd crush add-bucket BUCKET_NAME BUCKET_TYPE
```

17.5.6 移动存储桶

要将某个存储桶移到 CRUSH 索引层次结构中的不同位置，请执行以下命令：

```
cephuser@adm > ceph osd crush move BUCKET_NAME BUCKET_TYPE=BUCKET_NAME [...]
```

例如：

```
cephuser@adm > ceph osd crush move bucket1 datacenter=dc1 room=room1 row=foo
 rack=bar host=foo-bar-1
```

17.5.7 删除存储桶

要从 CRUSH 索引层次结构中删除某个存储桶，请执行以下命令：

```
cephuser@adm > ceph osd crush remove BUCKET_NAME
```

注意：仅限空存储桶

从 CRUSH 层次结构中只能删除空存储桶。

17.6 洗刷归置组

除了为对象创建多个副本外，Ceph 还可通过**洗刷**归置组来确保数据完整性（请参见《部署指南》，第 1 章 “SES 和 Ceph”，第 1.3.2 节 “归置组”了解有关归置组的详细信息）。Ceph 的洗刷类似于在对象存储层运行 **fsck**。对于每个归置组，Ceph 都会生成一个包含所有对象的编目，并比较每个主对象及其副本，以确保不会有缺失或不匹配的对象。每天的浅层洗刷会检查对象大小和属性，而每周的深层洗刷则会读取数据并使用校验和来确保数据完整性。

洗刷对于维护数据完整性非常重要，但该操作可能会降低性能。您可以通过调整以下设置来增加或减少洗刷操作：

`osd max scrubs`

同时针对一个 Ceph OSD 执行的洗刷操作数量上限。默认值为 1。

`osd scrub begin hour`、`osd scrub end hour`

按小时定义的一天内可以执行洗刷的时间段（0 到 24）。默认开始时间为 0，结束时间为 24。

> **重要**
>
> 如果归置组的洗刷间隔超出 `osd scrub max interval` 设置的值，则无论您定义了何种洗刷时间段，都将执行洗刷。

`osd scrub during recovery`

允许恢复期间执行洗刷。如果将此选项设置为“false”，则当存在工作的恢复进程时，将禁止安排新的洗刷。已在运行的洗刷将继续执行。此选项有助于降低忙碌集群上的负载。默认值为“true”。

`osd scrub thread timeout`

洗刷线程超时前的最长时间（以秒为单位）。默认值为 60。

`osd scrub finalize thread timeout`

洗刷完成线程超时前的最长时间（以秒为单位）。默认值为 60*10。

`osd scrub load threshold`

规范化的最大负载。当系统负载（由 `getloadavg()` 与 `online cpus` 数量之比定义）高于此数字时，Ceph 将不会执行洗刷。默认值为 0.5。

`osd scrub min interval`

当 Ceph 集群负载较低时洗刷 Ceph OSD 的最短间隔（以秒为单位）。默认值为 60*60*24（一天一次）。

`osd scrub max interval`

无论集群负载如何都洗刷 Ceph OSD 的最长间隔（以秒为单位）。默认值为 7*60*60*24（一周一次）。

`osd scrub chunk min`

单次操作期间要洗刷的对象存储块数量下限。洗刷期间，Ceph 会阻止向单个块写入数据。默认值为 5。

`osd scrub chunk max`

单次操作期间要洗刷的对象存储块数量上限。默认值为 25。

`osd scrub sleep`

洗刷下一组块之前休眠的时间。增大此值会降低整个洗刷操作的速度，但对客户端操作的影响较小。默认值为 0。

`osd deep scrub interval`

深层洗刷（完整读取所有数据）的间隔。`osd scrub load threshold` 选项不会影响此设置。默认值为 60*60*24*7（一周一次）。

`osd scrub interval randomize ratio`

在安排归置组的下一次洗刷作业时，为 `osd scrub min interval` 值增加一个随机延迟。该延迟为一个随机的值，小于 `osd scrub min interval` * `osd scrub interval randomized ratio` 所得结果。因此，该默认设置实际上是将洗刷随机地安排在允许的时间段 [1, 1.5] * `osd scrub min interval` 内执行。默认值为 0.5。

`osd deep scrub stride`

执行深层洗刷时读取的大小。默认值为 524288 (512 kB)。

18 管理存储池

Ceph 将数据存储在存储池中。存储池是用于存储对象的逻辑组。如果您先部署集群而不创建存储池，Ceph 会使用默认存储池来存储数据。下面着重指出与 Ceph 存储池有关的重要特性：

- **恢复能力**：Ceph 存储池通过复制或编码其中包含的数据来提供恢复能力。可将每个存储池设置为 `replicated` 或 `erasure coding`。对于副本存储池，您可以进一步设置存储池中每个数据对象将拥有的副本数。可丢失的副本数（OSD、CRUSH 存储桶/叶）比副本数少一个。您可以使用纠删码设置 `k` 和 `m` 的值，其中 `k` 是数据块的数量，`m` 是编码块的数量。对于纠删码存储池，编码块的数量决定了在不丢失数据的情况下可丢失的 OSD（CRUSH 存储桶/树叶）数量。
- **归置组**：您可以设置存储池的归置组数。典型配置为每个 OSD 使用约 100 个归置组，以提供最佳平衡而又不会耗费太多计算资源。设置多个存储池时，请务将存储池和集群作为整体考虑，确保设置合理的归置组数。
- **CRUSH 规则**：在存储池中存储数据时，系统会根据映射到该存储池的 CRUSH 规则集来放置对象及其副本或块（如果是纠删码存储池）。您可为存储池创建自定义 CRUSH 规则。
- **快照**：使用 **`ceph osd pool mksnap`** 创建快照时，可高效创建特定存储池的快照。

要将数据组织到存储池中，可以列出、创建和删除存储池。您还可以查看每个存储池的用量统计数字。

18.1 创建存储池

可以创建 `replicated`（用于保留对象的多个副本，以便从丢失的 OSD 恢复）或 `erasure`（用于获得通用 RAID 5 或 6 功能）类型的存储池。副本存储池所需的原始存储空间较多，而纠删码存储池所需的原始存储空间较少。默认设置为 `replicated`。有关纠删码存储池的详细信息，请参见第 19 章 “纠删码存储池”。

要创建副本存储池，请执行以下命令：

```
cephuser@adm > ceph osd pool create POOL_NAME
```

注意

自动扩展器负责处理剩余的可选参数。有关详细信息，请参见第 17.4.12 节 “启用 PG 自动扩展器”。

要创建纠删码存储池，请执行以下命令：

```
cephuser@adm > ceph osd pool create POOL_NAME erasure CRUSH_RULESET_NAME \
EXPECTED_NUM_OBJECTS
```

如果超出每个 OSD 的归置组限制，则 **`ceph osd pool create`** 命令可能会失败。该限制通过 `mon_max_pg_per_osd` 选项设置。

POOL_NAME

存储池的名称，必须唯一。必须指定此选项。

POOL_TYPE

存储池类型，可以是 `replicated`（用于保留对象的多个副本，以便从失败的 OSD 恢复）或 `erasure`（用于获得某种通用 RAID5 功能）。副本池需要的原始存储较多，但可实现所有 Ceph 操作。纠删码存储池需要的原始存储较少，但只能实现部分可用的操作。默认 `POOL_TYPE` 设置为 `replicated`。

CRUSH_RULESET_NAME

此存储池的 CRUSH 规则集的名称。如果所指定的规则集不存在，则创建副本存储池的操作将会失败，并显示 -ENOENT。对于副本存储池，它是由 `osd pool default CRUSH replicated ruleset` 配置变量指定的规则集。此规则集必须存在。对于纠删码存储池，如果使用默认纠删码配置，则规则集为“erasure-code”，否则为 `POOL_NAME`。如果此规则集尚不存在，系统将隐式创建该规则集。

erasure_code_profile=profile

仅适用于纠删码存储池。使用纠删码配置。该配置必须是 **`osd erasure-code-profile set`** 所定义的现有配置。

注意

如果出于任何原因禁用了存储池上的自动扩展器（`pg_autoscale_mode` 设为 off），您可以手动计算和设置 PG 数量。有关计算存储池的适当归置组数的详细信息，请参见第 17.4 节 “归置组”。

EXPECTED_NUM_OBJECTS

此存储池的预期对象数。如果设置此值（与一个为负数的 `filestore merge threshold` 值），系统在创建存储池时会分割 PG 文件夹。这可避免因运行时文件夹拆分导致的延迟影响。

18.2 列出存储池

要列出集群的存储池，请执行以下命令：

```
cephuser@adm > ceph osd pool ls
```

18.3 重命名存储池

要重命名存储池，请执行以下命令：

```
cephuser@adm > ceph osd pool rename CURRENT_POOL_NAME NEW_POOL_NAME
```

如果重命名了存储池，且为经过身份验证的用户使用了按存储池功能，则必须用新的存储池名称更新用户的功能。

18.4 删除存储池

警告：删除存储池的操作不可逆

存储池中可能包含重要数据。删除存储池会导致存储池中的所有数据消失，且无法恢复。

不小心删除存储池十分危险，因此 Ceph 实施了两个机制来防止删除存储池。要删除存储池，必须先禁用这两个机制。

第一个机制是 NODELETE 标志。每个存储池都有这个标志，其默认值是“false”。要确定某个存储池的此标志值，请运行以下命令：

```
cephuser@adm > ceph osd pool get pool_name nodelete
```

如果命令输出 `nodelete: true`，则只有在使用以下命令更改该标志后，才能删除存储池：

```
cephuser@adm > ceph osd pool set pool_name nodelete false
```

第二个机制是集群范围的配置参数 `mon allow pool delete`，其默认值为“false”。这表示默认不能删除存储池。显示的错误消息是：

```
Error EPERM: pool deletion is disabled; you must first set the
mon_allow_pool_delete config option to true before you can destroy a pool
```

若要规避此安全设置删除存储池，可以临时将 `mon allow pool delete` 设置为“true”，删除存储池，然后将该参数恢复为“false”：

```
cephuser@adm > ceph tell mon.* injectargs --mon-allow-pool-delete=true
cephuser@adm > ceph osd pool delete pool_name pool_name --yes-i-really-really-
mean-it
cephuser@adm > ceph tell mon.* injectargs --mon-allow-pool-delete=false
```

`injectargs` 命令会显示以下消息：

```
injectargs:mon_allow_pool_delete = 'true' (not observed, change may require
 restart)
```

这主要用于确认该命令已成功执行。它不是错误。

如果为您创建的存储池创建了自己的规则组和规则，则应该考虑在不再需要该存储池时删除规则组和规则。

18.5 其他操作

18.5.1 将存储池与应用关联

在使用存储池之前，需要将它们与应用关联。将与 CephFS 搭配使用或由对象网关自动创建的存储池会自动关联。

对于其他情况，可以手动将自由格式的应用名称与存储池关联：

```
cephuser@adm > ceph osd pool application enable POOL_NAME APPLICATION_NAME
```

提示：默认应用名称

CephFS 使用应用名称 `cephfs`，RADOS 块设备使用 `rbd`，对象网关使用 `rgw`。

一个存储池可以与多个应用关联，每个应用都可具有自己的元数据。要列出与存储池关联的一个（或多个）应用，请发出以下命令：

```
cephuser@adm > ceph osd pool application get pool_name
```

18.5.2 设置存储池配额

您可以设置存储池配额，限定每个存储池的最大字节数和/或最大对象数。

```
cephuser@adm > ceph osd pool set-
quota POOL_NAME MAX_OBJECTS OBJ_COUNT MAX_BYTES BYTES
```

例如：

```
cephuser@adm > ceph osd pool set-quota data max_objects 10000
```

要删除配额，请将其值设置为 0。

18.5.3 显示存储池统计数据

要显示存储池的用量统计数字，请执行以下命令：

```
cephuser@adm > rados df
 POOL_NAME                    USED OBJECTS CLONES COPIES MISSING_ON_PRIMARY
 UNFOUND DEGRADED  RD_OPS      RD  WR_OPS      WR USED COMPR UNDER COMPR
 .rgw.root                 768 KiB       4      0     12                  0
  0        0      44  44 KiB       4   4 KiB        0 B         0 B
 cephfs_data               960 KiB       5      0     15                  0
  0        0    5502 2.1 MiB      14  11 KiB        0 B         0 B
 cephfs_metadata           1.5 MiB      22      0     66                  0
  0        0      26  78 KiB     176 147 KiB        0 B         0 B
 default.rgw.buckets.index     0 B       1      0      3                  0
  0        0       4   4 KiB       1     0 B        0 B         0 B
 default.rgw.control           0 B       8      0     24                  0
  0        0       0     0 B       0     0 B        0 B         0 B
 default.rgw.log               0 B     207      0    621                  0
  0        0 5372132 5.1 GiB 3579618     0 B        0 B         0 B
 default.rgw.meta          961 KiB       6      0     18                  0
  0        0     155 140 KiB      14   7 KiB        0 B         0 B
 example_rbd_pool          2.1 MiB      18      0     54                  0
  0        0 3350841 2.7 GiB     118  98 KiB        0 B         0 B
 iscsi-images              769 KiB       8      0     24                  0
  0        0 1559261 1.3 GiB      61  42 KiB        0 B         0 B
 mirrored-pool             1.1 MiB      10      0     30                  0
  0        0  475724 395 MiB      54  48 KiB        0 B         0 B
 pool2                         0 B       0      0      0                  0
  0        0       0     0 B       0     0 B        0 B         0 B
 pool3                     333 MiB      37      0    111                  0
  0        0 3169308 2.5 GiB   14847 118 MiB        0 B         0 B
 pool4                     1.1 MiB      13      0     39                  0
  0        0 1379568 1.1 GiB   16840  16 MiB        0 B         0 B
```

各列的说明如下：

USED

存储池使用的字节数。

OBJECTS

存储池中存储的对象数。

CLONES

存储池中存储的克隆数。如果在创建快照时某个客户端向对象写入数据，系统将会创建原始对象的克隆，而不是对该对象进行修改，这样便不会修改已截取快照的原始对象内容。

COPIES

对象副本的数量。例如，如果某个复制系数为 3 的副本存储池有“x”个对象，它通常将会有 3 * x 个副本。

MISSING_ON_PRIMARY

当主 OSD 上的副本缺失时，处于降级状态（不是所有副本都存在）的对象的数量。

UNFOUND

未找到的对象数。

DEGRADED（已降级）

已降级的对象数。

RD_OPS

针对此存储池请求的读取操作总数。

RD

从此存储池读取的字节总数。

WR_OPS

针对此存储池请求的写入操作总数。

WR

写入存储池的字节总数。请注意，该数值与存储池的用量不同，因为您可能会多次写入同一对象。如此一来，存储池的用量虽然不变，但写入存储池的字节数会增长。

USED COMPR

为压缩数据分配的字节数。

UNDER COMPR

压缩数据在未压缩时占用的字节数。

18.5.4 获取存储池值

要从存储池中获取值，请运行以下 **get** 命令：

```
cephuser@adm > ceph osd pool get POOL_NAME KEY
```

您可以获取第 18.5.5 节 “设置存储池值”中所列键以及下列键的值：

PG_NUM

存储池的归置组数。

PGP_NUM

计算数据归置时要使用的归置组的有效数量。有效范围等于或小于 PG_NUM。

提示：存储池的所有值

要列出与特定存储池相关的所有值，请运行以下命令：

```
cephuser@adm > ceph osd pool get POOL_NAME all
```

18.5.5 设置存储池值

要设置存储池的值，请执行以下命令：

```
cephuser@adm > ceph osd pool set POOL_NAME KEY VALUE
```

以下是按存储池类型排序的存储池值列表：

通用存储池值

crash_replay_interval

允许客户端重放已确认但未提交的请求的秒数。

pg_num

存储池的归置组数。如果您将新 OSD 添加到集群，请确认所有存储池上专用于新 OSD 的归置组的值。

pgp_num

计算数据归置时要使用的归置组的有效数量。

crush_ruleset

用于在集群中映射对象归置的规则组。

hashpspool

为给定存储池设置 (1) 或取消设置 (0) HASHPSPOOL 标志。启用此标志会更改算法，以采用更佳的方式将 PG 分配到 OSD 之间。对之前 HASHPSPOOL 标志设为默认值 0 的存储池启用此标志后，集群会开始回填，以使所有 PG 恢复正确归置状态。请注意，此操作可能会在集群上产生大量 I/O 负载，因此请勿对负载较高的生产集群启用该标志（由 0 更改为 1）。

nodelete

防止删除存储池。

nopgchange

防止更改存储池的 `pg_num` 和 `pgp_num`。

noscrub、nodeep-scrub

禁用（深层）洗刷特定存储池的数据以解决临时高 I/O 负载问题。

write_fadvise_dontneed

在给定存储池的读/写请求上设置或取消设置 `WRITE_FADVISE_DONTNEED` 标志，以绕过将数据放入缓存。默认为 `false`。适用于副本存储池和 EC 存储池。

scrub_min_interval

集群负载低时洗刷存储池的最小间隔（秒）。默认值 `0` 表示使用来自 Ceph 配置文件的 `osd_scrub_min_interval` 值。

scrub_max_interval

不论集群负载如何都洗刷存储池的最大间隔（秒）。默认值 `0` 表示使用来自 Ceph 配置文件的 `osd_scrub_max_interval` 值。

deep_scrub_interval

深层洗刷存储池的间隔（秒）。默认值 `0` 表示使用来自 Ceph 配置文件的 `osd_deep_scrub` 值。

副本存储池值

size

设置存储池中对象的副本数。有关更多详细信息，请参见第 18.5.6 节 “设置对象副本数”。仅用于副本存储池。

min_size

设置 I/O 所需的最小副本数。有关更多详细信息，请参见第 18.5.6 节 “设置对象副本数”。仅用于副本存储池。

nosizechange

防止更改存储池的大小。创建存储池时，默认值取自 `osd_pool_default_flag_nosizechange` 参数的值，默认设为 `false`。仅适用于副本存储池，因为无法更改 EC 存储池的大小。

hit_set_type

对缓存池启用命中集跟踪。请参见布隆过滤器 (http://en.wikipedia.org/wiki/Bloom_filter) 以了解更多信息。此选项可用的值如下：`bloom`、`explicit_hash`、`explicit_object`。默认值为 `bloom`，其他值仅用于测试。

hit_set_count

要为缓存池存储的命中集数。该数值越高，`ceph-osd` 守护进程耗用的 RAM 越多。默认值为 `0`。

hit_set_period

缓存池的命中集期间的时长（以秒为单位）。该数值越高，`ceph-osd` 守护进程耗用的 RAM 越多。创建存储池时，默认值取自 `osd_tier_default_cache_hit_set_period` 参数的值，默认设为 `1200`。仅适用于副本存储池，因为 EC 存储池不能用作缓存层。

hit_set_fpp

布隆命中集类型的误报率。请参见布隆过滤器 (http://en.wikipedia.org/wiki/Bloom_filter) 以了解更多信息。有效范围是 0.0 - 1.0，默认值为 `0.05`

use_gmt_hitset

为缓存分层创建命中集时，强制 OSD 使用 GMT（格林威治标准时间）时戳。这可确保在不同时区中的节点返回相同的结果。默认值为 `1`。不应该更改此值。

cache_target_dirty_ratio

在缓存分层代理将已修改（脏）对象刷新到后备存储池之前，包含此类对象的缓存池百分比。默认值为 `0.4`。

cache_target_dirty_high_ratio

在缓存分层代理将已修改（脏）对象刷新到速度更快的后备存储池之前，包含此类对象的缓存池百分比。默认值为 `0.6`。

cache_target_full_ratio

在缓存分层代理将未修改（正常）对象从缓存池赶出之前，包含此类对象的缓存池百分比。默认值为 `0.8`。

target_max_bytes

触发 `max_bytes` 阈值后，Ceph 将会开始刷新或赶出对象。

target_max_objects

触发 `max_objects` 阈值时，Ceph 将开始刷新或赶出对象。

hit_set_grade_decay_rate

两次连续的 `hit_set` 之间的温度降低率。默认值为 `20`。

hit_set_search_last_n

计算温度时在 `hit_set` 中对出现的项最多计 `N` 次。默认值为 `1`。

cache_min_flush_age

在缓存分层代理将对象从缓存池刷新到存储池之前的时间（秒）。

cache_min_evict_age

在缓存分层代理将对象从缓存池中赶出之前的时间（秒）。

纠删码存储池值

fast_read

如果对纠删码存储池启用此标志，则读取请求会向所有分片发出子读取命令，并一直等到接收到足够解码的分片，才会为客户端提供服务。对于 **jerasure** 和 **isa** 纠删插件，前 `K` 个副本返回时，就会使用从这些副本解码的数据立即处理客户端的请求。采用此方法会产生较高的 CPU 负载，而磁盘/网络负载则较低。目前，此标志仅支持用于纠删码存储池。默认值为 `0`。

18.5.6 设置对象副本数

要设置副本存储池上的对象副本数，请执行以下命令：

```
cephuser@adm > ceph osd pool set poolname size num-replicas
```

`num-replicas` 包括对象本身。例如，如果您想用对象和对象的两个副本组成对象的三个实例，请指定 3。

警告：设置的副本不能少于 3 个

如果将 `num-replicas` 设置为 2，数据将只有**一个**副本。例如，如果一个对象实例发生故障，则需要在恢复期间确定自上次洗刷后，另一个副本没有损坏（有关详细信息，请参见第 17.6 节 “洗刷归置组”）。

将存储池设置为具有一个副本意味着存储池中的数据对象只有**一个**实例。如果 OSD 发生故障，您将丢失数据。如果要短时间存储临时数据，可能就会用到只有一个副本的存储池。

提示：设置 3 个以上的副本

为存储池设置 4 个副本可将可靠性提高 25%。

如果有两个数据中心，您至少需要为存储池设置 4 个副本，使每个数据中心都有两个副本。如此，当其中一个数据中心发生故障时，仍有两个副本存在，并且如果又有一个磁盘发生故障，您仍可确保不会丢失数据。

注意

对象可以接受降级模式下副本数量低于 `pool size` 的 I/O。要设置 I/O 所需副本的最小数目，应该使用 `min_size` 设置。例如：

```
cephuser@adm > ceph osd pool set data min_size 2
```

这可确保数据池中没有对象会接收到副本数量低于 `min_size` 的 I/O。

提示：获取对象副本数

要获取对象副本数，请执行以下命令：

```
cephuser@adm > ceph osd dump | grep 'replicated size'
```

Ceph 将列出存储池，并高亮显示 `replicated size` 属性。Ceph 默认会创建对象的两个副本（共三个副本，或者大小为 3）。

18.6 存储池迁移

创建存储池（请参见第 18.1 节 “创建存储池”）时，您需要指定存储池的初始参数，例如存储池类型或归置组数量。如果您稍后决定更改其中的任一参数（例如将副本存储池转换为纠删码存储池，或者减少归置组数量），您需要将存储池数据迁移到其参数适合您的部署的另一个存储池。

本节介绍两种迁移方法：用于常规存储池数据迁移的**缓存层**方法，以及使用 **`rbd migrate`** 子命令将 RBD 映像迁移到新存储池的方法。每种方法都包含特定设置和限制。

18.6.1 限制

- 您可以使用**缓存层**方法从副本存储池迁移到 EC 存储池或另一个副本存储池。不支持从 EC 存储池迁移。
- 您无法将 RBD 映像和 CephFS 导出项从副本存储池迁移到纠删码存储池。原因是 EC 存储池不支持 `omap`，而 RBD 和 CephFS 会使用 `omap` 存储其元数据。例如，系统将无法刷新 RBD 的报头对象。但您可以将数据迁移到 EC 存储池，而将元数据保留在副本存储池中。
- 使用 **`rbd migration`** 方法可确保在迁移映像时最大限度地减少客户端停机时间。您只需在 `prepare` 步骤之前停止客户端，并在此后再将其启动即可。请注意，只有支持此特性的 `librbd` 客户端（Ceph Nautilus 或更新版本）才能在 `prepare` 步骤之后打开映像，而较旧的 `librbd` 客户端或 `krbd` 客户端在执行 `commit` 步骤之前都无法打开映像。

18.6.2 使用缓存层迁移

该方法的原理十分简单，只需将需要迁移的存储池按相反的顺序加入缓存层中即可。下面是将名为 “testpool” 的副本存储池迁移到纠删码存储池的示例：

过程 18.1： 将副本存储池迁移到纠删码存储池

1. 创建一个名为“newpool”的新纠删码存储池。有关存储池创建参数的详细说明，请参见第 18.1 节 “创建存储池”。

```
cephuser@adm > ceph osd pool create newpool erasure default
```

确认使用的客户端密钥环提供的针对“newpool”的用户权限至少与“testpool”的相同。

您现在有两个池，即装满数据的原始副本存储池“testpool”和新的空纠删码存储池“newpool”：

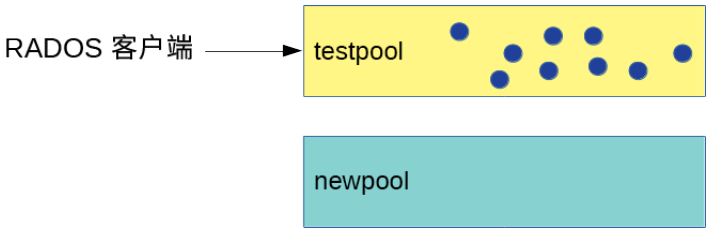


图 18.1： 迁移前的存储池

2. 设置缓存层，并将副本存储池“testpool”配置为缓存池。通过使用 `-force-nonempty` 选项，即使存储池已有数据，您也可以添加缓存层：

```
cephuser@adm > ceph tell mon.* injectargs \
 '--mon_debug_unsafe_allow_tier_with_nonempty_snaps=1'
cephuser@adm > ceph osd tier add newpool testpool --force-nonempty
cephuser@adm > ceph osd tier cache-mode testpool proxy
```

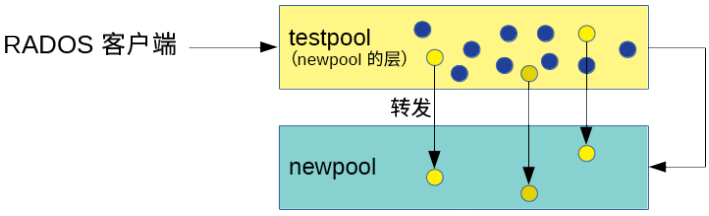


图 18.2： 缓存层设置

3. 强制缓存池将所有对象移到新池中：

```
cephuser@adm > rados -p testpool cache-flush-evict-all
```

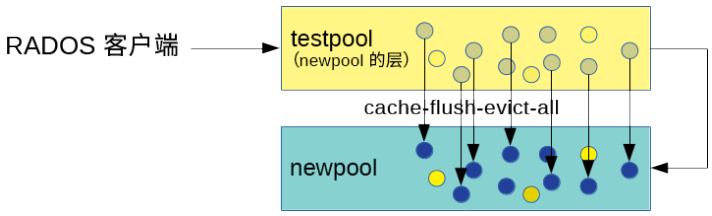


图 18.3： 数据刷新

4. 您需要指定一个覆盖层，以便在旧池中搜索对象，直到所有数据都已刷新到新的纠删码存储池。

```
cephuser@adm > ceph osd tier set-overlay newpool testpool
```

有了覆盖层，所有操作都会转到旧的副本存储池“testpool”：

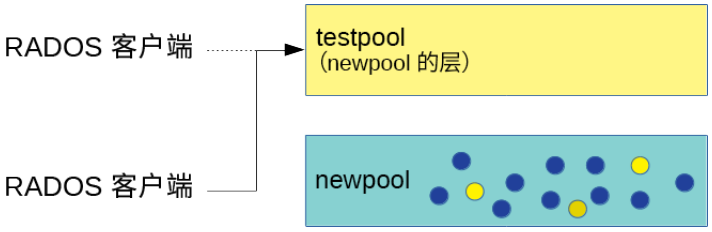


图 18.4： 设置覆盖层

现在，您可以将所有客户端都切换为访问新池中的对象。

5. 所有数据都迁移到纠删码存储池“newpool”后，删除覆盖层和旧超速缓冲池“testpool”：

```
cephuser@adm > ceph osd tier remove-overlay newpool
cephuser@adm > ceph osd tier remove newpool testpool
```

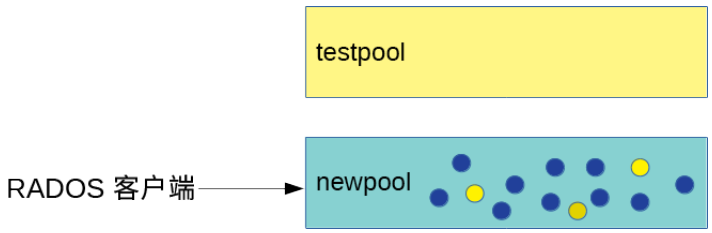


图 18.5： 迁移完成

6. 运行

```
cephuser@adm > ceph tell mon.* injectargs \
 '--mon_debug_unsafe_allow_tier_with_nonempty_snaps=0'
```

18.6.3 迁移 RBD 映像

建议采用如下方式将 RBD 映像从一个副本存储池迁移到另一个副本存储池。

1. 禁止客户端（例如虚拟机）访问 RBD 映像。
2. 在目标存储池中创建新映像，将其父项设置为源映像：

```
cephuser@adm > rbd migration prepare SRC_POOL/IMAGE TARGET_POOL/IMAGE
```

提示：仅将数据迁移到纠删码存储池

如果您只需将映像数据迁移到新 EC 存储池，而将元数据保留在原始副本存储池中，请改为运行以下命令：

```
cephuser@adm > rbd migration prepare SRC_POOL/IMAGE \
 --data-pool TARGET_POOL/IMAGE
```

3. 让客户端访问目标存储池中的映像。
4. 将数据迁移到目标存储池：

```
cephuser@adm > rbd migration execute SRC_POOL/IMAGE
```

5. 删除旧映像：

```
cephuser@adm > rbd migration commit SRC_POOL/IMAGE
```

18.7 存储池快照

存储池快照是整个 Ceph 存储池的状态快照。通过存储池快照，可以保留存储池状态的历史。创建存储池快照会占用与存储池大小成正比的存储空间。在创建存储池快照之前，始终需要检查相关存储是否有足够的磁盘空间。

18.7.1 创建存储池快照

要创建存储池快照，请运行以下命令：

```
cephuser@adm > ceph osd pool mksnap POOL-NAME SNAP-NAME
```

例如：

```
cephuser@adm > ceph osd pool mksnap pool1 snap1
created pool pool1 snap snap1
```

18.7.2 列出存储池快照

要列出存储池的现有快照，请运行以下命令：

```
cephuser@adm > rados lssnap -p POOL_NAME
```

例如：

```
cephuser@adm > rados lssnap -p pool1
1 snap1 2018.12.13 09:36:20
2 snap2 2018.12.13 09:46:03
2 snaps
```

18.7.3 删除存储池快照

要删除存储池的某个快照，请运行以下命令：

```
cephuser@adm > ceph osd pool rmsnap POOL-NAME SNAP-NAME
```

18.8 数据压缩

BlueStore（有关更多详细信息，请参见《部署指南》, 第 1 章 “SES 和 Ceph”, 第 1.4 节 “BlueStore”）提供即时数据压缩，以节省磁盘空间。压缩率取决于系统中存储的数据。请注意，压缩/解压缩需要额外的 CPU 资源。

您可以全局配置数据压缩（请参见第 18.8.3 节 “全局压缩选项”），然后覆盖各个存储池的特定压缩设置。

无论存储池是否包含数据，您都可以启用或禁用存储池数据压缩，或者随时更改压缩算法和模式。

启用存储池压缩之后，将不会向现有数据应用压缩。

禁用某个存储池的压缩之后，将会解压缩该存储池的所有数据。

18.8.1 启用压缩

要对名为 `POOL_NAME` 的存储池启用数据压缩，请运行以下命令：

```
cephuser@adm > ceph osd pool set POOL_NAME
 compression_algorithm COMPRESSION_ALGORITHM
cephuser@adm > ceph osd pool set POOL_NAME compression_mode COMPRESSION_MODE
```

提示：禁用存储池压缩

要对存储池禁用数据压缩，请使用“none”压缩算法：

```
cephuser@adm > ceph osd pool set POOL_NAME compression_algorithm none
```

18.8.2 存储池压缩选项

完整的压缩设置列表：

compression_algorithm

可用的值有 `none`、`zstd`、`snappy`。默认值为 `snappy`。

使用的压缩算法取决于特定使用情形。以下是一些相关的建议：

- 只要您没有充分的理由更改默认值 `snappy`，就请使用该值。
- `zstd` 可提供较佳的压缩率，但当压缩少量数据时，会导致 CPU 开销较高。
- 针对实际数据的样本运行这些算法的基准测试，观察集群的 CPU 和内存使用率。

compression_mode

可用的值有 `none`、`aggressive`、`passive`、`force`。默认值为 `none`。

- `none`：从不压缩
- `passive`：如果提示 `COMPRESSIBLE`，则压缩
- `aggressive`：除非提示 `INCOMPRESSIBLE`，才压缩
- `force`：始终压缩

compression_required_ratio

值：双精度型，比例 = SIZE_COMPRESSED / SIZE_ORIGINAL。默认值为 `0.875`，这表示如果压缩未将占用空间至少减少 12.5%，将不会压缩对象。

由于净增益低，存储高于此比例的对象时不会压缩。

compression_max_blob_size

值：无符号整数，大小以字节为单位。默认值：`0`

所压缩对象的最大大小。

compression_min_blob_size

值：无符号整数，大小以字节为单位。默认值：`0`

所压缩对象的最小大小。

18.8.3 全局压缩选项

可在 Ceph 配置中设置以下配置选项，并将其应用于所有 OSD 而不仅仅是单个存储池。第 18.8.2 节 “存储池压缩选项”中列出的存储池特定配置优先。

bluestore_compression_algorithm

请参见 compression_algorithm

bluestore_compression_mode

请参见 compression_mode

bluestore_compression_required_ratio

请参见 compression_required_ratio

bluestore_compression_min_blob_size

值：无符号整数，大小以字节为单位。默认值：`0`

所压缩对象的最小大小。系统默认会忽略该设置，并使用 `bluestore_compression_min_blob_size_hdd` 和 `bluestore_compression_min_blob_size_ssd` 的值。如果设置为非零值，则该设置优先。

bluestore_compression_max_blob_size

值：无符号整数，大小以字节为单位。默认值：`0`

在将对象分割为更小的块之前所压缩对象的最大大小。系统默认会忽略该设置，并使用 `bluestore_compression_max_blob_size_hdd` 和 `bluestore_compression_max_blob_size_ssd` 的值。如果设置为非零值，则该设置优先。

bluestore_compression_min_blob_size_ssd

值：无符号整数，大小以字节为单位。默认值：`8K`

压缩并存储在固态硬盘上的对象的最小大小。

bluestore_compression_max_blob_size_ssd

值：无符号整数，大小以字节为单位。默认值：`64K`

在将对象分割为更小的块之前压缩并存储在固态硬盘上的对象的最大大小。

bluestore_compression_min_blob_size_hdd

值：无符号整数，大小以字节为单位。默认值：`128K`

压缩并存储在普通硬盘上的对象的最小大小。

bluestore_compression_max_blob_size_hdd

值：无符号整数，大小以字节为单位。默认值：`512K`

在将对象分割为更小的块之前压缩并存储在硬盘上的对象的最大大小。

19 纠删码存储池

Ceph 提供了一种在存储池中正常复制数据的替代方案，称为**纠删**存储池或**纠删码**存储池。纠删码存储池不能提供**副本**存储池的所有功能（例如，它们无法存储 RBD 存储池的元数据），但其所需的原始存储空间更少。一个能够存储 1 TB 数据的默认纠删码存储池需要 1.5 TB 的原始存储空间，以应对发生单个磁盘故障的情况。从这方面而言，纠删码存储池优于副本存储池，因为后者需要 2 TB 的原始存储空间才能实现相同目的。

有关纠删码的背景信息，请参见 https://en.wikipedia.org/wiki/Erasure_code 。

有关 EC 存储池相关的存储池值列表，请参考纠删码存储池值。

19.1 纠删码存储池的先决条件

要使用纠删码，您需要：

- 在 CRUSH 索引中定义纠删码规则。
- 定义指定要使用的编码算法的纠删码配置。
- 创建使用上述规则和配置的存储池。

请记住，一旦创建好存储池且存储池中包含数据，便无法更改配置和配置中的详细信息。

确保**纠删码存储池**的 CRUSH 规则对 `step` 使用 `indep`。有关详细信息，请参见第 17.3.2 节 “firstn 和 indep”。

19.2 创建示例纠删码存储池

最简单的纠删码存储池相当于 RAID5，至少需要三个主机。以下过程介绍如何创建用于测试的存储池。

1. 命令 **`ceph osd pool create`** 用于创建类型为**纠删**的存储池。`12` 表示归置组的数量。使用默认参数时，该存储池能够处理一个 OSD 的故障。

```
cephuser@adm > ceph osd pool create ecpool 12 12 erasure
```

```
pool 'ecpool' created
```

2. 字符串 ABCDEFGHI 将写入名为 NYAN 的对象。

```
cephuser@adm > echo ABCDEFGHI | rados --pool ecpool put NYAN -
```

3. 为了进行测试，现在可以禁用 OSD，例如，断开其网络连接。

4. 要测试该存储池是否可以处理多台设备发生故障的情况，可以使用 **rados** 命令来访问文件的内容。

```
cephuser@adm > rados --pool ecpool get NYAN -
ABCDEFGHI
```

19.3 纠删码配置

调用 **ceph osd pool create** 命令来创建**纠删码存储池**时，除非指定了其他配置，否则会使用默认的配置。配置定义数据冗余。要进行这种定义，可以设置随意命名为 k 和 m 的两个参数。k 和 m 定义要将数据片段拆分成多少个块，以及要创建多少个编码块。然后，冗余块将存储在不同的 OSD 上。

纠删池配置所需的定义：

chunk

如果调用该编码函数，它会返回相同大小的块：可串联起来以重构造原始对象的数据块，以及可用于重构建丢失的块的编码块。

k

数据块的数量，即要将原始对象分割成的块数量。例如，如果 k = 2，则会将一个 10 kB 对象分割成各为 5 kB 的 k 个对象。纠删码存储池的默认 min_size 为 k + 1。不过，我们建议将 min_size 设置为 k + 2 或更大的值，以防丢失写入和数据。

m

编码块的数量，即编码函数计算的额外块的数量。如果有 2 个编码块，则表示可以移出 2 个 OSD，而不会丢失数据。

crush-failure-domain

定义要将块分布到的设备。其值需要设置为某个存储桶类型。有关所有的存储桶类型，请参见第 17.2 节 “存储桶”。如果故障域为机柜，则会将块存储在不同的机柜上，以提高机柜发生故障时的恢复能力。请记住，这需要 k+m 个机柜。

使用第 19.2 节 “创建示例纠删码存储池”中所用的默认纠删码配置时，如果单个 OSD 或主机发生故障，将不会丢失集群数据。因此，要存储 1 TB 数据，需要额外提供 0.5 TB 原始存储空间。也就是说，需要 1.5 TB 原始存储空间才能存储 1 TB 的数据（因为 k=2、m=1）。这相当于常见的 RAID 5 配置。作为对比，副本存储池需要 2 TB 原始存储空间才能存储 1 TB 数据。

可使用以下命令显示默认配置的设置：

```
cephuser@adm > ceph osd erasure-code-profile get default
directory=.libs
k=2
m=1
plugin=jerasure
crush-failure-domain=host
technique=reed_sol_van
```

选择适当的配置非常重要，因为在创建存储池后便无法修改配置。需要创建使用不同配置的新存储池，并将之前的存储池中的所有对象移到新存储池（请参见第 18.6 节 “存储池迁移”）。

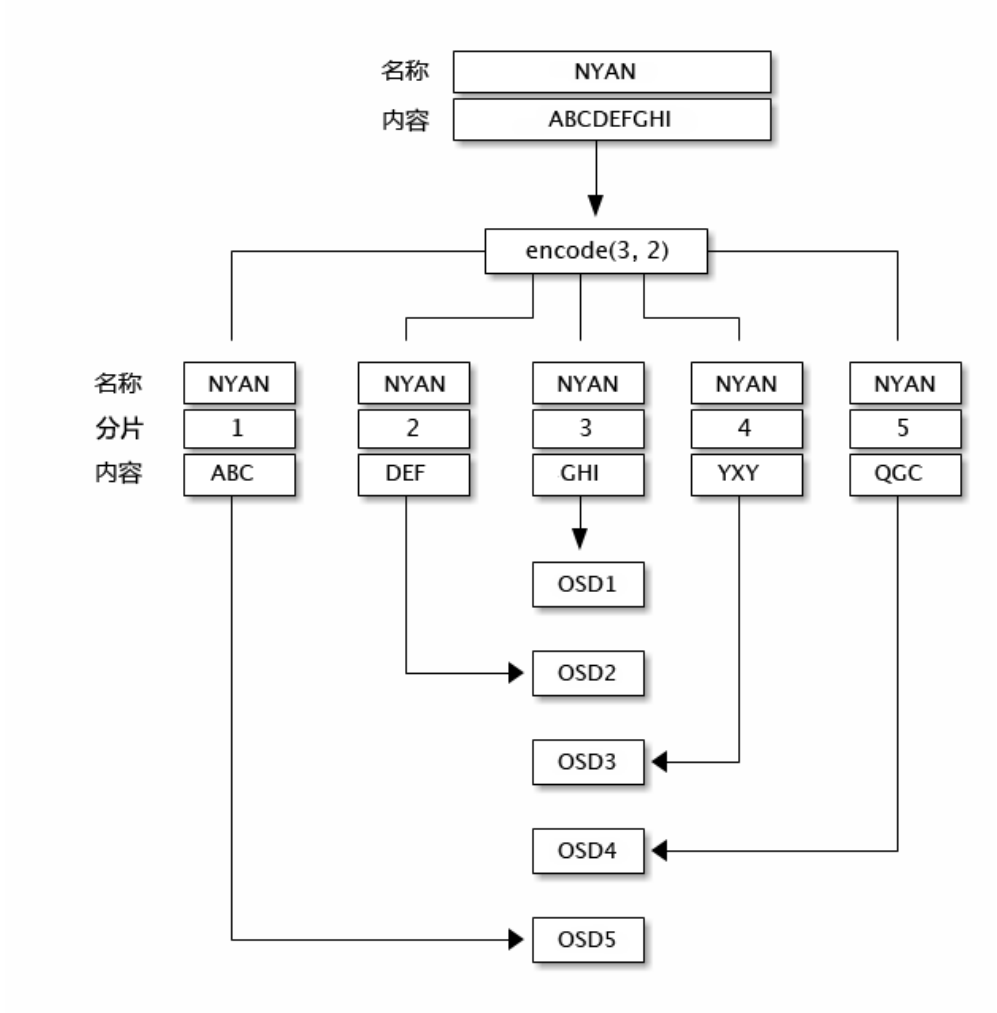
最重要的几个配置参数是 k、m 和 crush-failure-domain，因为它们定义存储开销和数据持久性。例如，如果在两个机柜发生故障并且存储开销达到 66% 的情况下，必须能够维系所需的体系结构，可定义以下配置。请注意，这仅适用于拥有“rack”类型的存储桶的 CRUSH 索引：

```
cephuser@adm > ceph osd erasure-code-profile set myprofile \
   k=3 \
   m=2 \
   crush-failure-domain=rack
```

对于此新配置，可以重复第 19.2 节 “创建示例纠删码存储池”中的示例：

```
cephuser@adm > ceph osd pool create ecpool 12 12 erasure myprofile
cephuser@adm > echo ABCDEFGHI | rados --pool ecpool put NYAN -
cephuser@adm > rados --pool ecpool get NYAN -
ABCDEFGHI
```

NYAN 对象将分割成三个 (k=3)，并将创建两个额外的块 (m=2)。m 值定义可以同时丢失多少个 OSD 而不会丢失任何数据。crush-failure-domain=rack 将创建一个 CRUSH 规则组，用于确保不会将两个块存储在同一个机柜中。



19.3.1 创建新纠删码配置

以下命令可创建新纠删码配置：

```
# ceph osd erasure-code-profile set NAME \
 directory=DIRECTORY \
 plugin=PLUGIN \
 stripe_unit=STRIPE_UNIT \
 KEY=VALUE ... \
```

```
--force
```

DIRECTORY

可选。设置从中加载纠删码插件的目录名称。默认为 `/usr/lib/ceph/erasure-code`。

PLUGIN

可选。使用纠删码插件可计算编码块和恢复缺失的块。可用的插件有“jerasure”、“isa”、“lrc”和“shes”。默认为“jerasure”。

STRIPE_UNIT

可选。数据块中每个条带的数据量。例如，如果配置拥有 2 个数据块且 stripe_unit 等于 4K，则会将范围 0-4K 的数据置于块 0 中，将 4K-8K 置于块 1 中，然后再将 8K-12K 置于块 0 中。需要有多个 4K 才能实现最佳性能。默认值取自创建存储池时的 Monitor 配置选项 `osd_pool_erasure_code_stripe_unit`。使用此配置的存储池的“stripe_width”等于数据块的数量乘以此“stripe_unit”。

KEY=VALUE

专用于选定纠删码插件的选项键/值对。

--force

可选。覆盖名称相同的现有配置，并允许设置不按 4K 对齐的 stripe_unit。

19.3.2 删除纠删码配置

以下命令可按 NAME 所标识的纠删码配置删除相应配置：

```
# ceph osd erasure-code-profile rm NAME
```

重要

如果某个存储池引用了该配置，则删除将会失败。

19.3.3 显示纠删码配置的详细信息

以下命令可按 NAME 所标识的纠删码配置显示其详细信息：

```
# ceph osd erasure-code-profile get NAME
```

19.3.4 列出纠删码配置

以下命令可列出所有纠删码配置的名称：

```
# ceph osd erasure-code-profile ls
```

19.4 标记含 RADOS 块设备的纠删码存储池

要将 EC 池标记为 RBD 池，请对其进行相应标记：

```
cephuser@adm > ceph osd pool application enable rbd ec_pool_name
```

RBD 可在 EC 池中存储映像**数据**。但是，映像报头和元数据仍需要存储在副本存储池中。为此，假设您的存储池命名为“rbd”：

```
cephuser@adm > rbd create rbd/image_name --size 1T --data-pool ec_pool_name
```

您可以像使用任何其他映像一样正常使用该映像，只不过所有数据都将存储在 `ec_pool_name` 池而非“rbd”池中。

20 RADOS 块设备

一个块就是由若干字节组成的序列，例如 4 MB 的数据块。基于块的存储接口是使用旋转媒体（例如硬盘、CD、软盘）存储数据最常见的方式。块设备接口的普及，也使得虚拟块设备成为与大量数据存储系统（例如 Ceph）进行交互的理想选择。

Ceph 块设备允许共享物理资源，并且可以调整大小。它们会在 Ceph 集群中的多个 OSD 上等量存储数据。Ceph 块设备会利用 RADOS 功能，例如创建快照、复制和一致性。Ceph 的 RADOS 块设备 (RBD) 使用内核扩展模块或 `librbd` 库与 OSD 交互。

图 20.1： RADOS 协议

Ceph 的块设备为内核扩展模块提供高性能及无限的可扩展性。它们支持虚拟化解决方案（例如 QEMU）或依赖于 `libvirt` 的基于云的计算系统（例如 OpenStack）。您可以使用同一个集群来同时操作对象网关、CephFS 和 RADOS 块设备。

20.1 块设备命令

`rbd` 命令可让您创建、列出、内省和删除块设备映像。您还可以使用它来执行其他操作，例如，克隆映像、创建快照、将映像回滚到快照或查看快照。

20.1.1 在副本存储池中创建块设备映像

将块设备添加到客户端之前，您需要在现有存储池中创建一个相关的映像（请参见第 18 章 “管理存储池”）：

```
cephuser@adm > rbd create --size MEGABYTES POOL-NAME/IMAGE-NAME
```

例如，要创建名为“myimage”的 1 GB 映像，并使其将信息存储在名为“mypool”的存储池中，请执行以下命令：

```
cephuser@adm > rbd create --size 1024 mypool/myimage
```

提示：映像大小单位

如果您省略了大小单位缩写（“G”或“T”），映像大小将以兆字节为单位。请在大小数值之后使用“G”或“T”来指定千兆字节或太字节。

20.1.2 在纠删码存储池中创建块设备映像

可以将块设备映像的数据直接存储到纠删码 (EC) 存储池中。RADOS 块设备映像由**数据**和**元数据**两部分组成。您只能将 RADOS 块设备映像的数据部分存储到 EC 存储池中。要执行此操作，存储池的 `overwrite` 标志需要设置为 **true**，并且用于存储存储池的所有 OSD 都必须使用 BlueStore。

不能将映像的元数据部分存储在 EC 存储池中。您可以使用 **`rbd create`** 命令的 `--pool=` 选项指定用于存储映像元数据的副本存储池，也可以将 `pool/` 指定为映像名称的前缀。

创建 EC 存储池：

```
cephuser@adm > ceph osd pool create EC_POOL 12 12 erasure
cephuser@adm > ceph osd pool set EC_POOL allow_ec_overwrites true
```

指定用于存储元数据的副本存储池：

```
cephuser@adm > rbd create IMAGE_NAME --size=1G --data-pool EC_POOL --pool=POOL
```

或：

```
cephuser@adm > rbd create POOL/IMAGE_NAME --size=1G --data-pool EC_POOL
```

20.1.3 列出块设备映像

要列出名为“mypool”的存储池中的块设备，请执行以下命令：

```
cephuser@adm > rbd ls mypool
```

20.1.4 检索映像信息

要从名为“mypool”的存储池内的映像“myimage”检索信息，请运行以下命令：

```
cephuser@adm > rbd info mypool/myimage
```

20.1.5 调整块设备映像的大小

RADOS 块设备映像是瘦配置 — 在您开始将数据保存到这些映像之前，它们实际上并不会使用任何物理存储。但是，这些映像具有您使用 `--size` 选项设置的最大容量。如果您要增大（或减小）映像的最大大小，请运行以下命令：

```
cephuser@adm > rbd resize --size 2048 POOL_NAME/IMAGE_NAME # to increase
cephuser@adm > rbd resize --size 2048 POOL_NAME/IMAGE_NAME --allow-shrink # to
 decrease
```

20.1.6 删除块设备映像

要删除与“mypool”存储池内的映像“myimage”对应的块设备，请运行以下命令：

```
cephuser@adm > rbd rm mypool/myimage
```

20.2 挂载和卸载

创建 RADOS 块设备之后，便可以像任何其他磁盘设备一样使用它：进行格式化、将其挂载以便能够交换文件，以及在完成时将其卸载。

`rbd` 命令默认使用 Ceph `admin` 用户帐户访问集群。此帐户拥有集群的完全管理访问权限。这会带来意外造成损害的风险，类似于以 `root` 身份登录 Linux 工作站之类。因此，最好创建具有较少特权的用户帐户，并将这些帐户用于正常的读/写 RADOS 块设备访问。

20.2.1 创建 Ceph 用户帐户

要创建拥有 Ceph Manager、Ceph Monitor 和 Ceph OSD 用户权限的新用户帐户，请将 **`ceph`** 命令与 **`auth get-or-create`** 子命令一起使用：

```
cephuser@adm > ceph auth get-or-create client.ID mon 'profile rbd' osd
 'profile profile name \
  [pool=pool-name] [, profile ...]' mgr 'profile rbd [pool=pool-name]'
```

例如，要创建一个名为 qemu 的用户，该用户拥有对存储池 vms 的读写访问权限以及对存储池 images 的只读访问权限，请执行以下命令：

```
ceph auth get-or-create client.qemu mon 'profile rbd' osd 'profile rbd pool=vms,
 profile rbd-read-only pool=images' \
  mgr 'profile rbd pool=images'
```

ceph auth get-or-create 命令的输出将是指定用户的密钥环，可将其写入 /etc/ceph/ceph.client.ID.keyring。

注意

使用 **rbd** 命令时，可以通过提供可选的 **--id** ID 参数来指定用户 ID。

有关管理 Ceph 用户帐户的更多详细信息，请参见第 30 章 “cephx 身份验证”。

20.2.2 用户身份验证

要指定用户名，请使用 --id user-name。如果您使用 cephx 身份验证，则还需要指定密钥。该机密可能来自密钥环，或某个包含机密的文件：

```
cephuser@adm > rbd device map --pool rbd myimage --id admin --keyring /path/to/
keyring
```

或

```
cephuser@adm > rbd device map --pool rbd myimage --id admin --keyfile /path/to/
file
```

20.2.3 准备 RADOS 块设备以供使用

1. 确保您的 Ceph 集群的存储池中包含要映射的磁盘映像。假设存储池名为 mypool，映像名为 myimage。

```
cephuser@adm > rbd list mypool
```

2. 将映像映射到新的块设备：

```
cephuser@adm > rbd device map --pool mypool myimage
```

3. 列出所有映射的设备：

```
cephuser@adm > rbd device list
id pool   image   snap device
0  mypool myimage -    /dev/rbd0
```

我们要使用的设备是 `/dev/rbd0`。

提示：RBD 设备路径

您可以使用 `/dev/rbd/POOL_NAME/IMAGE_NAME`作为永久设备路径来代替 `/dev/rbdDEVICE_NUMBER`。例如：

```
       /dev/rbd/mypool/myimage
```

4. 在 `/dev/rbd0` 设备上创建 XFS 文件系统：

```
# mkfs.xfs /dev/rbd0
      log stripe unit (4194304 bytes) is too large (maximum is 256KiB)
      log stripe unit adjusted to 32KiB
      meta-data=/dev/rbd0              isize=256    agcount=9,
 agsize=261120 blks
      =                       sectsz=512   attr=2, projid32bit=1
      =                       crc=0        finobt=0
      data     =                       bsize=4096   blocks=2097152,
 imaxpct=25
      =                       sunit=1024   swidth=1024 blks
      naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
      log      =internal log           bsize=4096   blocks=2560, version=2
      =                       sectsz=512   sunit=8 blks, lazy-count=1
      realtime =none                   extsz=4096   blocks=0, rtextents=0
```

5. 使用您的挂载点替换 `/mnt`，挂载设备并检查其是否正确挂载：

```
# mount /dev/rbd0 /mnt
      # mount | grep rbd0
      /dev/rbd0 on /mnt type xfs (rw,relatime,attr2,inode64,sunit=8192,...
```

现在，您便可以将数据移入/移出设备，就如同它是本地目录一样。

提示：增大 RBD 设备的大小

如果您发现 RBD 设备的大小不再够用，可以轻松增大大小。

1. 增大 RBD 映像的大小，例如增大到 10GB。

```
cephuser@adm > rbd resize --size 10000 mypool/myimage
         Resizing image: 100% complete...done.
```

2. 扩大文件系统以填充设备的新大小：

```
# xfs_growfs /mnt
[...]
data blocks changed from 2097152 to 2560000
```

6. 当您访问完设备后，可以将其取消映射并卸载。

```
cephuser@adm > rbd device unmap /dev/rbd0
# unmount /mnt
```

提示：手动挂载和卸载

提供了 **`rbdmap`** 脚本和 `systemd` 单元，可更加顺畅地在引导后映射和挂载 RBD，并在关机前将其卸载。有关详细信息，请参见第 20.2.4 节 “**`rbdmap`**：在引导时映射 RBD 设备”。

20.2.4 rbdmap：在引导时映射 RBD 设备

rbdmap 是一个外壳脚本，可对一个或多个 RBD 映像自动执行 **rbd map** 和 **rbd device unmap** 操作。虽然您随时都可以手动运行该脚本，但其主要优势是在引导时自动映射和挂载 RBD 映像（以及在关机时卸载和取消映射），此操作由 Init 系统触发。systemd 软件包中随附了一个 单元文件 rbdmap.serviceceph-common 用于执行此操作。

该脚本使用单个参数，可以是 map 或 unmap。使用任一参数时，该脚本都会分析配置文件。它默认为 /etc/ceph/rbdmap，但可通过环境变量 rbdmapFILE 覆盖。该配置文件的每一行相当于一个要映射或取消映射的 RBD 映像。

配置文件采用以下格式：

```
image_specification rbd_options
```

image_specification

存储池中映像的路径。以 pool_name/image_name 格式指定。

rbd_options

要传递给底层 **rbd device map** 命令的参数的可选列表。这些参数及其值应该以逗号分隔的字符串形式指定，例如：

```
PARAM1=VAL1,PARAM2=VAL2,...
```

该示例让 **rbdmap** 脚本运行以下命令：

```
cephuser@adm > rbd device map POOL_NAME/IMAGE_NAME --PARAM1 VAL1 --PARAM2
 VAL2
```

下面的示例中介绍了如何使用相应的密钥指定用户名和密钥环：

```
cephuser@adm > rbdmap device map mypool/myimage id=rbd_user,keyring=/etc/
ceph/ceph.client.rbd.keyring
```

以 **rbdmap map** 形式运行时，该脚本会分析配置文件，并且对于每个指定的 RBD 映像，它会尝试先映射映像（使用 **rbd device map** 命令），再挂载映像。

以 **rbdmap unmap** 形式运行时，配置文件中列出的映像将卸载并取消映射。

rbdmap unmap-all 会尝试卸载然后取消映射所有当前已映射的 RBD 映像，而不论它们是否列在配置文件中。

如果成功，**rbd device map** 操作会将映像映射到 /dev/rbdX 设备，此时会触发 udev 规则，以创建易记设备名称符号链接 /dev/rbd/pool_name/image_name，该链接指向实际映射的设备。

为了挂载和卸载成功，易记设备名称在 /etc/fstab 中需有对应项。写入 RBD 映像的 /etc/fstab 项时，指定“noauto”（或“nofail”）挂载选项。这可防止 Init 系统过早（尚未出现有问题的设备时）尝试挂载设备，因为 rbdmap.service 通常是在引导序列中相当靠后的时间触发。

有关 **rbd** 选项的完整列表，请参见 **rbd** 手册页 (**man 8 rbd**)。

有关 **rbdmap** 用法的示例，请参见 **rbdmap** 手册页 (**man 8 rbdmap**)。

20.2.5 增大 RBD 设备的大小

如果您发现 RBD 设备的大小不再够用，可以轻松增大大小。

1. 增大 RBD 映像的大小，例如增大到 10GB。

```
cephuser@adm > rbd resize --size 10000 mypool/myimage
 Resizing image: 100% complete...done.
```

2. 扩大文件系统以填入设备的新大小。

```
# xfs_growfs /mnt
 [...]
 data blocks changed from 2097152 to 2560000
```

20.3 快照

RBD 快照是 RADOS 块设备映像的快照。通过快照，您可以保留映像状态的历史。Ceph 还支持快照分层，这可让您轻松快速地克隆 VM 映像。Ceph 使用 **rbd** 命令和许多高级接口（包括 QEMU、libvirt、OpenStack 和 CloudStack）支持块设备快照。

注意

在创建映像的快照之前，请停止输入和输出操作，并刷新所有待处理写操作。如果映像包含文件系统，则在创建快照时，文件系统必须处于一致状态。

20.3.1 启用和配置 cephx

如果启用了 cephx，则必须指定用户名或 ID，以及包含用户相应密钥的密钥环的路径。有关详细信息，请参见第 30 章 “cephx 身份验证”。您还可以添加 CEPH_ARGS 环境变量，以免重新输入以下参数。

```
cephuser@adm > rbd --id user-ID --keyring=/path/to/secret commands
cephuser@adm > rbd --name username --keyring=/path/to/secret commands
```

例如：

```
cephuser@adm > rbd --id admin --keyring=/etc/ceph/ceph.keyring commands
cephuser@adm > rbd --name client.admin --keyring=/etc/ceph/ceph.keyring commands
```

提示

将用户和机密添加到 CEPH_ARGS 环境变量，如此您便无需每次都输入它们。

20.3.2 快照基础知识

下面的过程说明如何在命令行上使用 **rbd** 命令创建、列出和删除快照。

20.3.2.1 创建快照

要使用 **rbd** 创建快照，请指定 snap create 选项、存储池名称和映像名称。

```
cephuser@adm > rbd --pool pool-name snap create --snap snap-name image-name
cephuser@adm > rbd snap create pool-name/image-name@snap-name
```

例如：

```
cephuser@adm > rbd --pool rbd snap create --snap snapshot1 image1
cephuser@adm > rbd snap create rbd/image1@snapshot1
```

20.3.2.2 列出快照

要列出映像的快照，请指定存储池名称和映像名称。

```
cephuser@adm > rbd --pool pool-name snap ls image-name
cephuser@adm > rbd snap ls pool-name/image-name
```

例如：

```
cephuser@adm > rbd --pool rbd snap ls image1
cephuser@adm > rbd snap ls rbd/image1
```

20.3.2.3 回滚快照

要使用 **rbd** 回滚快照，请指定 `snap rollback` 选项、存储池名称、映像名称和快照名称。

```
cephuser@adm > rbd --pool pool-name snap rollback --snap snap-name image-name
cephuser@adm > rbd snap rollback pool-name/image-name@snap-name
```

例如：

```
cephuser@adm > rbd --pool pool1 snap rollback --snap snapshot1 image1
cephuser@adm > rbd snap rollback pool1/image1@snapshot1
```

注意

将映像回滚到快照意味着会使用快照中的数据重写当前版本的映像。执行回滚所需的时间将随映像大小的增加而延长。从快照**克隆较快**，而从映像到快照的**回滚较慢**，因此克隆是返回先前存在状态的首选方法。

20.3.2.4 删除快照

要使用 **rbd** 删除快照，请指定 `snap rm` 选项、存储池名称、映像名称和用户名。

```
cephuser@adm > rbd --pool pool-name snap rm --snap snap-name image-name
cephuser@adm > rbd snap rm pool-name/image-name@snap-name
```

例如：

```
cephuser@adm > rbd --pool pool1 snap rm --snap snapshot1 image1
cephuser@adm > rbd snap rm pool1/image1@snapshot1
```

注意

Ceph OSD 会以异步方式删除数据，因此删除快照不能立即释放磁盘空间。

20.3.2.5 清除快照

要使用 **rbd** 删除映像的所有快照，请指定 `snap purge` 选项和映像名称。

```
cephuser@adm > rbd --pool pool-name snap purge image-name
cephuser@adm > rbd snap purge pool-name/image-name
```

例如：

```
cephuser@adm > rbd --pool pool1 snap purge image1
cephuser@adm > rbd snap purge pool1/image1
```

20.3.3 快照分层

Ceph 支持为一个块设备快照创建多个写入时复制 (COW) 克隆的功能。快照分层可让 Ceph 块设备客户端能够极快地创建映像。例如，您可以创建块设备映像并将 Linux VM 写入其中，然后创建映像的快照、保护快照，并创建您所需数量的“写入时复制”克隆。快照是只读的，因此克隆快照简化了语义，如此可快速创建克隆。

注意

下面的命令行示例中提到的“父”和“子”这两个术语是指 Ceph 块设备快照（父）和从快照克隆的相应映像（子）。

每个克隆的映像（子）都存储了对其父映像的引用，这可让克隆的映像打开父快照并读取其内容。

快照的 COW 克隆的行为方式与任何其他 Ceph 块设备映像完全相同。可针对克隆的映像执行读取、写入、克隆和调整大小操作。系统对克隆的映像没有特殊限制。但是，快照的写入时复制克隆会引用快照，因此您**必须**在克隆快照之前保护快照。

注意：不支持 `--image-format 1`

您无法为通过弃用的 **`rbd create --image-format 1`** 选项创建的映像创建快照。Ceph 仅支持克隆默认的 **format 2** 映像。

20.3.3.1 分层入门

Ceph 块设备分层是一个简单的过程。您必须有一个映像。您必须创建映像的快照。您必须保护快照。在您执行这些步骤之后，就可以开始克隆快照了。

克隆的映像具有对父快照的引用，并且包含存储池 ID、映像 ID 和快照 ID。包含存储池 ID 意味着您可以将快照从一个存储池克隆到另一个存储池中的映像。

- **映像模板**：一种常见的块设备分层用例是创建主映像和用作克隆模板的快照。例如，用户可为 Linux 发行套件（如 SUSE Linux Enterprise Server）创建映像并为它创建快照。用户可以定期更新映像和创建新的快照（例如，先执行 **`zypper ref && zypper patch`**，接着执行 **`rbd snap create`**）。随着映像日趋成熟，用户可以克隆任何一个快照。
- **扩展模板**：更高级的用例包括扩展比基本映像提供的信息更多的模板映像。例如，用户可以克隆映像（VM 模板）并安装其他软件（例如，数据库、内容管理系统或分析系统），然后创建扩展映像的快照，这个扩展映像可以如基本映像一样更新。
- **模板池**：使用块设备分层的一种方法是创建包含主映像（用作模板）的池，然后创建这些模板的快照。之后，您便可以扩大用户的只读特权，使他们可以克隆快照，却不能写入存储池或在存储池中执行。
- **映像迁移/恢复**：使用块设备分层的一种方法是将数据从一个存储池迁移或恢复到另一个存储池。

20.3.3.2 保护快照

克隆会访问父快照。如果用户意外删除了父快照，则所有克隆都会损坏。为了防止数据丢失，您需要先保护快照，然后才能克隆它。

```
cephuser@adm > rbd --pool pool-name snap protect \
 --image image-name --snap snapshot-name
cephuser@adm > rbd snap protect pool-name/image-name@snapshot-name
```

例如：

```
cephuser@adm > rbd --pool pool1 snap protect --image image1 --snap snapshot1
cephuser@adm > rbd snap protect pool1/image1@snapshot1
```

注意

您无法删除受保护的快照。

20.3.3.3 克隆快照

要克隆快照，您需要指定父存储池、映像、快照、子存储池和映像名称。您需要先保护快照，然后才能克隆它。

```
cephuser@adm > rbd clone --pool pool-name --image parent-image \
 --snap snap-name --dest-pool pool-name \
 --dest child-image
cephuser@adm > rbd clone pool-name/parent-image@snap-name \
pool-name/child-image-name
```

例如：

```
cephuser@adm > rbd clone pool1/image1@snapshot1 pool1/image2
```

注意

您可以将快照从一个存储池克隆到另一个存储池中的映像。例如，可以在一个存储池中将只读映像和快照作为模板维护，而在另一个存储池中维护可写入克隆。

20.3.3.4 取消保护快照

必须先取消保护快照，然后才能删除它。另外，您**无法**删除克隆所引用的快照。您需要先平展快照的每个克隆，然后才能删除快照。

```
cephuser@adm > rbd --pool pool-name snap unprotect --image image-name \
 --snap snapshot-name
cephuser@adm > rbd snap unprotect pool-name/image-name@snapshot-name
```

例如：

```
cephuser@adm > rbd --pool pool1 snap unprotect --image image1 --snap snapshot1
cephuser@adm > rbd snap unprotect pool1/image1@snapshot1
```

20.3.3.5 列出快照的子项

要列出快照的子项，请执行以下命令：

```
cephuser@adm > rbd --pool pool-name children --image image-name --snap snap-name
cephuser@adm > rbd children pool-name/image-name@snapshot-name
```

例如：

```
cephuser@adm > rbd --pool pool1 children --image image1 --snap snapshot1
cephuser@adm > rbd children pool1/image1@snapshot1
```

20.3.3.6 平展克隆的映像

克隆的映像会保留对父快照的引用。删除子克隆对父快照的引用时，可通过将信息从快照复制到克隆，高效“平展”映像。平展克隆所需的时间随着映像大小的增加而延长。要删除快照，必须先平展子映像。

```
cephuser@adm > rbd --pool pool-name flatten --image image-name
cephuser@adm > rbd flatten pool-name/image-name
```

例如：

```
cephuser@adm > rbd --pool pool1 flatten --image image1
cephuser@adm > rbd flatten pool1/image1
```

注意

由于平展的映像包含快照中的所有信息，平展的映像占用的存储空间将比分层克隆多。

20.4 RBD 映像镜像

RBD 映像可以在两个 Ceph 集群之间异步镜像。此功能有两种模式：

基于日志

此模式使用 RBD 日志映像功能来确保集群之间的复制在时间点和崩溃时保持一致。在修改实际映像之前，向 RBD 映像的每次写入都会先记录到关联日志中。`remote` 集群将从日志中读取并向其本地映像副本重放更新。由于向 RBD 映像的每次写入都将导致向 Ceph 集群的两次写入，因此在使用 RBD 日志映像功能时，预计写入延迟将增加近一倍。

基于快照

此模式使用定期安排或手动创建的 RBD 映像镜像快照，以在集群之间复制崩溃时保持一致的 RBD 映像。`remote` 集群将决定两个镜像快照之间的任何数据或元数据更新，并将增量复制到映像的本地副本。借助 RBD fast-diff 映像功能，可以快速计算更新的数据块，而无需扫描完整的 RBD 映像。由于此模式不确保在时间点保持一致，因此在故障转移期间使用该模式之前，需要同步完整的快照增量。任何部分应用的快照增量都将回滚到使用该模式前最后一个完全同步的快照。

基于对等集群中的每个存储池配置镜像。可以对存储池中的特定映像子集进行配置，也可以配置为在仅使用基于日志的镜像时自动镜像存储池中的所有映像。镜像使用 **rbd** 命令进行配置。`rbd-mirror` 守护进程负责从 `remote` 对等集群提取映像更新，并将它们应用于 `local` 集群中的映像。

根据所需的复制需求，RBD 镜像可以配置为单向或双向复制：

单向复制

当数据仅会从主集群镜像到次集群时，`rbd-mirror` 守护进程仅在次集群上运行。

双向复制

当数据从一个集群上的主映像镜像到另一个集群上的非主映像（反之亦然）时，`rbd-mirror` 守护进程将在两个集群上运行。

重要

`rbd-mirror` 守护进程的每个实例需要能够同时连接到 `local` 和 `remote` Ceph 集群。例如，所有 Monitor 和 OSD 主机。此外，网络需要两个数据中心之间有足够的带宽来处理镜像工作负载。

20.4.1 存储池配置

以下过程说明如何使用 **`rbd`** 命令来执行配置镜像的基本管理任务。镜像在 Ceph 集群中逐池进行配置。

您需要在两个同伴集群上执行存储池配置步骤。为清楚起见，这些过程假设可从单台主机访问名为 `local` 和 `remote` 的两个集群。

有关如何连接到不同的 Ceph 集群的更多详细信息，请参见 **`rbd`** 手册页 (**`man 8 rbd`**)。

提示：多个集群

以下示例中的集群名称对应于同名 `/etc/ceph/remote.conf` 的 Ceph 配置文件以及同名 `/etc/ceph/remote.client.admin.keyring` 的 Ceph 密钥环文件。

20.4.1.1 在存储池上启用镜像

要针对存储池启用镜像，请指定 **`mirror pool enable`** 子命令、存储池名称和镜像模式。镜像模式可以是存储池或映像：

pool

将会镜像启用了日志特性的存储池中的所有映像。

image

需要针对每个映像明确启用镜像。有关更多信息，请参见第 20.4.2.1 节 “启用映像镜像”。

例如：

```
cephuser@adm > rbd --cluster local mirror pool enable POOL_NAME pool
```

```
cephuser@adm > rbd --cluster remote mirror pool enable POOL_NAME pool
```

20.4.1.2 禁用镜像

要对存储池禁用镜像，请指定 **`mirror pool disable`** 子命令和存储池名称。使用这种方法对存储池禁用镜像时，还会对已为其明确启用镜像的所有映像（该存储池中）禁用镜像。

```
cephuser@adm > rbd --cluster local mirror pool disable POOL_NAME
cephuser@adm > rbd --cluster remote mirror pool disable POOL_NAME
```

20.4.1.3 引导对等

为了使 `rbd-mirror` 守护进程发现其对等集群，需要将对等注册到存储池，并需要创建用户帐户。此过程可以通过 **`rbd`**、**`mirror pool peer bootstrap create`** 和 **`mirror pool peer bootstrap import`** 命令自动完成。

要使用 **`rbd`** 手动创建新的引导令牌，请指定 **`mirror pool peer bootstrap create`** 命令、存储池名称以及描述 `local` 集群的可选易记站点名称：

```
cephuser@local > rbd mirror pool peer bootstrap create \
 [--site-name LOCAL_SITE_NAME] POOL_NAME
```

`mirror pool peer bootstrap create` 的输出将是应提供给 **`mirror pool peer bootstrap import`** 命令的令牌。例如，在 `local` 集群上：

```
cephuser@local > rbd --cluster local mirror pool peer bootstrap create --site-
name local image-pool
eyJmc2lkIjoiOWY1MjgyZGItYjg5OS00NTk2LTgwOTgtMzIwYzFmYzM5NmYzIiwiY2xpZW50X2lkIjoicmJkLW1pcnJv
 \
joiQVFBUnczOWQwdkhvQmhBQVlMM1I4RmR5dHNJQU50bkFTZ0lOTVE9PSIsIm1vbl9ob3N0IjoiW3YyOjE5Mi4xNjguM
```

要使用 **`rbd`** 命令手动导入另一个集群创建的引导令牌，请使用以下语法：

```
rbd mirror pool peer bootstrap import \
 [--site-name LOCAL_SITE_NAME] \
 [--direction DIRECTION \
 POOL_NAME TOKEN_PATH
```

位置：

`LOCAL_SITE_NAME`

用于描述 `local` 集群的可选易记站点名称。

`DIRECTION`

镜像方向。默认设为 `rx-tx` 进行双向镜像，但也可设为 `rx-only` 进行单向镜像。

`POOL_NAME`

存储池的名称。

`TOKEN_PATH`

指向所创建令牌的文件路径（或设为 `-` 以通过标准输入来读取）。

例如，在 `remote` 集群上：

```
cephuser@remote > cat <<EOF > token
eyJmc2lkIjoiOWY1MjgyZGItYjg5OS00NTk2LTgwOTgtMzIwYzFmYzM5NmYzIiwiY2xpZW50X2lkIjoicmJkLW1pcnJv
EOF
```

```
cephuser@adm > rbd --cluster remote mirror pool peer bootstrap import \
 --site-name remote image-pool token
```

20.4.1.4 手动添加集群对等

除了按第 20.4.1.3 节 “引导对等”中所述引导对等之外，您还可以手动指定对等。远程 `rbd-mirror` 守护进程需要访问本地集群才能执行镜像。创建远程 `rbd-mirror` 守护进程将使用的新的本地 Ceph 用户，例如 `rbd-mirror-peer`：

```
cephuser@adm > ceph auth get-or-create client.rbd-mirror-peer \
 mon 'profile rbd' osd 'profile rbd'
```

使用以下语法通过 **rbd** 命令添加镜像对等 Ceph 集群：

```
rbd mirror pool peer add POOL_NAME CLIENT_NAME@CLUSTER_NAME
```

例如：

```
cephuser@adm > rbd --cluster site-a mirror pool peer add image-pool client.rbd-
mirror-peer@site-b
cephuser@adm > rbd --cluster site-b mirror pool peer add image-pool client.rbd-
mirror-peer@site-a
```

默认情况下，`rbd-mirror` 守护进程需要有权访问位于 `/etc/ceph/.CLUSTER_NAME.conf` 的 Ceph 配置文件。它提供对等集群的 MON 的 IP 地址和名为 `CLIENT_NAME` 的客户端的密钥环（位于默认或自定义密钥环搜索路径中，例如 `/etc/ceph/CLUSTER_NAME.CLIENT_NAME.keyring`）。

或者，对等集群的 MON 和/或客户端密钥可以安全地存储在本地 Ceph config-key 存储区中。要在添加镜像对等时指定对等集群连接属性，请使用 `--remote-mon-host` 和 `--remote-key-file` 选项。例如：

```
cephuser@adm > rbd --cluster site-a mirror pool peer add image-pool \
 client.rbd-mirror-peer@site-b --remote-mon-host 192.168.1.1,192.168.1.2 \
 --remote-key-file /PATH/TO/KEY_FILE
cephuser@adm > rbd --cluster site-a mirror pool info image-pool --all
Mode: pool
Peers:
  UUID        NAME   CLIENT                MON_HOST                KEY
  587b08db... site-b client.rbd-mirror-peer 192.168.1.1,192.168.1.2 AQAeuZdb...
```

20.4.1.5 删除集群对等

要删除镜像同伴集群，请指定 **`mirror pool peer remove`** 子命令、存储池名称和对等 UUID（可通过 **`rbd mirror pool info`** 命令获得）：

```
cephuser@adm > rbd --cluster local mirror pool peer remove POOL_NAME \
 55672766-c02b-4729-8567-f13a66893445
cephuser@adm > rbd --cluster remote mirror pool peer remove POOL_NAME \
 60c0e299-b38f-4234-91f6-eed0a367be08
```

20.4.1.6 数据存储池

在目标集群中创建映像时，`rbd-mirror` 会按如下所述选择数据存储池：

- 如果目标集群配置了默认数据存储池（使用 `rbd_default_data_pool` 配置选项），则会使用该默认数据存储池。
- 否则，如果源映像使用单独的数据存储池，并且目标集群上存在同名的存储池，则将使用该存储池。
- 如果以上两种情况都不成立，将不会设置数据存储池。

20.4.2 RBD 映像配置

与存储池配置不同，映像配置只需要针对单个镜像同伴 Ceph 集群执行。

系统会将镜像的 RBD 映像指定为**主要**或**非主要**。这是映像的属性，而不是存储池的属性。不能修改指定为非主要的映像。

当首次对某个映像启用镜像时（如果存储池镜像模式是“存储池”并且映像已启用日志映像功能，则为隐式启用，或可通过 **rbd** 命令显式启用（请参见第 20.4.2.1 节 “启用映像镜像”），映像会自动升级为主要映像。

20.4.2.1 启用映像镜像

如果镜像配置为使用 `image` 模式，则需要为存储池中的每个映像明确启用镜像。要使用 **rbd** 为特定映像启用镜像，请指定 **`mirror image enable`** 子命令以及存储池和映像名称：

```
cephuser@adm > rbd --cluster local mirror image enable \
 POOL_NAME/IMAGE_NAME
```

镜像映像模式可以是 `journal`，也可以是 `snapshot`：

journal（默认模式）

如果配置为使用 `journal` 模式，镜像将使用 RBD 日志映像功能来复制映像内容。如果尚未在映像上启用 RBD 日志映像功能，该功能将自动启用。

snapshot

如果配置为使用 `snapshot` 模式，镜像将使用 RBD 映像镜像快照来复制映像内容。如果启用该模式，将自动创建初始镜像快照。可通过 **rbd** 命令创建其他 RBD 映像镜像快照。

例如：

```
cephuser@adm > rbd --cluster local mirror image enable image-pool/image-1
 snapshot
cephuser@adm > rbd --cluster local mirror image enable image-pool/image-2
 journal
```

20.4.2.2 启用映像日志功能

RBD 镜像使用 RBD 日志功能来确保复制的映像始终在崩溃时保持一致状态。使用 `image` 镜像模式时，如果在映像上启用了镜像，则将自动启用日志功能。使用 `pool` 镜像模式时，必须先启用 RBD 映像日志功能，然后才能将映像镜像到对等集群。在创建映像时，可以通过将 `--image-feature exclusive-lock,journaling` 选项提供给 **`rbd`** 命令来启用该功能。

或者，日志功能可以针对预先存在的 RBD 映像动态启用。要启用日志，请指定 **`feature enable`** 子命令、存储池和映像名称以及功能名称：

```
cephuser@adm > rbd --cluster local feature enable POOL_NAME/IMAGE_NAME
 exclusive-lock
cephuser@adm > rbd --cluster local feature enable POOL_NAME/IMAGE_NAME
 journaling
```

注意：选项依赖性

`journaling` 功能依赖于 `exclusive-lock` 功能。如果 `exclusive-lock` 功能尚未启用，则您需要先启用它，再启用 `journaling` 功能。

提示

默认情况下，您可以通过将 `rbd default features = layering,exclusive-lock,object-map,deep-flatten,journaling` 添加到 Ceph 配置文件，在所有新映像上启用日志功能。

20.4.2.3 创建映像镜像快照

使用基于快照的镜像时，每当要镜像 RBD 映像的已更改内容，都需要创建镜像快照。要使用 **rbd** 手动创建镜像快照，请指定 **mirror image snapshot** 命令以及存储池和映像名称：

```
cephuser@adm > rbd mirror image snapshot POOL_NAME/IMAGE_NAME
```

例如：

```
cephuser@adm > rbd --cluster local mirror image snapshot image-pool/image-1
```

默认情况下，每个映像只能创建三个镜像快照。如果达到此限制，将自动修剪最近的镜像快照。可以视需要通过 rbd_mirroring_max_mirroring_snapshots 配置选项覆盖该限制。此外，删除映像或禁用镜像时，会自动删除镜像快照。

如果定义了镜像快照日程安排，也可以定期自动创建镜像快照。可以在全局、存储池或映像级别安排镜像快照。可以在任何级别定义多个镜像快照日程安排，但只有与单个镜像映像相匹配的最具针对性的快照日程安排才会运行。

要使用 **rbd** 创建镜像快照日程安排，请指定 **mirror snapshot schedule add** 命令以及可选的存储池或映像名称、间隔和可选的开始时间。

可以分别使用后缀 d、h 或 m 指定以天、小时或分钟为单位的间隔。可使用 ISO 8601 时间格式指定可选的开始时间。例如：

```
cephuser@adm > rbd --cluster local mirror snapshot schedule add --pool image-
pool 24h 14:00:00-05:00
cephuser@adm > rbd --cluster local mirror snapshot schedule add --pool image-
pool --image image1 6h
```

要使用 **rbd** 删除镜像快照日程安排，请指定 **mirror snapshot schedule remove** 命令以及与相应的添加日程安排命令相匹配的选项。

要使用 **rbd** 列出特定级别（全局、存储池或映像）的所有快照日程安排，请指定 **mirror snapshot schedule ls** 命令以及可选的存储池或映像名称。此外，还可以指定 --recursive 选项，以列出指定及以下级别的所有日程安排。例如：

```
cephuser@adm > rbd --cluster local mirror schedule ls --pool image-pool --
recursive
POOL        NAMESPACE IMAGE  SCHEDULE
image-pool  -         -      every 1d starting at 14:00:00-05:00
```

```
image-pool            image1 every 6h
```

要获知使用 **rbd** 为基于快照的镜像 RBD 映像创建下一个快照的时间，请指定 **mirror snapshot schedule status** 命令以及可选的存储池或映像名称。例如：

```
cephuser@adm > rbd --cluster local mirror schedule status
SCHEDULE TIME       IMAGE
2020-02-26 18:00:00 image-pool/image1
```

20.4.2.4 禁用映像镜像

要为特定映像禁用镜像，请指定 **mirror image disable** 子命令以及存储池和映像名称：

```
cephuser@adm > rbd --cluster local mirror image disable POOL_NAME/IMAGE_NAME
```

20.4.2.5 升级和降级映像

在需要将主要指定移动到同伴集群中映像的故障转移情况下，您需要停止访问主要映像、降级当前主要映像、升级新的主要映像，然后继续访问替代集群上的映像。

注意：强制升级

可以使用 `--force` 选项强制升级。降级不能传播到同伴集群时（例如，当集群发生故障或通讯中断时），就需要强制升级。这将导致两个同伴集群之间出现节点分裂情况，并且映像不再同步，直到发出了 **resync** 子命令。

要将特定映像降级为非主要映像，请指定 **mirror image demote** 子命令以及存储池和映像名称：

```
cephuser@adm > rbd --cluster local mirror image demote POOL_NAME/IMAGE_NAME
```

要将存储池中的所有主要映像都降级为非主要映像，请指定 **mirror pool demote** 子命令以及存储池名称：

```
cephuser@adm > rbd --cluster local mirror pool demote POOL_NAME
```

要将特定映像升级为主要映像，请指定 **`mirror image promote`** 子命令以及存储池和映像名称：

```
cephuser@adm > rbd --cluster remote mirror image promote POOL_NAME/IMAGE_NAME
```

要将存储池中的所有非主要映像都升级为主要映像，请指定 **`mirror pool promote`** 子命令以及存储池名称：

```
cephuser@adm > rbd --cluster local mirror pool promote POOL_NAME
```

提示：拆分 I/O 负载

因为主要或非主要状态是针对映像的，所以可以使用两个集群来分割 I/O 负载并进行故障转移或故障回复。

20.4.2.6 强制映像重新同步

如果 `rbd-mirror` 守护进程检测到分区事件，则在该情况解决之前，它不会尝试镜像受影响的映像。要继续镜像映像，请先降级确定过期的映像，然后请求与主要映像重新同步。要请求映像重新同步，请指定 **`mirror image resync`** 子命令以及存储池和映像名称：

```
cephuser@adm > rbd mirror image resync POOL_NAME/IMAGE_NAME
```

20.4.3 检查镜像状态

系统会存储每个主要镜像映像的同伴集群复制状态。此状态可使用 **`mirror image status`** 和 **`mirror pool status`** 子命令检索：

要请求镜像映像状态，请指定 **`mirror image status`** 子命令以及存储池和映像名称：

```
cephuser@adm > rbd mirror image status POOL_NAME/IMAGE_NAME
```

要请求镜像存储池摘要状态，请指定 **`mirror pool status`** 子命令以及存储池名称：

```
cephuser@adm > rbd mirror pool status POOL_NAME
```

提示：

将 `--verbose` 选项添加到 **`mirror pool status`** 子命令会额外地输出存储池中每个镜像映像的状态详细信息。

20.5 缓存设置

Ceph 块设备的用户空间实现 (`librbd`) 无法利用 Linux 页面缓存。因此，它具有自己的内存中缓存。RBD 缓存的行为与硬盘缓存类似。当 OS 发送屏障或刷新请求时，所有“脏”数据都会写入 OSD。这意味着只要虚拟机可以正确发送刷新请求，使用写回缓存与使用运行良好的物理硬盘一样安全。该缓存运用**最久未使用** (LRU) 算法，并且在写回模式下可以合并相邻请求以提高吞吐量。

Ceph 支持为 RBD 提供写回缓存。要启用该功能，请运行

```
cephuser@adm > ceph config set client rbd_cache true
```

`librbd` 默认不会执行任何缓存。写和读都直接到达存储集群，并且只有当数据的所有副本都写入磁盘后写操作才会返回。如果启用了缓存，写操作会立即返回，除非未刷新的字节数大于 `rbd cache max dirty` 选项中设置的数值。在此情况下，写操作会触发写回机制并一直阻塞，直至有足够多的字节得到刷新。

Ceph 支持为 RBD 提供直写缓存。您可以设置缓存的大小，以及从写回缓存切换到直写缓存的目标和限值。要启用直写模式，请运行

```
cephuser@adm > ceph config set client rbd_cache_max_dirty 0
```

这意味着只有当数据的所有副本都写入磁盘后写操作才会返回，但可能会从缓存中读取数据。缓存信息保存在客户端的内存中，并且每个 RBD 都有自己的缓存。由于对客户端而言缓存位于本地，因此如果有其他客户端访问映像，不会存在缓存一致性的问题。如果启用了缓存，在 RBD 上将不能运行 GFS 或 OCFS。

以下参数会影响 RADOS 块设备的行为。要设置这些参数，请使用 `client` 类别：

```
cephuser@adm > ceph config set client PARAMETER VALUE
```

`rbd cache`

对 RADOS 块设备 (RBD) 启用缓存。默认值为“true”。

`rbd cache size`

RBD 缓存大小（以字节为单位）。默认值为 32 MB。

`rbd cache max dirty`

使缓存触发写回机制的“脏”数据上限（以字节为单位）。`rbd cache max dirty` 的值必须小于 `rbd cache size` 的值。如果设置为 0，将使用直写缓存。默认值为 24 MB。

`rbd cache target dirty`

达到该“脏数据目标”后，缓存即会开始向数据存储空间写入数据。该设置不会使写入缓存的操作阻塞。默认值为 16 MB。

`rbd cache max dirty age`

写回开始前，脏数据在缓存中暂存的时间（以秒为单位）。默认值为 1。

`rbd cache writethrough until flush`

开始进入直写模式，在收到第一条刷新请求后切换到写回模式。启用此设置虽然较为保守，但却是一种安全的做法，如此可应对在 `rbd` 上运行的虚拟机太旧而无法发送刷新请求的情况（例如，内核低于 2.6.32 的 Linux 中的 Virtio 驱动程序）。默认值为“true”。

20.6 QoS 设置

一般而言，服务质量 (QoS) 指的是流量优先级设置和资源预留方法。它对于具有特殊要求的流量传输尤为重要。

重要：不受 iSCSI 支持

只有用户空间 RBD 实现 `librbd` 会使用下列 QoS 设置， **实现**不kRBD使用这些设置。由于 iSCSI 使用的是 kRBD，因此不使用 QoS 设置。不过，对于 iSCSI，您可以使用标准内核工具在内核块设备层上配置 QoS。

`rbd qos iops limit`

指定的每秒 I/O 操作次数上限。默认值为 0（无限制）。

`rbd qos bps limit`

指定的每秒 I/O 字节数上限。默认值为 0（无限制）。

`rbd qos read iops limit`

指定的每秒读操作次数上限。默认值为 0（无限制）。

`rbd qos write iops limit`

指定的每秒写操作次数上限。默认值为 0（无限制）。

`rbd qos read bps limit`

指定的每秒内读取的字节数上限。默认值为 0（无限制）。

`rbd qos write bps limit`

指定的每秒内写入的字节数上限。默认值为 0（无限制）。

`rbd qos iops burst`

指定的 I/O 操作次数突发上限。默认值为 0（无限制）。

`rbd qos bps burst`

指定的 I/O 字节数突发上限。默认值为 0（无限制）。

`rbd qos read iops burst`

指定的读操作次数突发上限。默认值为 0（无限制）。

`rbd qos write iops burst`

指定的写操作次数突发上限。默认值为 0（无限制）。

`rbd qos read bps burst`

指定的读取的字节数突发上限。默认值为 0（无限制）。

`rbd qos write bps burst`

指定的写入的字节数突发上限。默认值为 0（无限制）。

`rbd qos schedule tick min`

QoS 的最小时间表刻度（以毫秒为单位）。默认值为 50。

20.7 预读设置

RADOS 块设备支持预读/预提取功能，以优化小块的顺序读取。如果使用虚拟机，此操作通常应由 guest 操作系统处理，但引导加载器可能不会发出有效的读请求。如果禁用缓存，则会自动禁用预读功能。

重要：不受 iSCSI 支持

只有用户空间 RBD 实现 `librbd` 会使用下列预读设置， **实现**不kRBD使用这些设置。由于 iSCSI 使用的是 kRBD，因此不使用预读设置。不过，对于 iSCSI，您可以使用标准内核工具在内核块设备层上配置预读。

`rbd readahead trigger requests`

触发预读所必需的顺序读取请求数。默认值为 10。

`rbd readahead max bytes`

预读请求的最大大小。如果设置为 0，则会禁用预读功能。默认值为 512kB。

`rbd readahead disable after bytes`

从 RBD 映像读取该数量的字节后，该映像的预读功能将会禁用，直至其关闭。使用此设置，guest 操作系统引导时便可接管预读工作。如果设置为 0，预读将始终处于启用状态。默认值为 50 MB。

20.8 高级特性

RADOS 块设备支持可增强 RBD 映像功能的高级特性。您可以在创建 RBD 映像时在命令行上指定这些特性，或者在 Ceph 配置文件中使用 `rbd_default_features` 选项来指定。

您可以通过以下两种方式指定 `rbd_default_features` 选项的值：

- 指定为相应特性的内部值之和。每项特性都有自己的内部值，例如， “layering” 的内部值为 1， “fast-diff” 的内部值为 16。因此，若要默认激活这两项特性，请指定以下选项：

```
rbd_default_features = 17
```

- 指定为各特性的逗号分隔列表。上面的示例应如下所示：

```
rbd_default_features = layering,fast-diff
```

注意：iSCSI 不支持的特性

iSCSI 不支持具有以下特性的 RBD 映像：`deep-flatten`、`object-map`、`journaling`、`fast-diff`、`striping`

以下是 RBD 的高级特性列表：

`layering`

分层允许您使用克隆。

内部值为 1，默认值为“yes”。

`striping`

条带特性会将数据分布在多个对象之间，有助于提升顺序读取/写入工作负载的并行度。它可防止大型或繁忙的 RADOS 块设备出现单节点瓶颈。

内部值为 2，默认值为“yes”。

`exclusive-lock`

如果启用，客户端需要在写入数据之前锁定对象。仅当一次只有一个客户端在访问映像时才应启用互斥锁。内部值为 4。默认值为“yes”。

`object-map`

对象映射支持依赖于互斥锁支持。块设备采用的是精简供给，也就是说这些设备只存储实际存在的数据。对象映射支持有助于跟踪哪些对象实际存在（在驱动器上存储了数据）。启用对象映射支持可以加快克隆、导入和导出数据稀疏的映像以及删除所需的 I/O 操作。

内部值为 8，默认值为“yes”。

`fast-diff`

Fast-diff 支持依赖于对象映射支持和互斥锁支持。它会向对象映射添加另一个属性，使其更快地生成映像各快照之间的差异以及快照的实际数据使用率。

内部值为 16，默认值为“yes”。

`deep-flatten`

Deep-flatten 使 **`rbd flatten`**（请参见第 20.3.3.6 节 “平展克隆的映像”）除了对映像本身有效外，还对映像的所有快照有效。如果没有该特性，映像的快照将仍然依赖于其父映像，因而如果未删除快照，您将无法删除父映像。Deep-flatten 使父映像可独立于其克隆，即使这些克隆有快照也不例外。

内部值为 32，默认值为“yes”。

`journaling`

日志支持依赖于互斥锁支持。日志会按修改发生的顺序记录映像的所有修改。RBD 镜像（请参见第 20.4 节 “RBD 映像镜像”）会使用日志将崩溃一致性映像复制到远程集群。

内部值为 64，默认值为“no”。

20.9 使用旧内核客户端映射 RBD

旧客户端（例如 SLE11 SP4）可能无法映射 RBD 映像，因为使用 SUSE Enterprise Storage 7.1 部署的集群会强制执行一些旧客户端不支持的特性（RBD 映像级特性和 RADOS 级特性）。发生此情况时，OSD 日志将显示类似如下的消息：

```
2019-05-17 16:11:33.739133 7fcb83a2e700  0 -- 192.168.122.221:0/1006830 >> \
192.168.122.152:6789/0 pipe(0x65d4e0 sd=3 :57323 s=1 pgs=0 cs=0 l=1
 c=0x65d770).connect \
protocol feature mismatch, my 2fffffffffff < peer 4010ff8ffacffff missing
 401000000000000
```

警告：更改 CRUSH 索引存储桶类型将导致大规模重新平衡

如果您打算在 CRUSH 索引存储桶类型“straw”与“straw2”之间切换，请做好相应规划。这样做预计会对集群负载产生重大影响，因为更改存储桶类型将导致集群大规模重新平衡。

1. 禁用任何不支持的 RBD 映像特性。例如：

```
cephuser@adm > rbd feature disable pool1/image1 object-map
cephuser@adm > rbd feature disable pool1/image1 exclusive-lock
```

2. 将 CRUSH 索引存储桶类型由“straw2”更改为“straw”：

   a. 保存 CRUSH 索引：

   ```
   cephuser@adm > ceph osd getcrushmap -o crushmap.original
   ```

   b. 反编译 CRUSH 索引：

   ```
   cephuser@adm > crushtool -d crushmap.original -o crushmap.txt
   ```

   c. 编辑 CRUSH 索引，并用“straw”替换“straw2”。

   d. 重新编译 CRUSH 索引：

   ```
   cephuser@adm > crushtool -c crushmap.txt -o crushmap.new
   ```

   e. 设置新 CRUSH 索引：

   ```
   cephuser@adm > ceph osd setcrushmap -i crushmap.new
   ```

20.10 启用块设备和 Kubernetes

您可以通过 `ceph-csi` 驱动程序将 Ceph RBD 与 Kubernetes v1.13 及更高版本结合使用。此驱动程序会动态供给 RBD 映像以支持 Kubernetes 卷，并在引用 RBD 支持卷的工作节点运行中 pod 上作为块设备映射这些 RBD 映像（选择性地挂载映像中包含的文件系统）。

要将 Ceph 块设备与 Kubernetes 结合使用，必须在您的 Kubernetes 环境中安装和配置 `ceph-csi`。

重要

`ceph-csi` 默认使用 RBD 内核模块，它可能不支持所有 Ceph CRUSH 可调变量或 RBD 映像功能。

1. 默认情况下，Ceph 块设备使用 RBD 存储池。为 Kubernetes 卷存储创建存储池。确保 Ceph 集群正在运行，然后创建存储池：

```
cephuser@adm > ceph osd pool create kubernetes
```

2. 使用 RBD 工具初始化存储池：

```
cephuser@adm > rbd pool init kubernetes
```

3. 为 Kubernetes 和 `ceph-csi` 创建一个新用户。执行以下命令并记录生成的密钥：

```
cephuser@adm > ceph auth get-or-create client.kubernetes mon 'profile rbd'
 osd 'profile rbd pool=kubernetes' mgr 'profile rbd pool=kubernetes'
[client.kubernetes]
    key = AQD9o0Fd6hQRChAAt7fMaSZXduT3NWEqylNpmg==
```

4. `ceph-csi` 需要存储在 Kubernetes 中的 ConfigMap 对象来定义 Ceph 集群的 Ceph Monitor 地址。收集 Ceph 集群的唯一 fsid 和 Monitor 地址：

```
cephuser@adm > ceph mon dump
<...>
fsid b9127830-b0cc-4e34-aa47-9d1a2e9949a8
<...>
0: [v2:192.168.1.1:3300/0,v1:192.168.1.1:6789/0] mon.a
1: [v2:192.168.1.2:3300/0,v1:192.168.1.2:6789/0] mon.b
2: [v2:192.168.1.3:3300/0,v1:192.168.1.3:6789/0] mon.c
```

5. 生成 `csi-config-map.yaml` 文件，替换 `clusterID` 的 FSID，并替换 `monitors` 的 Monitor 地址，如下例所示：

```
kubectl@adm > cat <<EOF > csi-config-map.yaml
---
apiVersion: v1
kind: ConfigMap
data:
  config.json: |-
    [
      {
        "clusterID": "b9127830-b0cc-4e34-aa47-9d1a2e9949a8",
        "monitors": [
          "192.168.1.1:6789",
```

```
          "192.168.1.2:6789",
          "192.168.1.3:6789"
        ]
      }
    ]
metadata:
  name: ceph-csi-config
EOF
```

6. 生成后，将新的 ConfigMap 对象存储在 Kubernetes 中：

```
kubectl@adm > kubectl apply -f csi-config-map.yaml
```

7. `ceph-csi` 需要 cephx 身份凭证来与 Ceph 集群通讯。使用新创建的 Kubernetes 用户 ID 和 cephx 密钥生成 `csi-rbd-secret.yaml` 文件，如下例所示：

```
kubectl@adm > cat <<EOF > csi-rbd-secret.yaml
---
apiVersion: v1
kind: Secret
metadata:
  name: csi-rbd-secret
  namespace: default
stringData:
  userID: kubernetes
  userKey: AQD9o0Fd6hQRChAAt7fMaSZXduT3NWEqylNpmg==
EOF
```

8. 生成后，将新的秘密密钥对象存储在 Kubernetes 中：

```
kubectl@adm > kubectl apply -f csi-rbd-secret.yaml
```

9. 创建所需的 ServiceAccount 和 RBAC ClusterRole/ClusterRoleBinding Kubernetes 对象。这些对象不一定需要根据您的 Kubernetes 环境进行自定义，因此可以直接从 `ceph-csi` 部署 YAML 文件使用：

```
kubectl@adm > kubectl apply -f https://raw.githubusercontent.com/ceph/ceph-csi/master/deploy/rbd/kubernetes/csi-provisioner-rbac.yaml
```

```
kubectl@adm > kubectl apply -f https://raw.githubusercontent.com/ceph/ceph-
csi/master/deploy/rbd/kubernetes/csi-nodeplugin-rbac.yaml
```

10. 创建 `ceph-csi` 供给程序和节点插件：

```
kubectl@adm > wget https://raw.githubusercontent.com/ceph/ceph-csi/master/
deploy/rbd/kubernetes/csi-rbdplugin-provisioner.yaml
kubectl@adm > kubectl apply -f csi-rbdplugin-provisioner.yaml
kubectl@adm > wget https://raw.githubusercontent.com/ceph/ceph-csi/master/
deploy/rbd/kubernetes/csi-rbdplugin.yaml
kubectl@adm > kubectl apply -f csi-rbdplugin.yaml
```

重要

默认情况下，供给程序和节点插件 YAML 文件将提取 `ceph-csi` 容器的开发版本。应更新 YAML 文件以使用发布版本。

20.10.1 在 Kubernetes 中使用 Ceph 块设备

Kubernetes StorageClass 定义了一个存储类。可以创建多个 StorageClass 对象，以映射到不同的服务质量级别和功能。例如，NVMe 与基于 HDD 的存储池。

要创建映射到上面所创建 Kubernetes 存储池的 `ceph-csi` StorageClass，在确保 `clusterID` 属性与您的 Ceph 集群的 FSID 相匹配之后，可以使用以下 YAML文件：

```
kubectl@adm > cat <<EOF > csi-rbd-sc.yaml
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
   name: csi-rbd-sc
provisioner: rbd.csi.ceph.com
parameters:
   clusterID: b9127830-b0cc-4e34-aa47-9d1a2e9949a8
   pool: kubernetes
   csi.storage.k8s.io/provisioner-secret-name: csi-rbd-secret
   csi.storage.k8s.io/provisioner-secret-namespace: default
```

```
   csi.storage.k8s.io/node-stage-secret-name: csi-rbd-secret
   csi.storage.k8s.io/node-stage-secret-namespace: default
reclaimPolicy: Delete
mountOptions:
   - discard
EOF
kubectl@adm > kubectl apply -f csi-rbd-sc.yaml
```

`PersistentVolumeClaim` 是用户发出的抽象存储资源请求。然后，`PersistentVolumeClaim` 将与 pod 资源相关联以供应 `PersistentVolume`，它将由 Ceph 块映像提供支持。可以包含可选的 `volumeMode`，以便在挂载的文件系统（默认）或基于块设备的原始卷之间进行选择。

使用 `ceph-csi`，为 `volumeMode` 指定 `Filesystem` 可支持 `ReadWriteOnce` 和 `ReadOnlyMany accessMode` 声明，为 `volumeMode` 指定 `Block` 可支持 `ReadWriteOnce`、`ReadWriteMany` 和 `ReadOnlyMany accessMode` 声明。

例如，要创建使用上述创建的 `ceph-csi-based StorageClass` 的基于块的 `PersistentVolumeClaim`，可以使用以下 YAML 文件通过 `csi-rbd-sc StorageClass` 请求原始块存储：

```
kubectl@adm > cat <<EOF > raw-block-pvc.yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: raw-block-pvc
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Block
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-rbd-sc
EOF
kubectl@adm > kubectl apply -f raw-block-pvc.yaml
```

下面的示例演示了如何将上述 `PersistentVolumeClaim` 作为原始块设备绑定到 pod 资源：

```
kubectl@adm > cat <<EOF > raw-block-pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: pod-with-raw-block-volume
spec:
  containers:
    - name: fc-container
      image: fedora:26
      command: ["/bin/sh", "-c"]
      args: ["tail -f /dev/null"]
      volumeDevices:
        - name: data
          devicePath: /dev/xvda
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: raw-block-pvc
EOF
kubectl@adm > kubectl apply -f raw-block-pod.yaml
```

要创建使用上述创建的 `ceph-csi-based StorageClass` 的基于文件系统的 `PersistentVolumeClaim`，可以使用以下 YAML 文件通过 `csi-rbd-sc StorageClass` 请求挂载的文件系统（由 RBD 映像提供支持）：

```
kubectl@adm > cat <<EOF > pvc.yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rbd-pvc
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
```

```
      storage: 1Gi
  storageClassName: csi-rbd-sc
EOF
kubectl@adm > kubectl apply -f pvc.yaml
```

下面的示例演示了如何将上述 `PersistentVolumeClaim` 作为挂载的文件系统绑定到 pod 资源：

```
kubectl@adm > cat <<EOF > pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: csi-rbd-demo-pod
spec:
  containers:
    - name: web-server
      image: nginx
      volumeMounts:
        - name: mypvc
          mountPath: /var/lib/www/html
  volumes:
    - name: mypvc
      persistentVolumeClaim:
        claimName: rbd-pvc
        readOnly: false
EOF
kubectl@adm > kubectl apply -f pod.yaml
```

IV 访问集群数据

21 Ceph 对象网关 248

22 Ceph iSCSI 网关 303

23 集群化文件系统 320

24 通过 Samba 导出 Ceph 数据 331

25 NFS Ganesha 350

21 Ceph 对象网关

本章介绍对象网关相关管理任务的详细信息，例如，检查服务的状态，管理帐户、多站点网关或 LDAP 身份验证。

21.1 对象网关限制和命名限制

下面列出了对象网关的一些重要限制：

21.1.1 存储桶限制

通过 S3 API 访问对象网关时，存储桶名必须符合 DNS 且允许使用短划线字符“-”。当通过 Swift API 访问对象网关时，您可使用支持 UTF-8 的字符（斜杠字符“/”除外）的任何组合。存储桶名最多可包含 255 个字符。存储桶名必须唯一。

提示：使用符合 DNS 规范的存储桶名称

虽然通过 Swift API 访问时，可使用任何基于 UTF-8 的存储桶名，但仍建议您根据 S3 命名限制对存储桶命名，以免在通过 S3 API 访问同一个存储桶时发生问题。

21.1.2 存储的对象的限制

每个用户的对象数量上限

默认无限制（大约不超过 2^63）。

每个存储桶的对象数量上限

默认无限制（大约不超过 2^63）。

要上载/存储的对象的最大大小

单次上载的上限为 5GB。更大的对象可分为多个部分上载。多部分块的最大数量为 10000。

21.1.3 HTTP 报头限制

HTTP 报头和请求限制取决于所使用的 Web 前端。默认 Beast 会将 HTTP 报头大小限制为 16 kB。

21.2 部署对象网关

Ceph 对象网关采用与其他 Ceph 服务相同的过程进行部署，即使用 cephadm。有关详细信息，具体请参考《部署指南》, 第 8 章 “使用 cephadm 部署其余核心服务”, 第 8.2 节 “服务和归置规范”到《部署指南》, 第 8 章 “使用 cephadm 部署其余核心服务”, 第 8.3.4 节 “部署对象网关”。

21.3 操作对象网关服务

您可以像操作其他 Ceph 服务那样来操作对象网关，首先使用 **ceph orch ps** 命令标识服务名称，然后运行以下命令来操作服务，例如：

```
ceph orch daemon restart OGW_SERVICE_NAME
```

有关操作 Ceph 服务的完整信息，请参考第 14 章 “Ceph 服务的操作”。

21.4 配置选项

有关对象网关配置选项的列表，请参见第 28.5 节 “Ceph 对象网关”。

21.5 管理对象网关的访问方式

您可以使用与 S3 或 Swift 兼容的接口来与对象网关通讯。S3 接口与大部分 Amazon S3 RESTful API 都兼容。Swift 接口与大部分 OpenStack Swift API 都兼容。

这两个接口都要求创建特定的用户，并安装相关的客户端软件，以使用该用户的秘密密钥来与网关通讯。

21.5.1 访问对象网关

21.5.1.1 S3 接口访问

要访问 S3 接口，需要一个 REST 客户端。**S3cmd** 是一个命令行 S3 客户端。您可以在 OpenSUSE Build Service (https://build.opensuse.org/package/show/Cloud:Tools/s3cmd) 中找到它。该软件源包含既适用于 SUSE Linux Enterprise 发行套件又适用于基于 openSUSE 的发行套件的版本。

如果您想测试自己是否能够访问 S3 接口，也可以编写一个简短的 Python 脚本。该脚本将连接到对象网关，创建新存储桶，并列出所有存储桶。`aws_access_key_id` 和 `aws_secret_access_key` 的值取自第 21.5.2.1 节 “添加 S3 和 Swift 用户”中所述 **radosgw_admin** 命令返回的 `access_key` 和 `secret_key` 的值。

1. 安装 `python-boto` 软件包：

```
# zypper in python-boto
```

2. 创建名为 `s3test.py` 的新 Python 脚本，并在其中包含以下内容：

```
import boto
import boto.s3.connection
access_key = '11BS02LGFB6AL6H1ADMW'
secret_key = 'vzCEkuryfn060dfee4fgQPqFrncKEIkh3ZcdOANY'
conn = boto.connect_s3(
aws_access_key_id = access_key,
aws_secret_access_key = secret_key,
host = 'HOSTNAME',
is_secure=False,
calling_format = boto.s3.connection.OrdinaryCallingFormat(),
)
bucket = conn.create_bucket('my-new-bucket')
for bucket in conn.get_all_buckets():
  print "NAME\tCREATED".format(
  name = bucket.name,
  created = bucket.creation_date,
  )
```

请将 HOSTNAME 替换为在其中配置了对象网关服务的主机的主机名，例如 gateway_host。

3. 运行脚本：

```
python s3test.py
```

该脚本将输出类似下方所示的信息：

```
my-new-bucket 2015-07-22T15:37:42.000Z
```

21.5.1.2 Swift 接口访问

要通过 Swift 接口访问对象网关，需要使用 **swift** 命令行客户端。该接口的手册页 **man 1 swift** 介绍了有关其命令行选项的详细信息。

从 SUSE Linux Enterprise 12 SP3 到 SUSE Linux Enterprise 15 的“Public Cloud”扩展模块中都随附了该软件包。在安装该软件包之前，需要激活该扩展模块并刷新软件源：

```
# SUSEConnect -p sle-module-public-cloud/12/SYSTEM-ARCH
sudo zypper refresh
```

或

```
# SUSEConnect -p sle-module-public-cloud/15/SYSTEM-ARCH
# zypper refresh
```

要安装 **swift** 命令，请运行以下命令：

```
# zypper in python-swiftclient
```

使用以下语法进行 swift 访问：

```
> swift -A http://IP_ADDRESS/auth/1.0 \
-U example_user:swift -K 'SWIFT_SECRET_KEY' list
```

请将 IP_ADDRESS 替换为网关服务器的 IP 地址，将 _SECRET_KEY 替换为在**中针对** swift 用户执行 swiftradosgw-admin key create第 21.5.2.1 节 “添加 S3 和 Swift 用户” 命令所生成的输出中的相应值。

例如：

```
> swift -A http://gateway.example.com/auth/1.0 -U example_user:swift \
-K 'r5wWIxjOCeEO7DixD1FjTLmNYIViaC6JVhi3013h' list
```

输出为：

```
my-new-bucket
```

21.5.2 管理 S3 和 Swift 帐户

21.5.2.1 添加 S3 和 Swift 用户

需要创建用户、访问钥和机密才能让最终用户与网关交互。用户分两种类型：**用户**和**子用户**。与 S3 接口交互时使用**用户**，**子用户**是 Swift 接口的用户。每个子用户都与某个用户相关联。

要创建 Swift 用户，请执行以下步骤：

1. 要创建 Swift 用户（在我们的术语中称作**子用户**），需要先创建关联的**用户**。

   ```
   cephuser@adm > radosgw-admin user create --uid=USERNAME \
    --display-name="DISPLAY-NAME" --email=EMAIL
   ```

   例如：

   ```
   cephuser@adm > radosgw-admin user create \
      --uid=example_user \
      --display-name="Example User" \
      --email=penguin@example.com
   ```

2. 要创建用户的子用户（用于 Swift 接口），必须指定用户 ID（--uid=USERNAME）、子用户 ID 和该子用户的访问级别。

   ```
   cephuser@adm > radosgw-admin subuser create --uid=UID \
    --subuser=UID \
    --access=[ read | write | readwrite | full ]
   ```

   例如：

```
cephuser@adm > radosgw-admin subuser create --uid=example_user \
 --subuser=example_user:swift --access=full
```

3. 为用户生成秘密密钥。

```
cephuser@adm > radosgw-admin key create \
   --gen-secret \
   --subuser=example_user:swift \
   --key-type=swift
```

4. 这两个命令都会输出 JSON 格式的数据，其中显示了用户状态。请注意以下几行，并记住 `secret_key` 值：

```
"swift_keys": [
   { "user": "example_user:swift",
     "secret_key": "r5wWIxjOCeEO7DixD1FjTLmNYIViaC6JVhi3013h"}],
```

通过 S3 接口访问对象网关时，需要运行以下命令来创建 S3 用户：

```
cephuser@adm > radosgw-admin user create --uid=USERNAME \
 --display-name="DISPLAY-NAME" --email=EMAIL
```

例如：

```
cephuser@adm > radosgw-admin user create \
   --uid=example_user \
   --display-name="Example User" \
   --email=penguin@example.com
```

该命令还会创建用户的访问钥和秘密密钥。检查该命令输出中的 `access_key` 和 `secret_key` 关键字及其值：

```
[...]
 "keys": [
       { "user": "example_user",
         "access_key": "11BS02LGFB6AL6H1ADMW",
         "secret_key": "vzCEkuryfn060dfee4fgQPqFrncKEIkh3ZcdOANY"}],
 [...]
```

21.5.2.2 删除 S3 和 Swift 用户

删除 S3 用户与删除 Swift 用户的过程类似。不过，在删除 Swift 用户时，您可能需要同时删除该用户及其子用户。

要删除 S3 或 Swift 用户（包括其所有子用户），请在以下命令中指定 `user rm` 和用户 ID：

```
cephuser@adm > radosgw-admin user rm --uid=example_user
```

要删除子用户，请指定 `subuser rm` 和子用户 ID。

```
cephuser@adm > radosgw-admin subuser rm --uid=example_user:swift
```

可使用以下选项：

--purge-data

清除与该用户 ID 关联的所有数据。

--purge-keys

清除与该用户 ID 关联的所有密钥。

提示：删除子用户

删除某个子用户时，删除的是其对 Swift 接口的访问权限。该用户仍会保留在系统中。

21.5.2.3 更改 S3 和 Swift 用户的访问密钥与秘密密钥

访问网关时，`access_key` 和 `secret_key` 参数用于标识对象网关用户。更改现有用户密钥的过程与创建新用户密钥的过程相同，旧密钥将被重写。

对于 S3 用户，请运行以下命令：

```
cephuser@adm > radosgw-admin key create --uid=EXAMPLE_USER --key-type=s3 --gen-
access-key --gen-secret
```

对于 Swift 用户，请运行以下命令：

```
cephuser@adm > radosgw-admin key create --subuser=EXAMPLE_USER:swift --key-
type=swift --gen-secret
```

`--key-type=TYPE`

指定密钥的类型。值为 `swift` 或 `s3`。

`--gen-access-key`

生成随机访问钥（默认针对 S3 用户）。

`--gen-secret`

生成随机秘密密钥。

`--secret=KEY`

指定秘密密钥，例如手动生成的密钥。

21.5.2.4 启用用户配额管理

Ceph 对象网关允许您针对用户以及用户拥有的存储桶设置配额。配额包括一个存储桶中的最大对象数，以及最大存储大小 (MB)。

在启用用户配额之前，需要先设置该配额的参数：

```
cephuser@adm > radosgw-admin quota set --quota-scope=user --uid=EXAMPLE_USER \
 --max-objects=1024 --max-size=1024
```

`--max-objects`

指定最大对象数。指定负值会禁用检查。

`--max-size`

指定最大字节数。指定负值会禁用检查。

`--quota-scope`

设置配额的范围。选项包括 `bucket` 和 `user`。存储桶配额将应用到用户拥有的存储桶。用户配额将应用到用户。

设置用户配额后，可启用该配额：

```
cephuser@adm > radosgw-admin quota enable --quota-scope=user --uid=EXAMPLE_USER
```

要禁用配额，请执行以下命令：

```
cephuser@adm > radosgw-admin quota disable --quota-scope=user --uid=EXAMPLE_USER
```

要列出配额设置，请执行以下命令：

```
cephuser@adm > radosgw-admin user info --uid=EXAMPLE_USER
```

要更新配额统计数字，请执行以下命令：

```
cephuser@adm > radosgw-admin user stats --uid=EXAMPLE_USER --sync-stats
```

21.6 HTTP 前端

Ceph 对象网关支持两个嵌入式 HTTP 前端：**Beast** 和 **Civetweb**。

Beast 前端使用 Boost.Beast 库处理 HTTP 分析，使用 Boost.Asio 库处理异步网络 I/O。

Civetweb 前端使用 Civetweb HTTP 库，该库属于 Mongoose 的分支。

可使用 `rgw_frontends` 选项配置它们。有关配置选项的列表，请参见第 28.5 节 “Ceph 对象网关”。

21.7 为对象网关启用 HTTPS/SSL

要让对象网关可使用 SSL 进行安全通讯，您需要拥有 CA 颁发的证书，或创建自我签名证书。

21.7.1 创建自我签名证书

提示

如果您已拥有 CA 签名的有效证书，请跳过本节。

以下过程说明如何在 Salt 主控端上生成自我签名的 SSL 证书。

1. 如果您需要通过其他主体身份来识别您的对象网关，请将这些身份添加到 `/etc/ssl/openssl.cnf` 文件 `[v3_req]` 段落的 `subjectAltName` 选项中：

   ```
   [...]
   [ v3_req ]
   ```

```
subjectAltName = DNS:server1.example.com DNS:server2.example.com
[...]
```

提示：subjectAltName 中的 IP 地址

要在 `subjectAltName` 选项中使用 IP 地址而非域名，请将示例行替换为以下行：

```
subjectAltName = IP:10.0.0.10 IP:10.0.0.11
```

2. 使用 **`openssl`** 创建密钥和证书。输入需要包含在证书中的所有数据。建议您输入 FQDN 作为常用名。对证书签名前，确认“X509v3 Subject Alternative Name:”包含在请求的扩展中，并且生成的证书中设置了“X509v3 Subject Alternative Name:”。

```
root@master # openssl req -x509 -nodes -days 1095 \
 -newkey rsa:4096 -keyout rgw.key
 -out rgw.pem
```

3. 向证书文件追加密钥：

```
root@master # cat rgw.key >> rgw.pem
```

21.7.2 配置使用 SSL 的对象网关

要将对象网关配置为使用 SSL 证书，请使用 `rgw_frontends` 选项。例如：

```
cephuser@adm > ceph config set WHO rgw_frontends \
 beast ssl_port=443 ssl_certificate=config://CERT ssl_key=config://KEY
```

如果您未指定 `CERT` 和 `KEY` 配置键，对象网关服务将在以下配置键中查找 SSL 证书和密钥：

```
rgw/cert/RGW_REALM/RGW_ZONE.key
rgw/cert/RGW_REALM/RGW_ZONE.crt
```

如果您要覆盖默认的 SSL 密钥和证书位置，请使用以下命令将它们导入配置数据库：

```
ceph config-key set CUSTOM_CONFIG_KEY -i PATH_TO_CERT_FILE
```

然后通过 `config://` 指令使用您的自定义配置键。

21.8 同步模块

对象网关部署为多站点服务，您可以在不同区域之间镜像数据和元数据。**同步扩展模块**构建在多站点框架的基础上，可将数据和元数据转发到不同的外部层。利用同步扩展模块，可让系统在每当有数据发生更改（例如执行存储桶或用户创建等元数据操作）时即执行一组操作。当对象网关多站点更改最终在远程站点上保持一致时，更改将以异步方式传播。因而很多情况下都适合使用同步扩展模块，例如将对象存储备份到外部云集群、使用磁带机的自定义备份解决方案，或在 ElasticSearch 中为元数据编制索引。

21.8.1 配置同步模块

所有同步扩展模块的配置方式都是相似的。您需要创建一个新区域（请参见第 21.13 节 “多站点对象网关”了解更多详细信息）并为其设置 `--tier_type` 选项，例如针对云同步扩展模块设置 `--tier-type=cloud`：

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=ZONE-GROUP-NAME \
 --rgw-zone=ZONE-NAME \
 --endpoints=http://endpoint1.example.com,http://endpoint2.example.com, [...] \
 --tier-type=cloud
```

您可以使用以下命令来配置特定层：

```
cephuser@adm > radosgw-admin zone modify --rgw-zonegroup=ZONE-GROUP-NAME \
 --rgw-zone=ZONE-NAME \
 --tier-config=KEY1=VALUE1,KEY2=VALUE2
```

配置中的 `KEY` 指定您希望更新的配置变量，`VALUE` 指定该变量的新值。使用句点可访问嵌套的值。例如：

```
cephuser@adm > radosgw-admin zone modify --rgw-zonegroup=ZONE-GROUP-NAME \
 --rgw-zone=ZONE-NAME \
 --tier-config=connection.access_key=KEY,connection.secret=SECRET
```

您可以通过追加方括号“[]”及引用的条目来访问相应阵列条目。您可以使用方括号“[]”来添加新的阵列条目。索引值 -1 引用的是阵列中的最后一个条目。在同一个命令中无法创建新条目并再次引用该条目。例如，用于为以 `PREFIX` 开头的存储桶创建新配置的命令如下所示：

```
cephuser@adm > radosgw-admin zone modify --rgw-zonegroup=ZONE-GROUP-NAME \
 --rgw-zone=ZONE-NAME \
 --tier-config=profiles[].source_bucket=PREFIX'*'
cephuser@adm > radosgw-admin zone modify --rgw-zonegroup=ZONE-GROUP-NAME \
 --rgw-zone=ZONE-NAME \
 --tier-
config=profiles[-1].connection_id=CONNECTION_ID,profiles[-1].acls_id=ACLS_ID
```

提示：添加和删除配置条目

您可以使用 `--tier-config-add=KEY=VALUE` 参数来添加新的层配置条目。

您可以使用 `--tier-config-rm=KEY` 删除现有的条目。

21.8.2 同步区域

同步扩展模块配置对于区域而言位于本地。同步扩展模块会确定区域是要导出数据，还是只能使用已在另一区域中修改的数据。从 Luminous 版本开始，支持的同步插件有 `ElasticSearch`、`rgw` 和 `log`，其中 rgw 是在区域之间同步数据的默认同步插件，log 是记录远程区域中发生的元数据操作的普通同步插件。以下各节内容包含了使用 `ElasticSearch` 同步扩展模块的区域示例。其过程与配置任何其他同步插件的过程都相似。

注意：默认同步插件

`rgw` 是默认的同步插件，不需要进行明确配置。

21.8.2.1 要求和假设

我们假设已根据第 21.13 节 “多站点对象网关”中所述配置了一个简单的多站点，它由 `us-east` 和 `us-west` 这两个区域组成。现在，我们添加第三个区域 `us-east-es`，此区域只处理来自其他站点的元数据。此区域可与 `us-east` 位于同一 Ceph 集群中，也可位于不同的集群中。此区域只使用来自其他区域的元数据，此区域中的对象网关不会直接处理任何最终用户请求。

21.8.2.2 配置区域

1. 创建类似于第 21.13 节 “多站点对象网关”中所述区域的第三个区域，例如

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=us --rgw-zone=us-
east-es \
--access-key=SYSTEM-KEY --secret=SECRET --endpoints=http://rgw-es:80
```

2. 可通过以下命令为此区域配置同步扩展模块:

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --tier-
type=TIER-TYPE \
--tier-config={set of key=value pairs}
```

3. 例如，在 ElasticSearch 同步扩展模块中运行以下命令

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --tier-
type=elasticsearch \
--tier-config=endpoint=http://localhost:9200,num_shards=10,num_replicas=1
```

   有关支持的各个 tier-config 选项，请参见第 21.8.3 节 “ElasticSearch 同步模块”。

4. 最后，更新周期

```
cephuser@adm > radosgw-admin period update --commit
```

5. 现在，在区域中启动对象网关

```
cephuser@adm > ceph orch start rgw.REALM-NAME.ZONE-NAME
```

21.8.3 ElasticSearch 同步模块

此同步扩展模块会将来自其他区域的元数据写入 ElasticSearch。从 Luminous 版本开始，这些元数据就是我们当前存储在 ElasticSearch 中的数据字段的 JSON。

```
{
  "_index" : "rgw-gold-ee5863d6",
  "_type" : "object",
  "_id" : "34137443-8592-48d9-8ca7-160255d52ade.34137.1:object1:null",
```

```
  "_score" : 1.0,
  "_source" : {
    "bucket" : "testbucket123",
    "name" : "object1",
    "instance" : "null",
    "versioned_epoch" : 0,
    "owner" : {
      "id" : "user1",
      "display_name" : "user1"
    },
    "permissions" : [
      "user1"
    ],
    "meta" : {
      "size" : 712354,
      "mtime" : "2017-05-04T12:54:16.462Z",
      "etag" : "7ac66c0f148de9519b8bd264312c4d64"
    }
  }
}
```

21.8.3.1 ElasticSearch 层类型配置参数

endpoint

指定要访问的 ElasticSearch 服务器端点。

num_shards

（整数）数据同步初始化时将为 ElasticSearch 配置的分片数量。请注意，初始化之后将无法更改此数量。在此处进行任何更改都需要重构建 ElasticSearch 索引，并需要重新初始化数据同步进程。

num_replicas

（整数）数据同步初始化时将为 ElasticSearch 配置的副本数量。

explicit_custom_meta

(true | false) 指定是否将为所有用户自定义元数据编制索引，或者用户是否需要（在存储桶级别）配置应为哪些客户元数据项编制索引。此参数默认为 false

index_buckets_list

（逗号分隔的字符串列表）如果为空，则为所有存储桶编制索引。否则，只为此处指定的存储桶编制索引。可以提供存储桶前缀（例如“foo*”）或存储桶后缀（例如“*bar”）。

approved_owners_list

（逗号分隔的字符串列表）如果为空，将为所有所有者的存储桶编制索引（需遵守其他限制）；否则，将只为指定所有者拥有的存储桶编制索引。也可以提供后缀和前缀。

override_index_path

（字符串）如果非空，则此字符串将用作 ElasticSearch 索引路径。否则，将在同步初始化时确定并生成索引路径。

username

指定 ElasticSearch 的用户名（如果需要身份验证）。

password

指定 ElasticSearch 的密码（如果需要身份验证）。

21.8.3.2 元数据查询

由于 ElasticSearch 集群现在存储对象元数据，因此务必确保 ElasticSearch 端点不会向公众公开，只有集群管理员可访问它们。向最终用户自己公开元数据查询会造成问题，因为我们希望该用户只查询自己的元数据，而不能查询任何其他用户的元数据，这就要求 ElasticSearch 集群像 RGW 所做的那样来对用户进行身份验证，而这就导致了问题发生。

从 Luminous 版本开始，元数据主区域中的 RGW 可处理最终用户请求。这样就无需向公众公开 ElasticSearch 端点，同时也解决了身份验证和授权问题，因为 RGW 本身就能对最终用户请求进行身份验证。出于此目的，RGW 在存储桶 API 中引入了可处理 ElasticSearch 请求的新查询。所有这些请求必须发送到元数据主区域。

获取 ElasticSearch 查询

```
GET /BUCKET?query=QUERY-EXPR
```

请求参数：

- max-keys：要返回的最大项数
- marker：分页标识

```
expression := [(]<arg> <op> <value> [)][<and|or> ...]
```

运算符为下列其中一项：<、<=、==、>=、>

例如：

```
GET /?query=name==foo
```

将返回用户有权读取且名为“foo”的所有带索引键。输出内容将是 XML 格式的键列表，与 S3 的“列出存储桶”请求的响应类似。

配置自定义元数据字段

定义应该（在指定的存储桶中）为哪些自定义元数据项编制索引，以及这些键的类型是什么。如果配置了显式自定义元数据索引，则需要此定义，以便 rgw 为指定的自定义元数据值编制索引。如果未配置，在带索引元数据键的类型不是字符串的情况下，也需要此定义。

```
POST /BUCKET?mdsearch
x-amz-meta-search: <key [; type]> [, ...]
```

多个元数据字段必须用逗号加以分隔，可以使用分号“;”强制指定字段的类型。当前允许的类型有字符串（默认）、整数和日期。例如，如果您想要将自定义对象元数据 x-amz-meta-year 以整数类型编入索引，将 x-amz-meta-date 以日期类型编入索引，并将 x-amz-meta-title 以字符串类型编入索引，您需要运行以下命令：

```
POST /mybooks?mdsearch
x-amz-meta-search: x-amz-meta-year;int, x-amz-meta-release-date;date, x-
amz-meta-title;string
```

删除自定义元数据配置

删除自定义元数据存储桶配置。

```
DELETE /BUCKET?mdsearch
```

获取自定义元数据配置

检索自定义元数据存储桶配置。

```
GET /BUCKET?mdsearch
```

21.8.4 云同步模块

本节介绍用于将区域数据同步到远程云服务的扩展模块。该同步是单向的，无法将数据从远程区域同步回其本地区域。此扩展模块的主要目的是实现面向多个云服务提供商的数据同步，当前仅支持与 AWS (S3) 兼容的云提供程序。

要将数据同步到远程云服务，您需要配置用户身份凭证。由于许多云服务都对每个用户可创建的存储桶数量设定了限制，因此您可以配置源对象与存储桶的映射，使不同的目标映射至不同的存储桶和存储桶前缀。请注意，系统不会保留源访问列表 (ACL)。可以将特定源用户的权限映射到特定的目标用户。

由于 API 存在限制，因此无法保留原始对象修改时间和 HTTP 实体标记 (ETag)。云同步扩展模块会将它们存储为目标对象的元数据属性。

21.8.4.1 配置云同步模块

下面提供了一些普通和非普通云同步扩展模块配置示例。请注意，普通配置可能会与非普通配置发生冲突。

例 21.1： 普通配置

```
{
  "connection": {
    "access_key": ACCESS,
    "secret": SECRET,
    "endpoint": ENDPOINT,
    "host_style": path | virtual,
  },
  "acls": [ { "type": id | email | uri,
    "source_id": SOURCE_ID,
    "dest_id": DEST_ID } ... ],
  "target_path": TARGET_PATH,
}
```

例 21.2： 非普通配置

```
{
  "default": {
    "connection": {
      "access_key": ACCESS,
```

```
      "secret": SECRET,
      "endpoint": ENDPOINT,
      "host_style" path | virtual,
    },
    "acls": [
    {
      "type": id | email | uri,   #  optional, default is id
      "source_id": ID,
      "dest_id": ID
    } ... ]
    "target_path": PATH # optional
  },
  "connections": [
  {
    "connection_id": ID,
    "access_key": ACCESS,
    "secret": SECRET,
    "endpoint": ENDPOINT,
    "host_style": path | virtual,  # optional
  } ... ],
  "acl_profiles": [
  {
    "acls_id": ID, # acl mappings
    "acls": [ {
      "type": id | email | uri,
      "source_id": ID,
      "dest_id": ID
    } ... ]
  }
  ],
  "profiles": [
  {
   "source_bucket": SOURCE,
   "connection_id": CONNECTION_ID,
   "acls_id": MAPPINGS_ID,
   "target_path": DEST,          # optional
  } ... ],
}
```

下面是对所用配置术语的解释：

connection

表示与远程云服务的连接。包含“connection_id”、“access_key”、“secret”、“endpoint”和“host_style”。

access_key

将用于特定连接的远程云访问密钥。

secret

远程云服务的秘密密钥。

endpoint

远程云服务端点的 URL。

host_style

访问远程云端点时要使用的主机类型（“path”或“virtual”）。默认值为“path”。

acls

访问列表映射阵列。

acl_mapping

每个“acl_mapping”结构都包含“type”、“source_id”和“dest_id”。它们将定义每个对象的 ACL 变化。ACL 变化允许将源用户 ID 转换为目标 ID。

type

ACL 类型：“id”定义用户 ID，“email”按电子邮件定义用户，“uri”按 uri（组）定义用户。

source_id

源区域中的用户 ID。

dest_id

目标区域中的用户 ID。

target_path

用于定义目标路径创建方式的字符串。目标路径指定追加到源对象名称的前缀。可对目标路径进行配置，使其包含以下任何变量：

SID

表示同步实例 ID 的唯一字符串。

ZONEGROUP

区域组名称。

ZONEGROUP_ID

区域组 ID。

ZONE

区域名。

ZONE_ID

区域 ID。

BUCKET

源存储桶名称。

OWNER

源存储桶所有者 ID。

例如：target_path = rgwx-ZONE-SID/OWNER/BUCKET

acl_profiles

访问列表配置阵列。

acl_profile

每个配置均包含代表配置的“acls_id”和用于存放“acl_mappings”列表的“acls”阵列。

profiles

配置列表。每个配置均包含以下变量：

source_bucket

用于定义此配置源存储桶的存储桶名称或存储桶前缀（如果以 * 结尾）。

target_path

请参见上文的相关解释。

connection_id

将用于此配置的连接 ID。

acls_id

将用于此配置的 ACL 配置 ID。

21.8.4.2 专用于 S3 的可配置项

云同步扩展模块仅适用于与 AWS S3 兼容的后端。该扩展模块提供了一些可配置项，用来调整它在访问 S3 云服务时的行为：

```
{
  "multipart_sync_threshold": OBJECT_SIZE,
  "multipart_min_part_size": PART_SIZE
}
```

multipart_sync_threshold

针对大小大于或等于此值的对象，将使用多部分上载来与云服务进行同步。

multipart_min_part_size

通过多部分上载同步对象时所使用的部分大小下限。

21.8.5 存档同步模块

存档同步模块利用了对象网关中 S3 对象的版本控制功能。您可以配置一个**存档区域**，用来捕获其他区域中一段时间内出现的 S3 对象的不同版本。存档区域保留的版本历史只能通过与存档区域关联的网关来删除。

借助这样的体系结构，一些不受版本控制的区域可以通过自身的区域网关来镜像它们的数据和元数据，以便向最终用户提供高可用性，同时，存档区域会捕获所有数据更新，以便将它们合并为 S3 对象的版本。

通过在多区域配置中包含存档区域，您可以灵活地在一个区域中保留 S3 对象历史，同时能够节省其余区域中受版本控制的 S3 对象的副本原本需要的空间。

21.8.5.1 配置存档同步模块

提示：更多信息

有关配置多站点网关的详细信息，请参见第 21.13 节 “多站点对象网关”。

有关配置同步扩展模块的详细信息，请参见第 21.8 节 “同步模块”。

要使用存档同步模块，您需要创建一个新区域，并将其层类型设置为 `archive`：

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=ZONE_GROUP_NAME \
 --rgw-zone=OGW_ZONE_NAME \
 --endpoints=http://OGW_ENDPOINT1_URL[,http://OGW_ENDPOINT2_URL,...]
 --tier-type=archive
```

21.9 LDAP authentication

除了默认的本地用户身份验证以外，对象网关还能利用 LDAP 服务器服务来对用户进行身份验证。

21.9.1 身份验证机制

对象网关从令牌提取用户的 LDAP 身份凭证。可以基于用户名构造搜索过滤器。对象网关使用配置的服务帐户在目录中搜索匹配的项。如果找到了某个项，对象网关会尝试使用令牌中的密码绑定到所找到的判别名。如果身份凭证有效，绑定将会成功，并且对象网关会授予访问权限。

您可以通过将搜索范围设置为特定的组织单位，或者指定自定义搜索过滤器（例如，要求特定的组成员资格、自定义对象类或属性），来限制允许的用户。

21.9.2 要求

- **LDAP 或 Active Directory**：对象网关可访问的运行中 LDAP 实例。
- **服务帐户**：对象网关要使用且拥有搜索权限的 LDAP 身份凭证。
- **用户帐户**：LDAP 目录中的至少一个用户帐户。

重要：LDAP 用户和本地用户不能重叠

不得对本地用户以及要使用 LDAP 进行身份验证的用户使用相同的用户名。对象网关无法区分两者，会将它们视为同一个用户。

提示：健康检查

使用 **`ldapsearch`** 实用程序可校验服务帐户或 LDAP 连接。例如：

```
> ldapsearch -x -D "uid=ceph,ou=system,dc=example,dc=com" -W \
-H ldaps://example.com -b "ou=users,dc=example,dc=com" 'uid=*' dn
```

请务必在 Ceph 配置文件中使用相同的 LDAP 参数，以杜绝可能的问题。

21.9.3 将对象网关配置为使用 LDAP 身份验证

以下参数与 LDAP 身份验证有关：

`rgw_s3_auth_use_ldap`

将此选项设置为 `true` 会启用 S3 LDAP 身份验证。

`rgw_ldap_uri`

指定要使用的 LDAP 服务器。请务必使用 `ldaps://FQDN:PORT` 参数，以免公开传输明文身份凭证。

`rgw_ldap_binddn`

对象网关使用的服务帐户的判别名 (DN)。

`rgw_ldap_secret`

服务帐户的密码。

rgw_ldap_searchdn

指定在目录信息树中搜索用户的范围，可以是用户的组织单位，或某个更具体的组织单位 (OU)。

`rgw_ldap_dnattr`

在构造的搜索过滤器中用来匹配用户名的属性。根据所用的目录信息树 (DIT)，可能会是 `uid` 或 `cn`。

`rgw_search_filter`

如果未指定，则对象网关会使用 `rgw_ldap_dnattr` 设置自动构造搜索过滤器。使用此参数能非常灵活地缩小所允许用户列表的范围。有关详细信息，请参见第 21.9.4 节 “使用自定义搜索过滤器限制用户访问权限”。

21.9.4 使用自定义搜索过滤器限制用户访问权限

可通过两种方式使用 `rgw_search_filter` 参数。

21.9.4.1 用于进一步限制所构造的搜索过滤器的部分过滤器

部分过滤器的示例：

```
"objectclass=inetorgperson"
```

对象网关将照常使用令牌中的用户名和 `rgw_ldap_dnattr` 的值生成搜索过滤器。然后，构造的过滤器将与 `rgw_search_filter` 属性中的部分过滤器合并。根据所用的用户名和设置，最终的搜索过滤器可能会变成：

```
"(&(uid=hari)(objectclass=inetorgperson))"
```

在这种情况下，仅当在 LDAP 目录中找到了用户 “hari”，该用户具有对象类 “inetorgperson” 并且确实指定了有效密码时，才向他授予访问权限。

21.9.4.2 完整过滤器

完整过滤器必须包含 `USERNAME` 令牌，在尝试身份验证期间，该令牌将替换为用户名。在这种情况下，不再使用 `rgw_ldap_dnattr` 参数。例如，要将有效用户限制为特定的组，可使用以下过滤器：

```
"(&(uid=USERNAME)(memberOf=cn=ceph-users,ou=groups,dc=mycompany,dc=com))"
```

注意：memberOf 属性

在 LDAP 搜索中使用 `memberOf` 属性需要您实施的特定 LDAP 服务器提供服务器端支持。

21.9.5 生成用于 LDAP 身份验证的访问令牌

`radosgw-token` 实用程序基于 LDAP 用户名和密码生成访问令牌。它会输出 base-64 编码字符串，即实际的访问令牌。请使用偏好的 S3 客户端（请参见第 21.5.1 节 “访问对象网关”），将该令牌指定为访问钥，并使用空秘密密钥。

```
> export RGW_ACCESS_KEY_ID="USERNAME"
> export RGW_SECRET_ACCESS_KEY="PASSWORD"
cephuser@adm > radosgw-token --encode --ttype=ldap
```

重要：明文身份凭证

访问令牌是一个 base-64 编码的 JSON 结构，包含明文形式的 LDAP 身份凭证。

注意：Active Directory

对于 Active Directory，请使用 `--ttype=ad` 参数。

21.10 存储桶索引分片

对象网关将存储桶索引数据存储在索引池中，该池默认为 `.rgw.buckets.index`。如果将太多（成百上千个）对象放入单个存储桶中，并且不设置每个存储桶的最大对象数量配额 (`rgw bucket default quota max objects`)，索引池的性能可能会下降。**存储桶索引分片**可在允许每个存储桶中放入大量对象的同时，防止出现此类性能下降的情况。

21.10.1 存储桶索引重分片

如果随着存储桶的增大，其初始配置不再能满足需求，则需要对存储桶的索引池进行重分片。您可以使用自动联机存储桶索引重分片（请参见第 21.10.1.1 节 “动态重分片”），也可以手动脱机执行存储桶索引重分片（请参见第 21.10.1.2 节 “手动重分片”）。

21.10.1.1 动态重分片

从 SUSE Enterprise Storage 5 开始，我们支持联机存储桶重分片。此功能会检测每个存储桶的对象数量是否达到某个阈值，如果达到该阈值，将会相应地自动增加存储桶索引使用的分片数量。此进程会减少每个存储桶索引分片中的条目数。

该检测进程在以下情况和环境中运行：

- 当有新的对象添加到存储桶中时。
- 在定期扫描所有存储桶的后台进程中。扫描的目的是为了处理未在更新的现有存储桶。

需要重分片的存储桶将会添加到 `reshard_log` 队列，且将安排于稍后进行重分片。重分片线程在后台运行，将逐个执行已安排的重分片。

配置动态重分片

`rgw_dynamic_resharding`
: 启用或禁用动态存储桶索引重分片。可用的值为“true”或“false”。默认设为“true”。

`rgw_reshard_num_logs`
: 重分片日志的分片数。默认设为 16。

`rgw_reshard_bucket_lock_duration`
: 重分片期间将存储桶对象锁定的时长。默认设为 120 秒。

`rgw_max_objs_per_shard`
: 每个存储桶索引分片的最大对象数。默认设为 100000 个对象。

`rgw_reshard_thread_interval`
: 两轮重分片线程处理间隔的最长时间。默认设为 600 秒。

用于管理重分片过程的命令

将存储桶添加到重分片队列：

```
cephuser@adm > radosgw-admin reshard add \
 --bucket BUCKET_NAME \
 --num-shards NEW_NUMBER_OF_SHARDS
```

列出重分片队列：

```
cephuser@adm > radosgw-admin reshard list
```

处理/安排存储桶重分片：

```
cephuser@adm > radosgw-admin reshard process
```

显示存储桶重分片状态：

```
cephuser@adm > radosgw-admin reshard status --bucket BUCKET_NAME
```

取消待处理的存储桶重分片：

```
cephuser@adm > radosgw-admin reshard cancel --bucket BUCKET_NAME
```

21.10.1.2 手动重分片

第 21.10.1.1 节 “动态重分片”所述的动态重新分片仅适用于简单对象网关配置。对于多站点配置，请使用本节中所述的手动重新分片。

要手动对存储桶索引执行脱机重分片，请使用以下命令：

```
cephuser@adm > radosgw-admin bucket reshard
```

`bucket reshard` 命令执行以下操作：

- 为指定对象创建一组新的存储桶索引对象。
- 分散这些索引对象的所有条目。
- 创建新的存储桶实例。

- 列出新的存储桶实例以及存储桶，以便所有新的索引操作都能够应用到新的存储桶索引。
- 将旧的和新的存储桶 ID 打印到标准输出。

提示

选择分片数量时，请注意以下几点：确保每个分片不超过 100000 个条目。如果在各分片中均匀分布存储桶索引条目，使用质数数量的存储桶索引分片通常效果会更好。例如，使用 503 个存储桶索引分片会比使用 500 个效果好，因为前者为质数。

过程 21.1： 对存储桶索引重分片

1. 确保对存储桶执行的所有操作都已停止。
2. 备份原始存储桶索引：

```
cephuser@adm > radosgw-admin bi list \
 --bucket=BUCKET_NAME \
 > BUCKET_NAME.list.backup
```

3. 对存储桶索引重分片：

```
cephuser@adm > radosgw-admin bucket reshard \
--bucket=BUCKET_NAME \
--num-shards=NEW_SHARDS_NUMBER
```

提示：旧存储桶 ID

此命令还会将新的和旧的存储桶 ID 打印到其输出中。

21.10.2 新存储桶的存储桶索引分片

有两个选项会影响存储桶索引分片：

- 对于简单配置，请使用 `rgw_override_bucket_index_max_shards` 选项。
- 对于多站点配置，请使用 `bucket_index_max_shards` 选项。

将选项设为 `0` 将禁用存储桶索引分片。如果将其设为大于 `0` 的值，则会启用存储桶索引分片，并设置最大分片数。

下面的公式可帮助您计算建议的分片数：

```
number_of_objects_expected_in_a_bucket / 100000
```

注意，分片的最大数量为 7877。

21.10.2.1 多站点配置

多站点配置可使用另一个索引池来管理故障转移。要为一个区域组内的区域配置一致的分片数量，请在该区域组的配置中设置 `bucket_index_max_shards` 选项：

1. 将区域组配置导出到 `zonegroup.json` 文件：

   ```
   cephuser@adm > radosgw-admin zonegroup get > zonegroup.json
   ```

2. 编辑 `zonegroup.json` 文件，为每个指定的区域设置 `bucket_index_max_shards` 选项。

3. 重新设置区域组：

   ```
   cephuser@adm > radosgw-admin zonegroup set < zonegroup.json
   ```

4. 更新周期。请参见第 21.13.2.6 节 “更新周期”。

21.11 OpenStack Keystone 集成

OpenStack Keystone 是一项用于 OpenStack 产品的身份服务。您可以将对象网关与 Keystone 相集成，以设置接受 Keystone 身份验证令牌的网关。Ceph 对象网关端将会对 Keystone 授权可访问网关的用户进行校验，并视需要自动创建用户。对象网关会定期查询 Keystone，以获取已撤消令牌列表。

21.11.1 配置 OpenStack

配置 Ceph 对象网关前，需要先配置 OpenStack Keystone 以启用 Swift 服务，并将其指向 Ceph 对象网关：

1. **设置 Swift 服务。**要使用 OpenStack 来验证 Swift 用户，请先创建 Swift 服务：

```
> openstack service create \
 --name=swift \
 --description="Swift Service" \
 object-store
```

2. **设置端点。**创建 Swift 服务后，指向 Ceph 对象网关。用网关的区域组名或区域名称替换 `REGION_NAME`。

```
> openstack endpoint create --region REGION_NAME \
 --publicurl   "http://radosgw.example.com:8080/swift/v1" \
 --adminurl    "http://radosgw.example.com:8080/swift/v1" \
 --internalurl "http://radosgw.example.com:8080/swift/v1" \
 swift
```

3. **校验这些设置。**创建 Swift 服务并设置端点后，显示端点以确认所有设置正确无误。

```
> openstack endpoint show object-store
```

21.11.2 配置 Ceph 对象网关

21.11.2.1 配置 SSL 证书

Ceph 对象网关会定期查询 Keystone，以获取已撤消令牌列表。这些请求会被编码并签名。还可配置 Keystone 以提供自我签名令牌，这些令牌同样经过编码和签名。您需要配置网关以便其可以解码并校验这些已签名消息。因此，需要将 Keystone 用于创建请求的 OpenSSL 证书转换为“nss db”格式：

```
# mkdir /var/ceph/nss
# openssl x509 -in /etc/keystone/ssl/certs/ca.pem \
```

```
 -pubkey | certutil -d /var/ceph/nss -A -n ca -t "TCu,Cu,Tuw"
rootopenssl x509 -in /etc/keystone/ssl/certs/signing_cert.pem \
 -pubkey | certutil -A -d /var/ceph/nss -n signing_cert -t "P,P,P"
```

为允许 Ceph 对象网关与 OpenStack Keystone 交互，OpenStack Keystone 可使用自我签名的 SSL 证书。可在运行 Ceph 对象网关的节点上安装 Keystone 的 SSL 证书，也可以将选项 `rgw keystone verify ssl` 的值设为“false”。将 `rgw keystone verify ssl` 设为“false”意味着网关将不会尝试校验证书。

21.11.2.2 配置对象网关的选项

您可以使用以下选项配置 Keystone 集成：

`rgw keystone api version`
: Keystone API 的版本。有效选项为 2 或 3。默认设为 2。

`rgw keystone url`
: Keystone 服务器上的管理 RESTful API 的 URL 和端口号。采用 `SERVER_URL:PORT_NUMBER` 模式。

`rgw keystone admin token`
: 在 Keystone 内部为管理请求配置的令牌或共享密钥。

`rgw keystone accepted roles`
: 处理请求需要具有的角色。默认设为“Member, admin”。

`rgw keystone accepted admin roles`
: 允许用户获取管理特权的角色列表。

`rgw keystone token cache size`
: Keystone 令牌缓存中的最大条目数。

`rgw keystone revocation interval`
: 检查已撤消令牌前间隔的秒数。默认设为 15 * 60。

`rgw keystone implicit tenants`
: 在各自的同名租户中创建新用户。默认设为“false”。

`rgw s3 auth use keystone`

如果设为“true”，Ceph 对象网关将使用 Keystone 对用户进行身份验证。默认设为“false”。

`nss db path`

NSS 数据库的路径。

还可以配置 Keystone 服务租户、Keystone 的用户和密码（适用于 OpenStack Identity API 2.0 版本），配置方法与配置 OpenStack 服务的方法类似。使用此方法可避免在配置文件中设置共享密钥 `rgw keystone admin token`，生产环境中应禁用该共享密钥。服务租户身份凭证应具有管理员特权。有关更多详细信息，请参考官方 OpenStack Keystone 文档 (https://docs.openstack.org/keystone/latest/#setting-up-projects-users-and-roles)。相关配置选项如下：

`rgw keystone admin user`

Keystone 管理员用户名。

`rgw keystone admin password`

Keystone 管理员用户密码。

`rgw keystone admin tenant`

Keystone 2.0 版管理员用户租户。

Ceph 对象网关用户与 Keystone 租户一一映射。系统会为一个 Keystone 用户指定不同的角色，这些角色可能分布在不止一个租户上。当 Ceph 对象网关收到票据时，会查看为该票据指定的租户和用户角色，并根据 `rgw keystone accepted roles` 选项的设置接受或拒绝请求。

提示：映射到 OpenStack 租户

虽然 Swift 租户默认会映射到对象网关用户，但也可通过 `rgw keystone implicit tenants` 选项将其映射到 OpenStack 租户。如此会让容器使用租户名称空间，而不是对象网关默认采用的 S3 之类的全局名称空间。建议在规划阶段就决定好映射方法，以免产生混淆。这是因为以后切换选项只会影响租户下所映射的较新请求，而先前创建的旧存储桶仍将继续放在全局名称空间中。

对于 OpenStack Identity API 3 版本，您应使用以下选项替换 `rgw keystone admin tenant` 选项：

`rgw keystone admin domain`
: Keystone 管理员用户域。

`rgw keystone admin project`
: Keystone 管理员用户项目。

21.12 存储池归置和存储类别

21.12.1 显示归置目标

归置目标用于控制哪些存储池与特定存储桶相关联。存储桶的归置目标是在创建存储桶时选择的，无法修改。您可通过运行以下命令来显示存储桶的 `placement_rule`：

```
cephuser@adm > radosgw-admin bucket stats
```

区域组配置包含一个归置目标列表，其中的初始目标名为“default-placement”。区域配置随后会将每个区域组归置目标名称映射到其本地存储。此区域归置信息包括表示存储桶索引的“index_pool”名称、表示不完整多部分上载相关元数据的“data_extra_pool”名称，以及表示每个存储类别的“data_pool”名称。

21.12.2 存储类别

存储类别有助于自定义对象数据的归置。S3 存储桶生命周期规则可自动转换对象的存储类别。

存储类别根据归置目标定义。每个区域组归置目标会列出它的可用存储类别，其中初始类别名为“STANDARD”。区域配置负责为每个区域组的存储类别提供“data_pool”存储池名称。

21.12.3 配置区域组和区域

您可以对区域组和区域使用 **`radosgw-admin`** 命令来配置其归置。可以使用以下命令来查询区域组归置配置：

```
cephuser@adm > radosgw-admin zonegroup get
{
    "id": "ab01123f-e0df-4f29-9d71-b44888d67cd5",
    "name": "default",
    "api_name": "default",
    ...
    "placement_targets": [
        {
            "name": "default-placement",
            "tags": [],
            "storage_classes": [
                "STANDARD"
            ]
        }
    ],
    "default_placement": "default-placement",
    ...
}
```

要查询区域归置配置，请运行以下命令：

```
cephuser@adm > radosgw-admin zone get
{
    "id": "557cdcee-3aae-4e9e-85c7-2f86f5eddb1f",
    "name": "default",
    "domain_root": "default.rgw.meta:root",
    ...
    "placement_pools": [
        {
            "key": "default-placement",
            "val": {
                "index_pool": "default.rgw.buckets.index",
                "storage_classes": {
                    "STANDARD": {
                        "data_pool": "default.rgw.buckets.data"
                    }
                },
                "data_extra_pool": "default.rgw.buckets.non-ec",
                "index_type": 0
```

```
            }
        }
    ],
    ...
}
```

注意：之前未进行多站点配置

如果您之前未进行过任何多站点配置，系统将会为您创建“default”区域和区域组，并且您必须重启动 Ceph 对象网关，对该区域/区域组所做的更改才会生效。如果您已创建过多站点领域，则当您使用 **`radosgw-admin period update --commit`** 命令提交对区域/区域组的更改后，更改即会生效。

21.12.3.1 添加归置目标

要创建一个名为“temporary”的新归置目标，首先请将该归置目标添加到区域组：

```
cephuser@adm > radosgw-admin zonegroup placement add \
      --rgw-zonegroup default \
      --placement-id temporary
```

然后为该目标提供区域归置信息：

```
cephuser@adm > radosgw-admin zone placement add \
      --rgw-zone default \
      --placement-id temporary \
      --data-pool default.rgw.temporary.data \
      --index-pool default.rgw.temporary.index \
      --data-extra-pool default.rgw.temporary.non-ec
```

21.12.3.2 添加存储类别

要向“default-placement”目标添加一个名为“COLD”的新存储类别，首先请将该存储类别添加到区域组：

```
cephuser@adm > radosgw-admin zonegroup placement add \
```

```
      --rgw-zonegroup default \
      --placement-id default-placement \
      --storage-class COLD
```

然后为该存储类别提供区域归置信息：

```
cephuser@adm > radosgw-admin zone placement add \
      --rgw-zone default \
      --placement-id default-placement \
      --storage-class COLD \
      --data-pool default.rgw.cold.data \
      --compression lz4
```

21.12.4 归置自定义

21.12.4.1 编辑默认区域组归置

新存储桶默认将使用区域组的 `default_placement` 目标。您可以使用以下命令更改此区域组设置：

```
cephuser@adm > radosgw-admin zonegroup placement default \
      --rgw-zonegroup default \
      --placement-id new-placement
```

21.12.4.2 编辑默认用户归置

Ceph 对象网关用户可以通过在用户信息中设置非空的 `default_placement` 字段来覆盖区域组的默认归置目标。同样，`default_storage_class` 也可以覆盖默认应用于对象的 `STANDARD` 存储类别。

```
cephuser@adm > radosgw-admin user info --uid testid
{
    ...
    "default_placement": "",
    "default_storage_class": "",
```

```
    "placement_tags": [],
    ...
}
```

如果区域组的归置目标包含任何标记，则除非用户信息的“placement_tags”字段中至少包含一个相匹配的标记，否则用户将无法创建具有该归置目标的存储桶。这有助于限制对某些特定存储类型的访问。

radosgw-admin 无法直接修改这些字段，因此您需要手动编辑 JSON 格式：

```
cephuser@adm > radosgw-admin metadata get user:USER-ID > user.json
> vi user.json     # edit the file as required
cephuser@adm > radosgw-admin metadata put user:USER-ID < user.json
```

21.12.4.3 编辑 S3 默认存储桶归置

采用 S3 协议创建存储桶时，可在 `LocationConstraint` 选项中提供归置目标，以覆盖用户和区域组的默认归置目标。

通常，`LocationConstraint` 需要与区域组的 `api_name` 相匹配：

```
<LocationConstraint>default</LocationConstraint>
```

您可以向 `api_name` 添加自定义归置目标（需在该自定义目标前加上一个冒号）：

```
<LocationConstraint>default:new-placement</LocationConstraint>
```

21.12.4.4 编辑 Swift 存储桶归置

采用 Swift 协议创建存储桶时，您可以在 HTTP 报头的 `X-Storage-Policy` 中提供归置目标：

```
X-Storage-Policy: NEW-PLACEMENT
```

21.12.5 使用存储类别

所有归置目标均具有 `STANDARD` 存储类别，这是系统默认为新对象应用的存储类别。您可以使用对象的 `default_storage_class` 覆盖此默认值。

要创建不属于默认存储类别的对象，请在请求的 HTTP 报头中提供所需的存储类别名称。S3 协议使用 `X-Amz-Storage-Class` 报头，而 Swift 协议则使用 `X-Object-Storage-Class` 报头。

借助 **S3 对象生命周期管理**，您可以使用转换操作来转变对象数据的存储类别。

21.13 多站点对象网关

Ceph 支持为 Ceph 对象网关选择多种多站点配置选项：

多区域

由一个区域组和多个区域组成的配置，每个区域有一个或多个 `ceph-radosgw` 实例。每个区域由各自的 Ceph 存储集群提供支持。当区域组中的其中一个区域出现严重故障时，该区域组中的其他区域可为区域组提供灾难恢复。每个区域都处于活动状态且可接收写入操作。除了灾难恢复外，多个活动区域还可作为内容分发网络的基础。

多区域组

Ceph 对象网关支持多个区域组，每个区域组有一个或多个区域。对象会存储到一个区域组中的多个区域，而同属于一个领域的另一个区域组会共享全局对象名称空间，以确保对象 ID 在各区域组和区域间保持唯一。

注意

请务必注意，区域组**只**会在各区域组之间同步元数据。区域组内的各个区域之间会复制数据和元数据。但任何数据和元数据均不会在领域中共享。

多个领域

Ceph 对象网关支持领域（即全局唯一的名称空间）概念。支持使用多个领域，每个领域可包含一个或多个区域组。

您可以将每个对象网关配置为以主动/主动区域配置运行，以便允许向非主区域写入数据。多站点配置存储在称为领域的容器内。领域中会存储区域组、区域和包含多个版本号（用于跟踪对配置进行的更改）的时间周期。`rgw` 守护进程负责处理同步，因而不需要另外再使用同步代理。这种同步方法可让 Ceph 对象网关以主动/主动配置（而非主动/被动配置）运行。

21.13.1 要求和假设

多站点配置至少需要两个 Ceph 存储集群以及两个 Ceph 对象网关实例，每个 Ceph 对象网关实例对应一个 Ceph 存储集群。以下配置假设至少有两个 Ceph 存储集群分布在不同的地理位置。但该配置也可在同一站点运行。例如，有两个名为 `rgw1` 和 `rgw2` 的主机。

多站点配置需要有一个主区域组和一个主区域。主区域是有关多站点集群中所有元数据操作的真实数据来源。此外，每个区域组都需要有一个主区域。区域组可以有一个或多个次要区域或非主区域。在本指南中，`rgw1` 主机作为主区域组的主区域，`rgw2` 主机作为主区域组的次要区域。

21.13.2 配置主区域

多站点配置中的所有网关都需从主区域组和主区域内的主机上的 `ceph-radosgw` 守护进程检索其配置。要在多站点配置中配置您的网关，请选择 `ceph-radosgw` 实例以配置主区域组和主区域。

21.13.2.1 创建领域

领域表示全局唯一的名称空间，由包含一个或多个区域的一个或多个区域组组成。区域包含存储桶，而存储桶包含对象。借助领域，Ceph 对象网关可以在相同硬件上支持多个名称空间及其配置。领域包含周期概念。每个周期表示不同时间的区域组和区域配置状态。每次您更改区域组或区域时，请更新周期并提交更新。出于向后兼容的原因，Ceph 对象网关默认不会创建领域。作为最佳实践，我们建议您为新集群创建领域。

要为多站点配置创建名为 `gold` 的新领域，请在标识为用于主区域组和主区域中的主机上打开命令行界面。然后执行以下命令：

```
cephuser@adm > radosgw-admin realm create --rgw-realm=gold --default
```

如果集群仅有一个领域，请指定 `--default` 标志。如果指定了 `--default`，**`radosgw-admin`** 默认会使用此领域。如果未指定 `--default`，添加区域组和区域时需要指定 `--rgw-realm` 或 `--realm-id` 标志以标识相应的领域。

创建领域后，使用 **`radosgw-admin`** 可返回领域配置：

```
{
```

```
  "id": "4a367026-bd8f-40ee-b486-8212482ddcd7",
  "name": "gold",
  "current_period": "09559832-67a4-4101-8b3f-10dfcd6b2707",
  "epoch": 1
}
```

注意

Ceph 会为领域生成唯一的 ID，因此允许用户在需要时重命名领域。

21.13.2.2 创建主区域组

领域必须至少有一个区域组作为该领域的主区域组。要为多站点配置创建新的主区域组，请在标识为用于主区域组和主区域中的主机上打开命令行界面。通过执行以下命令创建名为 `us` 的主区域组：

```
cephuser@adm > radosgw-admin zonegroup create --rgw-zonegroup=us \
--endpoints=http://rgw1:80 --master --default
```

如果领域仅有一个区域组，请指定 `--default` 标志。如果指定了 `--default`，**`radosgw-admin`** 在添加新区域时默认会使用此区域组。如果未指定 `--default`，添加区域时需要指定 `--rgw-zonegroup` 或 `--zonegroup-id` 标志以标识添加或修改区域时的相应区域组。

创建主区域组后，使用 **`radosgw-admin`** 可返回区域组配置。例如：

```
{
 "id": "d4018b8d-8c0d-4072-8919-608726fa369e",
 "name": "us",
 "api_name": "us",
 "is_master": "true",
 "endpoints": [
     "http:\/\/rgw1:80"
 ],
 "hostnames": [],
 "hostnames_s3website": [],
 "master_zone": "",
 "zones": [],
 "placement_targets": [],
```

```
 "default_placement": "",
 "realm_id": "4a367026-bd8f-40ee-b486-8212482ddcd7"
}
```

21.13.2.3 创建主区域

重要

需要在将要置于区域内的 Ceph 对象网关节点上创建区域。

要为多站点配置创建新的主区域，请在标识为用于主区域组和主区域中的主机上打开命令行界面。执行以下命令：

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=us --rgw-zone=us-east-1
 \
--endpoints=http://rgw1:80 --access-key=SYSTEM_ACCESS_KEY --
secret=SYSTEM_SECRET_KEY
```

注意

上面的示例中未指定 `--access-key` 和 `--secret` 选项。在下一节中创建用户时会将这些设置添加到区域中。

创建主区域后，使用 **`radosgw-admin`** 可返回区域配置。例如：

```
 {
     "id": "56dfabbb-2f4e-4223-925e-de3c72de3866",
     "name": "us-east-1",
     "domain_root": "us-east-1.rgw.meta:root",
     "control_pool": "us-east-1.rgw.control",
     "gc_pool": "us-east-1.rgw.log:gc",
     "lc_pool": "us-east-1.rgw.log:lc",
     "log_pool": "us-east-1.rgw.log",
     "intent_log_pool": "us-east-1.rgw.log:intent",
     "usage_log_pool": "us-east-1.rgw.log:usage",
     "reshard_pool": "us-east-1.rgw.log:reshard",
```

```
      "user_keys_pool": "us-east-1.rgw.meta:users.keys",
      "user_email_pool": "us-east-1.rgw.meta:users.email",
      "user_swift_pool": "us-east-1.rgw.meta:users.swift",
      "user_uid_pool": "us-east-1.rgw.meta:users.uid",
      "otp_pool": "us-east-1.rgw.otp",
      "system_key": {
          "access_key": "1555b35654ad1656d804",
          "secret_key":
"h7GhxuBLTrlhVUyxSPUKUV8r/2EI4ngqJxD7iBdBYLhwluN30JaT3Q=="
      },
      "placement_pools": [
          {
              "key": "us-east-1-placement",
              "val": {
                  "index_pool": "us-east-1.rgw.buckets.index",
                  "storage_classes": {
                      "STANDARD": {
                          "data_pool": "us-east-1.rgw.buckets.data"
                      }
                  },
                  "data_extra_pool": "us-east-1.rgw.buckets.non-ec",
                  "index_type": 0
              }
          }
      ],
      "metadata_heap": "",
      "realm_id": ""
  }
```

21.13.2.4 删除默认的区域和组

重要

下面的步骤假设多站点配置使用的是新安装的系统，即其中尚未存储任何数据。如果您已在使用默认区域及其存储池存储数据，**切勿删除**该区域及其存储池，否则数据将被删除且不可恢复。

采用默认设置安装对象网关时会创建名为 `default` 的默认区域组。删除默认区域（如果存在）。务必先将其从默认区域组中删除。

```
cephuser@adm > radosgw-admin zonegroup delete --rgw-zonegroup=default
```

删除 Ceph 存储集群中的默认存储池（如果存在）：

重要

下面的步骤假设多站点配置使用的是新安装的系统，即当前未存储任何数据。如果您已在使用默认区域组存储数据，**切勿删除**该区域组。

```
cephuser@adm > ceph osd pool rm default.rgw.control default.rgw.control --yes-i-
really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.data.root default.rgw.data.root --
yes-i-really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.gc default.rgw.gc --yes-i-really-
really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.log default.rgw.log --yes-i-really-
really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.meta default.rgw.meta --yes-i-
really-really-mean-it
```

警告

如果您删除默认区域组，则将同时删除系统用户。如果未传播您的管理员用户密钥，Ceph Dashboard 的对象网关管理功能将会失败。如果您执行此步骤，请继续下一节以重新创建您的系统用户。

21.13.2.5 创建系统用户

`ceph-radosgw` 守护进程在提取领域和周期信息前必须进行身份验证。在主区域中，创建系统用户以简化守护进程之间的身份验证：

```
cephuser@adm > radosgw-admin user create --uid=zone.user \
```

```
--display-name="Zone User" --access-key=SYSTEM_ACCESS_KEY \
--secret=SYSTEM_SECRET_KEY --system
```

记下 `access_key` 和 `secret_key`，因为次要区域需要使用它们向主区域进行身份验证。

将系统用户添加到主区域：

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=us-east-1 \
--access-key=ACCESS-KEY --secret=SECRET
```

更新周期以使更改生效：

```
cephuser@adm > radosgw-admin period update --commit
```

21.13.2.6 更新周期

更新主区域配置后，请更新周期：

```
cephuser@adm > radosgw-admin period update --commit
```

更新周期后，使用 **`radosgw-admin`** 可返回周期配置。例如：

```
{
  "id": "09559832-67a4-4101-8b3f-10dfcd6b2707", "epoch": 1, "predecessor_uuid":
 "", "sync_status": [], "period_map":
  {
    "id": "09559832-67a4-4101-8b3f-10dfcd6b2707", "zonegroups": [],
 "short_zone_ids": []
  }, "master_zonegroup": "", "master_zone": "", "period_config":
  {
     "bucket_quota": {
     "enabled": false, "max_size_kb": -1, "max_objects": -1
     }, "user_quota": {
       "enabled": false, "max_size_kb": -1, "max_objects": -1
     }
  }, "realm_id": "4a367026-bd8f-40ee-b486-8212482ddcd7", "realm_name": "gold",
 "realm_epoch": 1
}
```

注意

更新周期会更改版本号，并会确保其他区域可收到更新的配置。

21.13.2.7 启动网关

在对象网关主机上，启动并启用 Ceph 对象网关服务。要标识集群的唯一 FSID，请运行 **ceph fsid**。要标识对象网关守护进程名称，请运行 **ceph orch ps --hostname** HOSTNAME。

```
cephuser@ogw > systemctl start ceph-FSID@DAEMON_NAME
cephuser@ogw > systemctl enable ceph-FSID@DAEMON_NAME
```

21.13.3 配置次要区域

区域组内的区域会复制所有数据，以确保各区域都具有相同数据。创建次要区域时，请在指定为次要区域提供服务的主机上执行以下所有操作。

注意

要添加第三个区域，请执行与添加次要区域相同的过程。请使用不同的区域名称。

重要

必须在主区域内的主机上执行元数据操作（例如创建用户）。主区域及次要区域均可接收存储桶操作，但次要区域会将存储桶操作重定向到主区域。如果主区域处于停用状态，存储桶操作将会失败。

21.13.3.1 提取领域

使用主区域组中的主区域的 URL 路径、访问密钥和秘密密钥将领域配置提取到主机。要提取非默认领域，请使用 `--rgw-realm` 或 `--realm-id` 配置选项指定该领域。

```
cephuser@adm > radosgw-admin realm pull --url=url-to-master-zone-gateway --
access-key=access-key --secret=secret
```

注意

提取领域还会检索远程的当前周期配置，并将其设为此主机上的当前周期。

如果此领域为默认领域或唯一的领域，请将其设为默认领域。

```
cephuser@adm > radosgw-admin realm default --rgw-realm=REALM-NAME
```

21.13.3.2 创建次要区域

要为多站点配置创建次要区域，请在标识为用于次要区域的主机上打开命令行界面。为区域指定区域组 ID、新区域名称和端点。**请勿**使用 `--master` 标志。所有区域默认均以主动/主动配置运行。如果次要区域不应接受写入操作，请指定 `--read-only` 标志以在主区域和次要区域之间创建主动/被动配置。此外还需提供生成的系统用户的 `access_key` 和 `secret_key`，这些数据存储在主区域组的主区域中。执行以下命令：

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=ZONE-GROUP-NAME\
 --rgw-zone=ZONE-NAME --endpoints=URL \
 --access-key=SYSTEM-KEY --secret=SECRET\
 --endpoints=http://FQDN:80 \
 [--read-only]
```

例如：

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=us --endpoints=http://
rgw2:80 \
--rgw-zone=us-east-2 --access-key=SYSTEM_ACCESS_KEY --secret=SYSTEM_SECRET_KEY
{
  "id": "950c1a43-6836-41a2-a161-64777e07e8b8",
  "name": "us-east-2",
  "domain_root": "us-east-2.rgw.data.root",
  "control_pool": "us-east-2.rgw.control",
  "gc_pool": "us-east-2.rgw.gc",
  "log_pool": "us-east-2.rgw.log",
  "intent_log_pool": "us-east-2.rgw.intent-log",
  "usage_log_pool": "us-east-2.rgw.usage",
  "user_keys_pool": "us-east-2.rgw.users.keys",
```

```
  "user_email_pool": "us-east-2.rgw.users.email",
  "user_swift_pool": "us-east-2.rgw.users.swift",
  "user_uid_pool": "us-east-2.rgw.users.uid",
  "system_key": {
      "access_key": "1555b35654ad1656d804",
      "secret_key": "h7GhxuBLTrlhVUyxSPUKUV8r\/2EI4ngqJxD7iBdBYLhwluN30JaT3Q=="
  },
  "placement_pools": [
      {
          "key": "default-placement",
          "val": {
              "index_pool": "us-east-2.rgw.buckets.index",
              "data_pool": "us-east-2.rgw.buckets.data",
              "data_extra_pool": "us-east-2.rgw.buckets.non-ec",
              "index_type": 0
          }
      }
  ],
  "metadata_heap": "us-east-2.rgw.meta",
  "realm_id": "815d74c2-80d6-4e63-8cfc-232037f7ff5c"
}
```

重要

下面的步骤假设多站点配置使用的是新安装的系统，即其中尚未存储任何数据。如果您已在使用默认区域及其存储池存储数据，**切勿删除**该区域及其存储池，否则数据将会丢失且不可恢复。

根据需要删除默认区域：

```
cephuser@adm > radosgw-admin zone delete --rgw-zone=default
```

根据需要删除 Ceph 存储集群中的默认存储池：

```
cephuser@adm > ceph osd pool rm default.rgw.control default.rgw.control --yes-i-
really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.data.root default.rgw.data.root --
yes-i-really-really-mean-it
```

```
cephuser@adm > ceph osd pool rm default.rgw.gc default.rgw.gc --yes-i-really-
really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.log default.rgw.log --yes-i-really-
really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.users.uid default.rgw.users.uid --
yes-i-really-really-mean-it
```

21.13.3.3 更新 Ceph 配置文件

通过在相应实例条目中添加 `rgw_zone` 配置选项和次要区域名称，来更新次要区域主机上的 Ceph 配置文件。

为此，请执行以下命令：

```
cephuser@adm > ceph config set SERVICE_NAME rgw_zone us-west
```

21.13.3.4 更新周期

更新主区域配置后，请更新周期：

```
cephuser@adm > radosgw-admin period update --commit
{
  "id": "b5e4d3ec-2a62-4746-b479-4b2bc14b27d1",
  "epoch": 2,
  "predecessor_uuid": "09559832-67a4-4101-8b3f-10dfcd6b2707",
  "sync_status": [ "[...]"
  ],
  "period_map": {
      "id": "b5e4d3ec-2a62-4746-b479-4b2bc14b27d1",
      "zonegroups": [
          {
              "id": "d4018b8d-8c0d-4072-8919-608726fa369e",
              "name": "us",
              "api_name": "us",
              "is_master": "true",
              "endpoints": [
                  "http:\/\/rgw1:80"
```

```
        ],
        "hostnames": [],
        "hostnames_s3website": [],
        "master_zone": "83859a9a-9901-4f00-aa6d-285c777e10f0",
        "zones": [
            {
                "id": "83859a9a-9901-4f00-aa6d-285c777e10f0",
                "name": "us-east-1",
                "endpoints": [
                    "http:\/\/rgw1:80"
                ],
                "log_meta": "true",
                "log_data": "false",
                "bucket_index_max_shards": 0,
                "read_only": "false"
            },
            {
                "id": "950c1a43-6836-41a2-a161-64777e07e8b8",
                "name": "us-east-2",
                "endpoints": [
                    "http:\/\/rgw2:80"
                ],
                "log_meta": "false",
                "log_data": "true",
                "bucket_index_max_shards": 0,
                "read_only": "false"
            }

        ],
        "placement_targets": [
            {
                "name": "default-placement",
                "tags": []
            }
        ],
        "default_placement": "default-placement",
        "realm_id": "4a367026-bd8f-40ee-b486-8212482ddcd7"
    }
```

```
        ],
        "short_zone_ids": [
            {
                "key": "83859a9a-9901-4f00-aa6d-285c777e10f0",
                "val": 630926044
            },
            {
                "key": "950c1a43-6836-41a2-a161-64777e07e8b8",
                "val": 4276257543
            }

        ]
    },
    "master_zonegroup": "d4018b8d-8c0d-4072-8919-608726fa369e",
    "master_zone": "83859a9a-9901-4f00-aa6d-285c777e10f0",
    "period_config": {
        "bucket_quota": {
            "enabled": false,
            "max_size_kb": -1,
            "max_objects": -1
        },
        "user_quota": {
            "enabled": false,
            "max_size_kb": -1,
            "max_objects": -1
        }
    },
    "realm_id": "4a367026-bd8f-40ee-b486-8212482ddcd7",
    "realm_name": "gold",
    "realm_epoch": 2
}
```

注意

更新周期会更改版本号，并会确保其他区域可收到更新的配置。

21.13.3.5 启动对象网关

在对象网关主机上，启动并启用 Ceph 对象网关服务：

```
cephuser@adm > ceph orch start rgw.us-east-2
```

21.13.3.6 检查同步状态

在次要区域启用并运行时，检查同步状态。同步操作会将主区域中创建的用户和存储桶复制到次要区域。

```
cephuser@adm > radosgw-admin sync status
```

下面的输出提供了同步操作的状态。例如：

```
realm f3239bc5-e1a8-4206-a81d-e1576480804d (gold)
    zonegroup c50dbb7e-d9ce-47cc-a8bb-97d9b399d388 (us)
         zone 4c453b70-4a16-4ce8-8185-1893b05d346e (us-west)
metadata sync syncing
              full sync: 0/64 shards
              metadata is caught up with master
              incremental sync: 64/64 shards
    data sync source: 1ee9da3e-114d-4ae3-a8a4-056e8a17f532 (us-east)
                      syncing
                      full sync: 0/128 shards
                      incremental sync: 128/128 shards
                      data is caught up with source
```

注意

次要区域会接受存储桶操作，但会将它们重定向到主区域，然后会与主区域同步以接收存储桶操作的结果。如果主区域处于停用状态，对次要区域执行的存储桶操作将会失败，但对象操作应该会成功完成。

21.13.3.7 对象的验证

默认情况下，当对象的同步成功完成后，不会再次验证对象。要启用验证，请将 `rgw_sync_obj_etag_verify` 选项设置为 `true`。启用此选项后，将会同步可选对象。一个额外的 MD5 校验和会验证是否在源和目标上计算了该数据。这是为了确保通过 HTTP（包括多站点同步）从远程服务器提取的对象的完整性。当需要更多计算资源时，此选项可能会降低 RGW 的性能。

21.13.4 常规的对象网关维护

21.13.4.1 检查同步状态

可以使用以下命令查询有关区域复制状态的信息：

```
cephuser@adm > radosgw-admin sync status
        realm b3bc1c37-9c44-4b89-a03b-04c269bea5da (gold)
    zonegroup f54f9b22-b4b6-4a0e-9211-fa6ac1693f49 (us)
         zone adce11c9-b8ed-4a90-8bc5-3fc029ff0816 (us-west)
        metadata sync syncing
              full sync: 0/64 shards
              incremental sync: 64/64 shards
              metadata is behind on 1 shards
              oldest incremental change not applied: 2017-03-22
 10:20:00.0.881361s
data sync source: 341c2d81-4574-4d08-ab0f-5a2a7b168028 (us-east)
                  syncing
                  full sync: 0/128 shards
                  incremental sync: 128/128 shards
                  data is caught up with source
          source: 3b5d1a3f-3f27-4e4a-8f34-6072d4bb1275 (us-3)
                  syncing
                  full sync: 0/128 shards
                  incremental sync: 128/128 shards
                  data is caught up with source
```

输出可能会因同步状态的不同而异。同步期间将分片分为两种不同的类型：

滞后分片

滞后分片是指需要进行完整数据同步的分片以及由于数据不是最新的而需进行增量数据同步的分片。

恢复分片

恢复分片是指在同步期间遇到错误并被标记为需重试的分片。错误大多数会在出现小问题（例如捕获到存储桶锁定）时发生。这类错误通常会自行解决。

21.13.4.2 检查日志

仅对于多站点，您可以检查元数据日志 (`mdlog`)、存储桶索引日志 (`bilog`) 和数据日志 (`datalog`)。您可以列出这些日志并对其进行剪裁。大多数情况下不需要如此，因为 `rgw_sync_log_trim_interval` 选项默认设置为 20 分钟。如果此选项未手动设置为 0，就无需对其进行剪裁，否则可能会产生副作用。

21.13.4.3 更改元数据主区域

重要

更改元数据的主区域时要小心。如果某个区域未完成与当前主区域的元数据同步，将其升级为主区域时，它将无法处理剩余的条目，这些更改将会丢失。因此，我们建议等到区域的 **`radosgw-admin`** 同步状态跟上元数据同步后，再将该区域升级为主区域。同样，如果当前主区域正在处理对元数据的更改，此时将另一个区域升级为主区域，这些更改也可能会丢失。为避免此情况，我们建议关闭之前的主区域上的所有对象网关实例。升级另一个区域后，可以使用 **`radosgw-admin`** 周期提取命令获取它的新周期，然后可以重启动网关。

要将一个区域（例如，`us` 区域组中的 `us-west` 区域）升级为元数据的主区域，请在该区域运行以下命令：

```
cephuser@ogw > radosgw-admin zone modify --rgw-zone=us-west --master
cephuser@ogw > radosgw-admin zonegroup modify --rgw-zonegroup=us --master
cephuser@ogw > radosgw-admin period update --commit
```

这样会生成一个新周期，并且 `us-west` 区域中的对象网关实例会将此周期发送到其他区域。

21.13.5 执行故障转移和灾难恢复

如果主区域发生故障，将故障转移到次要区域，以实现灾难恢复。

1. 将次要区域设为主区域和默认区域。例如：

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --master --
default
```

   默认情况下，Ceph 对象网关以主动/主动配置运行。如果已将集群配置为以主动/被动配置运行，则次要区域是只读区域。删除 `--read-only` 状态可允许区域接收写入操作。例如：

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --master --
default \
                                                --read-only=false
```

2. 更新周期以使更改生效：

```
cephuser@adm > radosgw-admin period update --commit
```

3. 重启动 Ceph 对象网关：

```
cephuser@adm > ceph orch restart rgw
```

如果之前的主区域已恢复，请逆向操作。

1. 在已恢复的区域中，从当前主区域提取最新的领域配置。

```
cephuser@adm > radosgw-admin realm pull --url=URL-TO-MASTER-ZONE-GATEWAY \
                            --access-key=ACCESS-KEY --secret=SECRET
```

2. 将已恢复的区域设为主区域和默认区域：

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --master --
default
```

3. 更新周期以使更改生效：

```
cephuser@adm > radosgw-admin period update --commit
```

4. 在已恢复的区域中重启动 Ceph 对象网关：

```
cephuser@adm > ceph orch restart rgw@rgw
```

5. 如果次要区域需要采用只读配置，请更新次要区域：

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --read-only
```

6. 更新周期以使更改生效：

```
cephuser@adm > radosgw-admin period update --commit
```

7. 在次要区域中重启动 Ceph 对象网关：

```
cephuser@adm > ceph orch restart@rgw
```

22 Ceph iSCSI 网关

本章重点介绍与 iSCSI 网关相关的管理任务。有关部署过程，请参见《部署指南》, 第 8 章 “使用 cephadm 部署其余核心服务”, 第 8.3.5 节 “部署 iSCSI 网关”。

22.1 ceph-iscsi 管理的目标

本节介绍如何从运行 Linux、Microsoft Windows 或 VMware 的客户端连接到 `ceph-iscsi` 管理的目标。

22.1.1 连接到 open-iscsi

使用 `ceph-iscsi` 连接 `open-iscsi` 支持的 iSCSI 目标需要执行两个步骤。首先，发起程序必须发现网关主机上可用的 iSCSI 目标，然后，必须登录并映射可用的逻辑单元 (LU)。

这两个步骤都需要 `open-iscsi` 守护进程处于运行状态。启动 `open-iscsi` 守护进程的方式取决于您的 Linux 发行套件：

- 在 SUSE Linux Enterprise Server (SLES) 和 Red Hat Enterprise Linux (RHEL) 主机上，运行 **`systemctl start iscsid`**（如果 **`systemctl`** 不可用，请运行 **`service iscsid start`**）。
- 在 Debian 和 Ubuntu 主机上，运行 **`systemctl start open-iscsi`**（或 **`service open-iscsi start`**）。

如果发起程序主机运行 SUSE Linux Enterprise Server，请参见 https://documentation.suse.com/sles/15-SP1/single-html/SLES-storage/#sec-iscsi-initiator 了解有关如何连接 iSCSI 目标的详细信息。

对于支持 `open-iscsi` 的任何其他 Linux 发行套件，请继续发现 `ceph-iscsi` 网关上的目标（本示例使用 iscsi1.example.com 作为端口地址；对于多路径访问，请使用 iscsi2.example.com 重复这些步骤）：

```
# iscsiadm -m discovery -t sendtargets -p iscsi1.example.com
```

```
192.168.124.104:3260,1 iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol
```

然后登录该端口。如果登录成功完成，则端口中所有基于 RBD 的逻辑单元将立即在系统 SCSI 总线上变为可用：

```
# iscsiadm -m node -p iscsi1.example.com --login
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol, portal: 192.168.124.104,3260] (multiple)
Login to [iface: default, target: iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol, portal: 192.168.124.104,3260] successful.
```

针对其他端口 IP 地址或主机重复此过程。

如果系统上已安装 `lsscsi` 实用程序，您可以使用它来枚举系统上可用的 SCSI 设备：

```
lsscsi
[8:0:0:0]    disk    SUSE     RBD              4.0   /dev/sde
[9:0:0:0]    disk    SUSE     RBD              4.0   /dev/sdf
```

在多路径配置（其中两个已连接的 iSCSI 设备代表一个相同的 LU）中，您还可以使用 `multipath` 实用程序检查多路径设备状态：

```
# multipath -ll
360014050cf9dcfcb2603933ac3298dca dm-9 SUSE,RBD
size=49G features='0' hwhandler='0' wp=rw
|-+- policy='service-time 0' prio=1 status=active
| `- 8:0:0:0 sde 8:64 active ready running
`-+- policy='service-time 0' prio=1 status=enabled
`- 9:0:0:0 sdf 8:80 active ready running
```

现在，可根据需要，将此多路径设备用作任何块设备。例如，可将该设备用作 Linux 逻辑卷管理 (LVM) 的物理卷，或者直接在该设备上创建文件系统。下面的示例说明如何在新连接的多路径 iSCSI 卷上创建 XFS 文件系统：

```
# mkfs -t xfs /dev/mapper/360014050cf9dcfcb2603933ac3298dca
log stripe unit (4194304 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/mapper/360014050cf9dcfcb2603933ac3298dca isize=256    agcount=17,
 agsize=799744 blks
```

```
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=12800000, imaxpct=25
         =                       sunit=1024   swidth=1024 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal log           bsize=4096   blocks=6256, version=2
         =                       sectsz=512   sunit=8 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

请注意，由于 XFS 是非集群文件系统，无论何时，您都只能将它挂载到单个 iSCSI 发起程序节点上。

任何时候如果要停止使用与特定目标关联的 iSCSI LU，请运行以下命令：

```
# iscsiadm -m node -p iscsi1.example.com --logout
Logging out of session [sid: 18, iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol, portal: 192.168.124.104,3260]
Logout of [sid: 18, target: iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol, portal: 192.168.124.104,3260] successful.
```

与执行发现和登录时一样，必须针对所有端口 IP 地址或主机名重复注销步骤。

22.1.1.1 配置多路径

多路径配置保留在客户端或发起程序上，并不依赖于任何 `ceph-iscsi` 配置。在使用块存储之前，请选择一个策略。编辑 `/etc/multipath.conf` 之后，请使用以下命令重启动 `multipathd`

```
# systemctl restart multipathd
```

对于包含友好名称的主动/被动配置，请将

```
defaults {
  user_friendly_names yes
}
```

添加到 `/etc/multipath.conf`。成功连接目标后，请运行

```
# multipath -ll
```

```
mpathd (36001405dbb561b2b5e439f0aed2f8e1e) dm-0 SUSE,RBD
size=2.0G features='0' hwhandler='0' wp=rw
|-+- policy='service-time 0' prio=1 status=active
| `- 2:0:0:3 sdl 8:176 active ready running
|-+- policy='service-time 0' prio=1 status=enabled
| `- 3:0:0:3 sdj 8:144 active ready running
`-+- policy='service-time 0' prio=1 status=enabled
  `- 4:0:0:3 sdk 8:160 active ready running
```

注意每个链路的状态。对于主动/主动配置，请将

```
defaults {
  user_friendly_names yes
}

devices {
  device {
    vendor "(LIO-ORG|SUSE)"
    product "RBD"
    path_grouping_policy "multibus"
    path_checker "tur"
    features "0"
    hardware_handler "1 alua"
    prio "alua"
    failback "immediate"
    rr_weight "uniform"
    no_path_retry 12
    rr_min_io 100
  }
}
```

添加到 `/etc/multipath.conf`。重启动 `multipathd` 并运行

```
# multipath -ll
mpathd (36001405dbb561b2b5e439f0aed2f8e1e) dm-3 SUSE,RBD
size=2.0G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
`-+- policy='service-time 0' prio=50 status=active
  |- 4:0:0:3 sdj 8:144 active ready running
  |- 3:0:0:3 sdk 8:160 active ready running
```

```
`- 2:0:0:3 sdl 8:176 active ready running
```

22.1.2 连接 Microsoft Windows（Microsoft iSCSI 发起程序）

要从 Windows 2012 服务器连接 SUSE Enterprise Storage iSCSI 目标，请执行以下步骤：

1. 打开 Windows 服务器管理器。在仪表盘中，选择工具 › iSCSI 发起程序。iSCSI 发起程序属性对话框随即显示。选择发现选项卡：

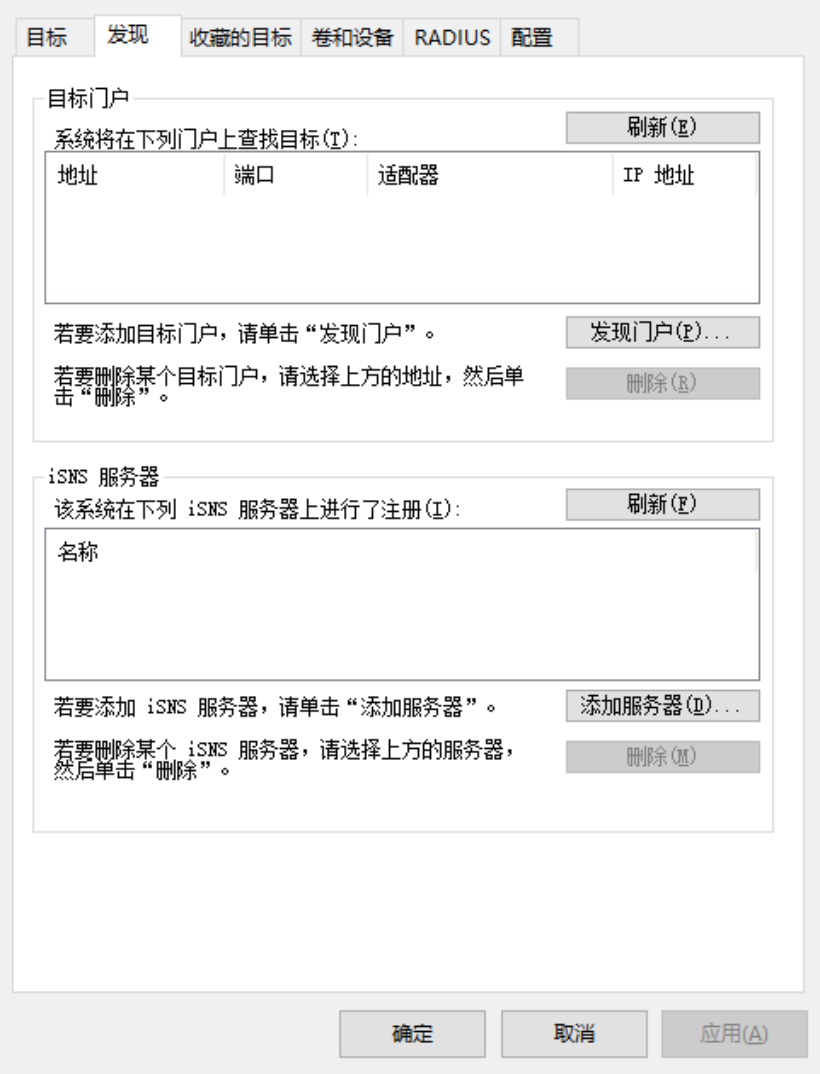


图 22.1： ISCSI 发起程序属性

2. 在发现目标门户对话框中的目标字段内，输入目标的主机名或 IP 地址，然后点击确定：

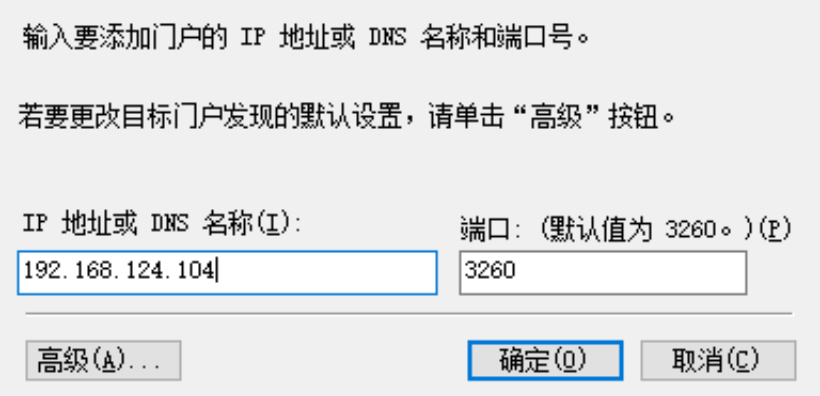


图 22.2： 发现目标门户

3. 针对所有其他网关主机名或 IP 地址重复此过程。完成后，查看目标门户列表：

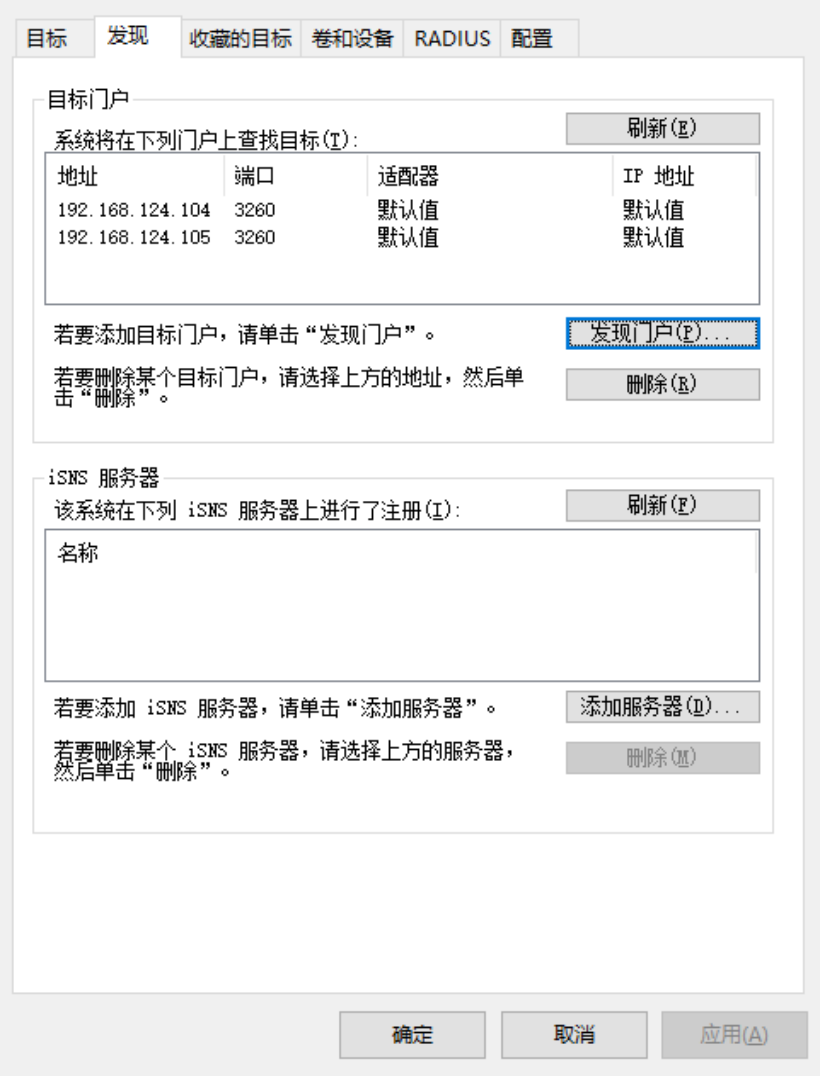


图 22.3： 目标门户

4. 接下来，切换到目标选项卡并查看已发现的目标。

图 22.4： 目标

5. 在目标选项卡中点击连接。连接目标对话框随机显示。选中启用多路径复选框以启用多路径 I/O (MPIO)，然后点击确定：

6. 连接目标对话框关闭后，选择属性查看目标的属性：

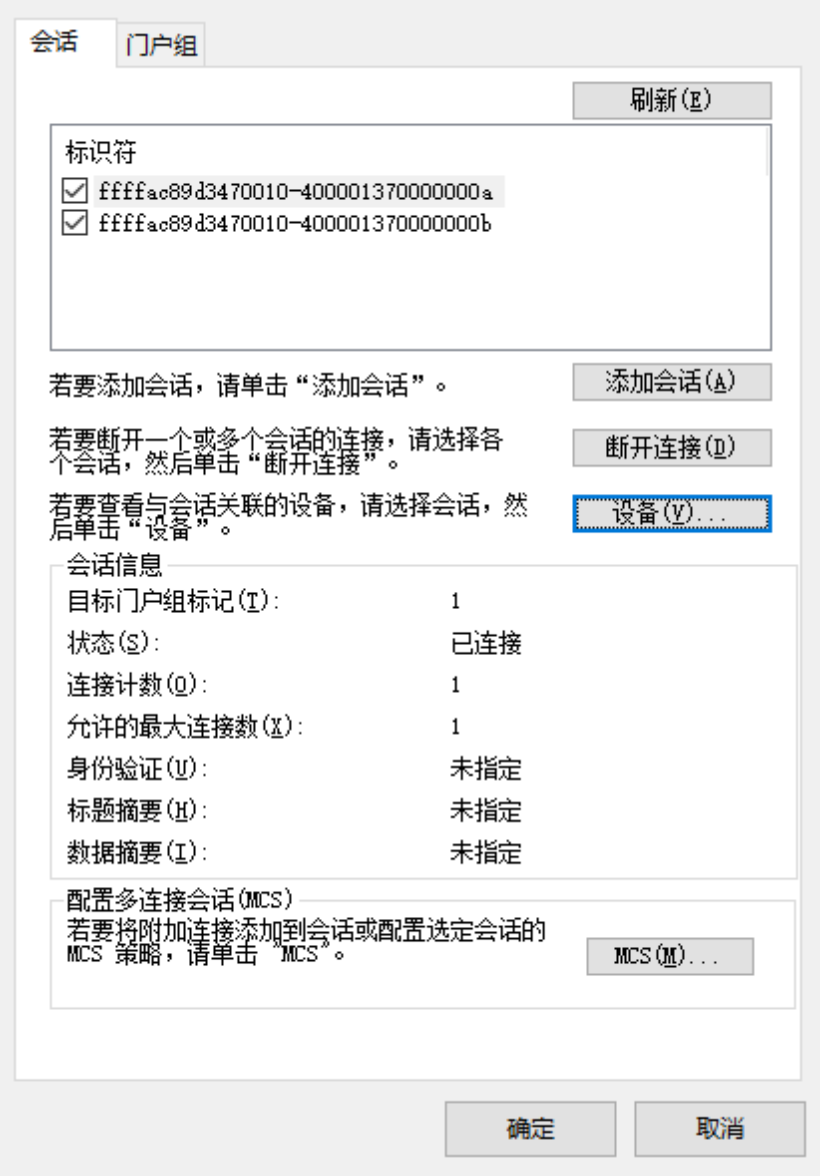


图 22.5： ISCSI 目标属性

7. 选择设备，然后点击 MPIO 查看多路径 I/O 配置：

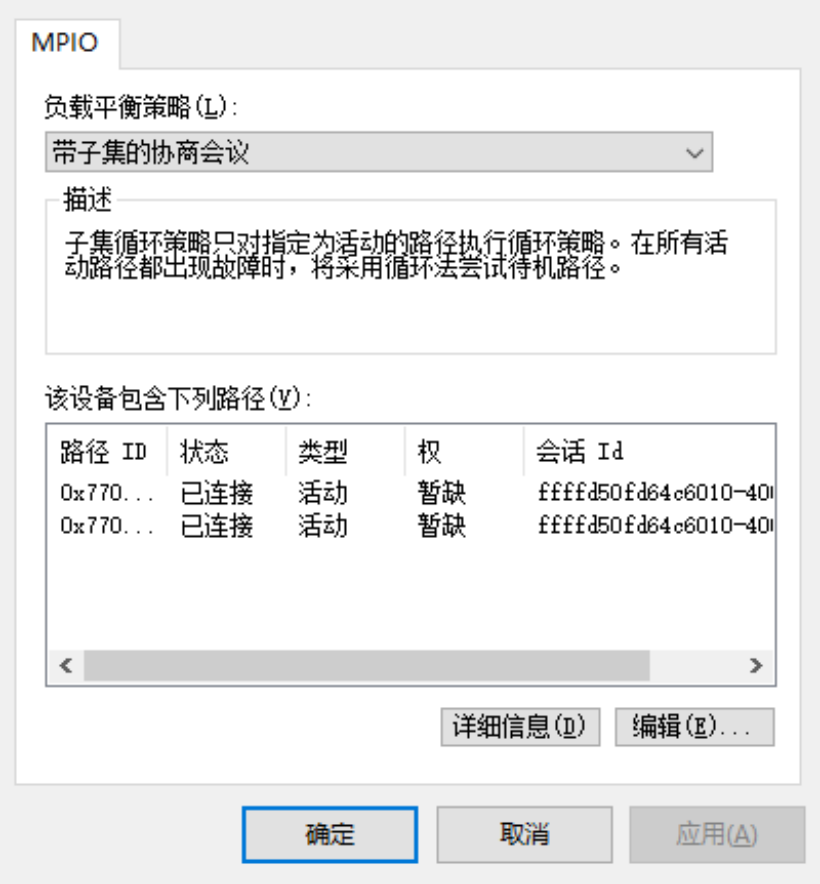


图 22.6： 设备详细信息

默认的负载平衡策略为带子集的协商会议。如果您偏向于单纯的故障转移配置，请将策略更改为仅故障转移。

iSCSI 发起程序的配置到此结束。现在，可以像使用任何其他 SCSI 设备一样使用 iSCSI 卷，并可将其初始化，使其可用作卷和驱动器。点击确定关闭 iSCSI 发起程序属性对话框，然后继续在服务器管理器仪表板中配置文件和储存服务角色。

观察新连接的卷。该卷标识为 iSCSI 总线上的 **SUSE RBD SCSI 多路径驱动器**，并且最初标记为**脱机**状态，其分区表类型为**未知**。如果新卷未立即显示，请从任务下拉框中选择重新扫描存储，以重新扫描 iSCSI 总线。

1. 右键点击 iSCSI 卷，然后从上下文菜单中选择新建卷。新建卷向导随即显示。点击下一步，突出显示新连接的 iSCSI 卷，然后点击下一步开始创建新卷。

图 22.7： **新建卷向导**

2. 该设备最初是空的，不包含任何分区表。当出现对话框指出将要使用 GPT 分区表初始化卷时，确认该操作：

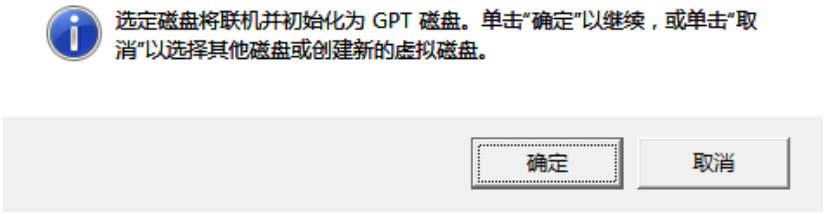


图 22.8： **脱机磁盘提示**

3. 选择卷大小。通常，用户会使用设备的全部容量。然后，指定新建卷将在其上变为可用状态的驱动器盘符或目录名称。接下来，选择要在新卷上创建的文件系统。最后，点击创建确认所做的选择并完成卷的创建：

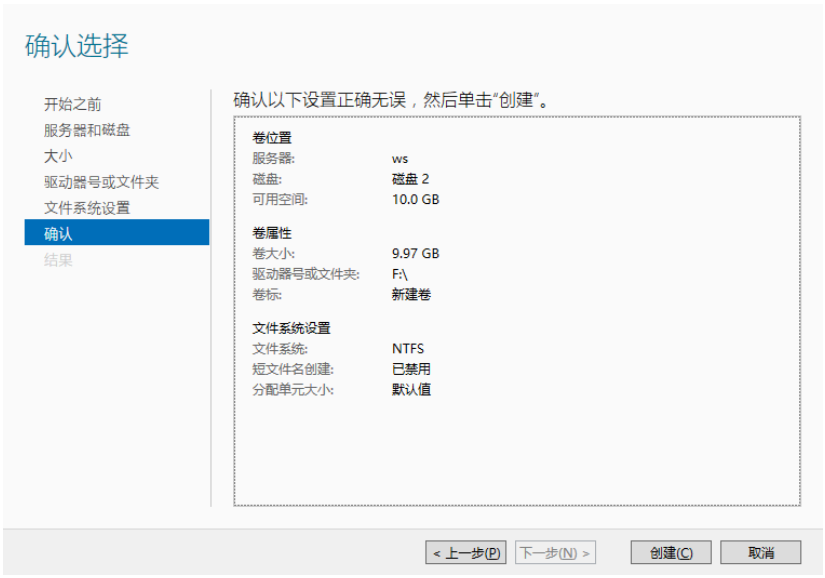


图 22.9： 确认选择的卷设置

完成该过程后，请检查结果，然后点击关闭结束驱动器初始化。完成初始化后，便可以像使用新初始化的本地驱动器一样使用该卷（及其 NTFS 文件系统）。

22.1.3 连接 VMware

1. 要连接到 `ceph-iscsi` 管理的 iSCSI 卷，需要有配置好的 iSCSI 软件适配器。如果 vSphere 配置中未提供此类适配器，请选择配置 › 储存适配器 › 添加 › iSCSI 软件发起程序来创建一个适配器。

2. 如果适用，请通过右键点击该适配器并从上下文菜单中选择属性，来选择该适配器的属性：

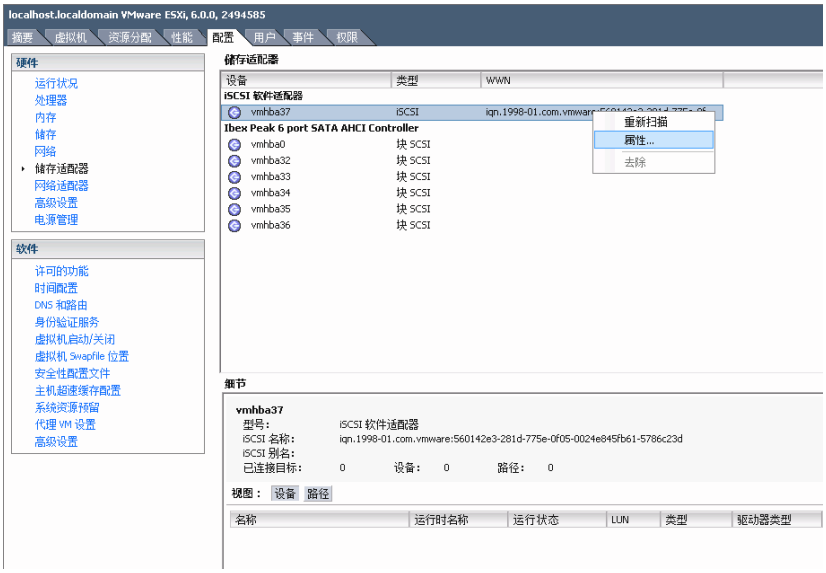


图 22.10： ISCSI 发起程序属性

3. 在 iSCSI 软件发起程序对话框中，点击配置按钮。然后转到动态发现选项卡并选择添加。

4. 输入 `ceph-iscsi` iSCSI 网关的 IP 地址或主机名。如果在故障转移配置中运行多个 iSCSI 网关，请针对要运行的所有网关重复此步骤。

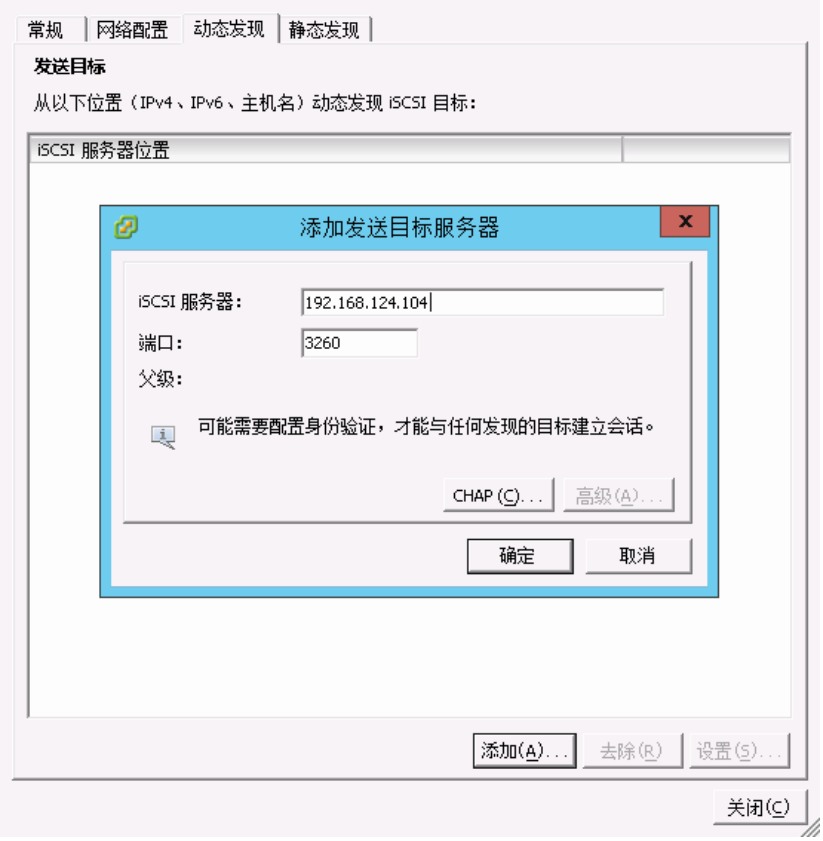


图 22.11： 添加目标服务器

输入所有 iSCSI 网关后，请在对话框中点击确定，启动 iSCSI 适配器的重新扫描。

5. 重新扫描完成后，新的 iSCSI 设备会显示在细节窗格中的储存适配器列表下。对于多路径设备，现在可以右键点击该适配器，然后从上下文菜单中选择管理路径：

图 22.12： 管理多路径设备

您现在应该会看到，所有路径的状态下面都带有绿灯。其中一个路径应已标记为主动 (I/O)，其他所有路径只是标记为主动：

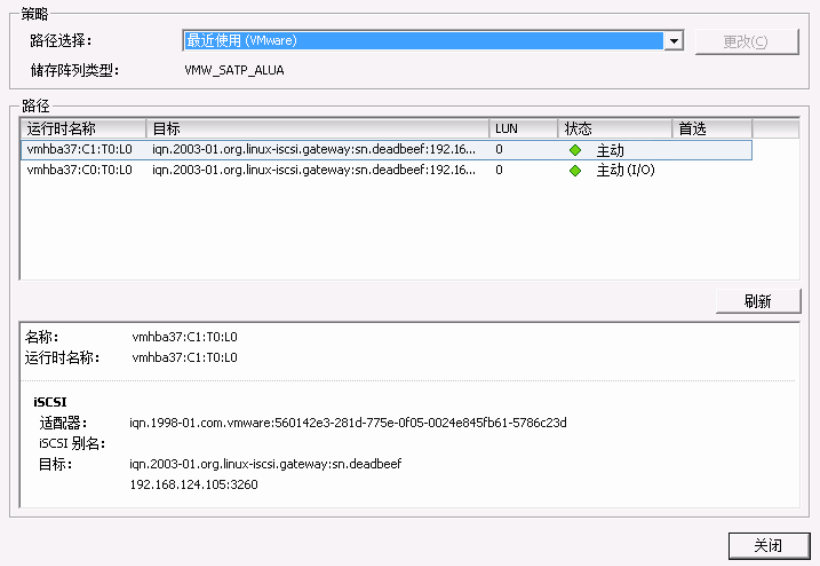


图 22.13： 多路径的路径列表

6. 现在，可以从储存适配器切换到标有储存的项目。在窗格右上角选择添加储存... 打开添加储存对话框。然后选择磁盘/LUN 并点击下一步。新添加的 iSCSI 设备会显示在选择磁盘/LUN 列表中。选择该设备，然后点击下一步继续：

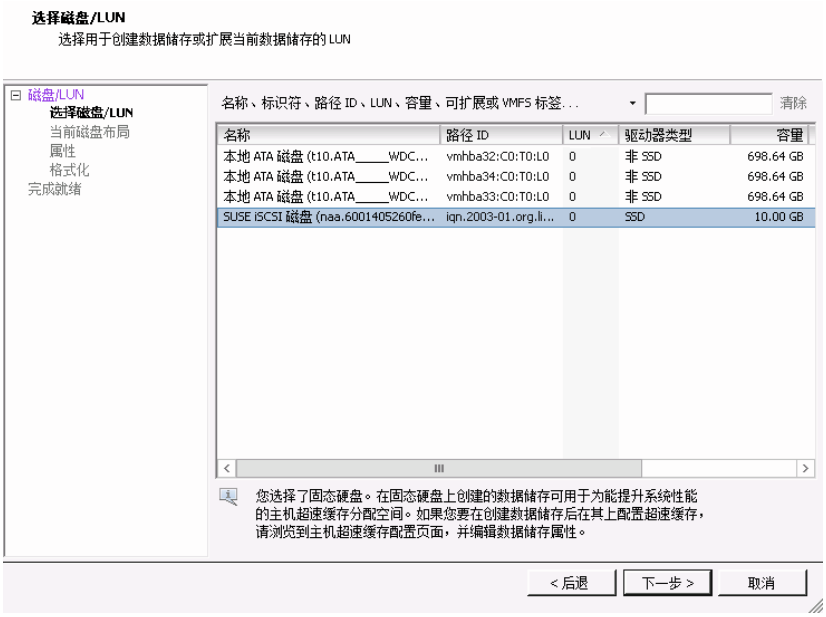


图 22.14： “添加储存”对话框

点击下一步接受默认的磁盘布局。

7. 在属性窗格中，为新数据储存指派名称，然后点击下一步。接受将卷的整个空间用于数据储存的默认设置，或者选择自定义空间设置以创建较小的数据储存：

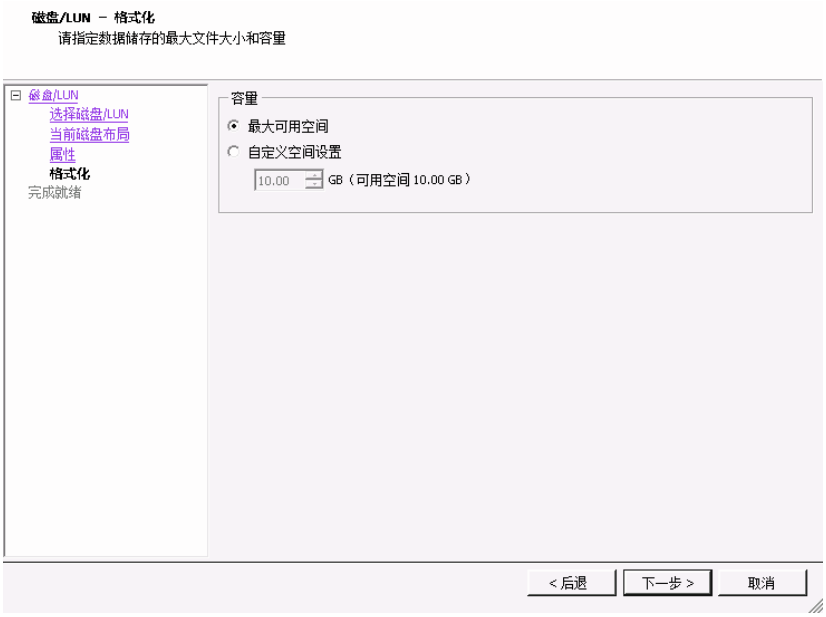


图 22.15： 自定义空间设置

点击完成以完成数据储存的创建。

新数据存储现在即会显示在数据存储列表中，您可以选择它来检索详细信息。现在，您可以像使用任何其他 vSphere 数据存储一样使用 `ceph-iscsi` 支持的 iSCSI 卷。

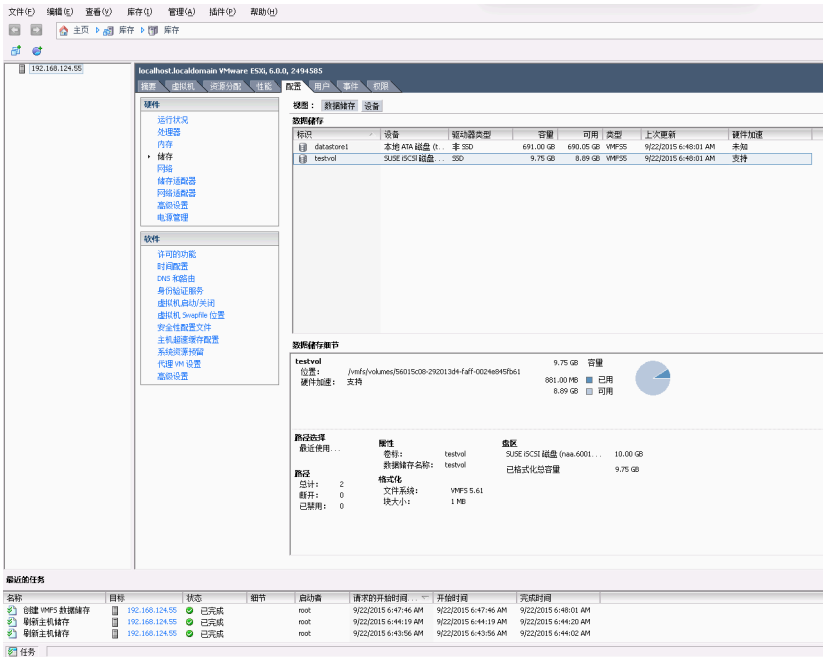


图 22.16： ISCSI 数据存储概述

22.2 结论

ceph-iscsi 是 SUSE Enterprise Storage 7.1 的一个关键组件，使用它可以从任何支持 iSCSI 协议的服务器或客户端访问高度可用的分布式块存储。在一个或多个 iSCSI 网关主机上使用 ceph-iscsi 可将 Ceph RBD 映像用作与 iSCSI 目标关联的逻辑单元 (LU)，并可根据需要以负载平衡且高度可用的方式来访问该逻辑单元。

由于所有 ceph-iscsi 配置都存储在 Ceph RADOS 对象存储中，ceph-iscsi 网关主机先天就不具有持久性状态，因而可以任意对其进行更换或者增减。因此，借助 SUSE Enterprise Storage 7.1，SUSE 客户可在市售硬件和完全开源的平台上运行真正的分布式、高度可用、有弹性且可自我修复的企业存储技术。

23 集群化文件系统

本章介绍在设置集群并导出 CephFS 后通常应执行的管理任务。如需有关设置 CephFS 的详细信息，请参见《部署指南》, 第 8 章 “使用 cephadm 部署其余核心服务”, 第 8.3.3 节 “部署元数据服务器”。

23.1 挂载 CephFS

当创建文件系统并且 MDS 处于工作状态后，您便可以从客户端主机挂载文件系统。

23.1.1 准备客户端

如果客户端主机运行的是 SUSE Linux Enterprise 12 SP2 或更高版本，系统无需额外配置即可挂载 CephFS。

如果客户端主机运行的是 SUSE Linux Enterprise 12 SP1，您需要应用所有最新的补丁，之后才能挂载 CephFS。

无论是哪一种情况，SUSE Linux Enterprise 中都包含了挂载 CephFS 需要的所有项目。不需要 SUSE Enterprise Storage 7.1 产品。

要支持完整的 **mount** 语法，在尝试挂载 CephFS 之前，应该先安装 `ceph-common` 软件包（随附于 SUSE Linux Enterprise 中）。

重要

如果不安装 `ceph-common` 软件包（因而也就未安装 **mount.ceph** 助手），将需要使用 Monitor 的 IP 而非其名称。这是因为内核客户端将无法执行名称解析。

基本挂载语法如下：

```
# mount -t ceph MON1_IP[:PORT],MON2_IP[:PORT],...:CEPHFS_MOUNT_TARGET \
MOUNT_POINT -o name=CEPHX_USER_NAME,secret=SECRET_STRING
```

23.1.2 创建秘密密钥文件

Ceph 集群默认是在启用身份验证的情况下运行的。应该创建一个文件用于存储您的秘密密钥（而不是密钥环本身）。要获取特定用户的秘密密钥，然后创建该文件，请执行以下操作：

过程 23.1： 创建秘密密钥

1. 在密钥环文件中查看特定用户的密钥：

```
cephuser@adm > cat /etc/ceph/ceph.client.admin.keyring
```

2. 复制要使用所挂载 Ceph FS 文件系统的用户的密钥。密钥通常类似下方所示：

```
AQCj2YpRiAe6CxAA7/ETt7Hcl9IyxyYciVs47w==
```

3. 为用户 **admin** 创建一个文件名包含用户名的文件，例如 `/etc/ceph/admin.secret`。
4. 将密钥值粘贴到上一步中创建的文件。
5. 设置对该文件的适当访问权限。该用户应是唯一有权读取该文件的用户，其他人不能有任何访问权限。

23.1.3 挂载 CephFS

可以使用 **`mount`** 命令挂载 CephFS。需要指定 Monitor 的主机名或 IP 地址。由于 SUSE Enterprise Storage 中默认会启用 `cephx` 身份验证，因此，您还需要指定一个用户名及其相关的机密：

```
# mount -t ceph ceph_mon1:6789:/ /mnt/cephfs \
 -o name=admin,secret=AQATSKdNGBnwLhAAnNDKnH65FmVKpXZJVasUeQ==
```

由于上一条命令会保留在外壳历史中，因此更安全的做法是从文件读取机密：

```
# mount -t ceph ceph_mon1:6789:/ /mnt/cephfs \
 -o name=admin,secretfile=/etc/ceph/admin.secret
```

请注意，机密文件应该只包含实际的密钥环机密。因此，在本示例中，该文件只包含下行：

```
AQATSKdNGBnwLhAAnNDKnH65FmVKpXZJVasUeQ==
```

提示：指定多个 Monitor

最好是在 **mount** 命令行中指定多个 Monitor 并以逗号分隔，以防在挂载时某个 Monitor 恰好停机。每个 Monitor 的地址采用主机[:端口]格式。如果未指定端口，默认会使用端口 6789。

在本地主机上创建挂载点：

```
# mkdir /mnt/cephfs
```

挂载 CephFS：

```
# mount -t ceph ceph_mon1:6789:/ /mnt/cephfs \
 -o name=admin,secretfile=/etc/ceph/admin.secret
```

如果要挂载文件系统的某个子集，可以指定子目录 `subdir`：

```
# mount -t ceph ceph_mon1:6789:/subdir /mnt/cephfs \
 -o name=admin,secretfile=/etc/ceph/admin.secret
```

可在 **mount** 命令中指定多个 Monitor 主机：

```
# mount -t ceph ceph_mon1,ceph_mon2,ceph_mon3:6789:/ /mnt/cephfs \
 -o name=admin,secretfile=/etc/ceph/admin.secret
```

重要：对根目录的读取访问权限

如果使用了实施路径限制的客户端，则 MDS 功能需要包含对根目录的读取访问权限。例如，密钥环可能如下所示：

```
client.bar
 key: supersecretkey
 caps: [mds] allow rw path=/barjail, allow r path=/
 caps: [mon] allow r
 caps: [osd] allow rwx
```

`allow r path=/` 部分表示路径受限的客户端能够查看根卷，但无法写入根卷。在要求完全隔离的用例中，这可能会造成问题。

23.2 卸载 CephFS

要卸载 CephFS，请使用 **`umount`** 命令：

```
# umount /mnt/cephfs
```

23.3 在 `/etc/fstab` 中挂载 CephFS

要在客户端启动时自动挂载 CephFS，请在其文件系统表 `/etc/fstab` 中插入相应的行：

```
mon1:6790,mon2:/subdir /mnt/cephfs ceph name=admin,secretfile=/etc/ceph/
secret.key,noatime,_netdev 0 2
```

23.4 多个活动 MDS 守护进程（主动/主动 MDS）

默认情况下，CephFS 是针对单个工作的 MDS 守护进程配置的。要调整大规模系统的元数据性能，可以启用多个工作的 MDS 守护进程，以便互相分担元数据工作负载。

23.4.1 使用主动/主动 MDS

如果按默认设置使用单个 MDS 时元数据性能出现瓶颈，可考虑使用多个工作的 MDS 守护进程。

增加守护进程并不会提高所有工作负载类型的性能。例如，增加 MDS 守护进程的数量不会让单个客户端上运行的单个应用受益，除非该应用在同时执行大量元数据操作。

通常能够因大量工作的 MDS 守护进程受益的工作负载是使用许多客户端的工作负载，也许是在许多独立目录中工作的工作负载。

23.4.2 增加 MDS 活动集群的大小

每个 CephFS 文件系统都有一项 `max_mds` 设置，用于控制将要创建的级别数。仅当某个备用守护进程可供新的级别使用时，文件系统中的实际级别数才会增加。例如，如果只有一个 MDS 守护进程在运行，并且 `max_mds` 设置为 2，将不会创建另一个级别。

在下面的示例中，我们将 `max_mds` 选项设置为 2，以便在保留默认级别的情况下再创建一个新级别。要查看更改，请在设置 `max_mds` 之前和之后运行 **`ceph status`**，然后观察包含 `fsmap` 的行：

```
cephuser@adm > ceph status
  [...]
  services:
    [...]
    mds: cephfs-1/1/1 up  {0=node2=up:active}, 1 up:standby
    [...]
cephuser@adm > ceph fs set cephfs max_mds 2
cephuser@adm > ceph status
  [...]
  services:
    [...]
    mds: cephfs-2/2/2 up  {0=node2=up:active,1=node1=up:active}
    [...]
```

新建的级别 (1) 会经历“正在创建”状态，然后进入“工作”状态。

重要：待机守护进程

即使使用多个工作 MDS 守护进程，当任何在运行工作守护进程的服务器发生故障时，高可用性系统也仍会要求待机守护进程接管工作。

因此，高可用性系统的 `max_mds` 合理最大值比系统中的 MDS 服务器总数小 1。要在发生多次服务器故障时保持可用性，可增加系统中待机守护进程的数量，使之与不会导致失去可用性的服务器故障数一致。

23.4.3 减小级别数量

所有级别（包括要删除的级别）首先必须是活跃的。这意味着，至少需要有 `max_mds` 个 MDS 守护进程可用。

首先，将 `max_mds` 设为一个较小的数字。例如，我们重新使用单个工作的 MDS：

```
cephuser@adm > ceph status
  [...]
  services:
    [...]
    mds: cephfs-2/2/2 up  {0=node2=up:active,1=node1=up:active}
    [...]
cephuser@adm > ceph fs set cephfs max_mds 1
cephuser@adm > ceph status
  [...]
  services:
    [...]
    mds: cephfs-1/1/1 up  {0=node2=up:active}, 1 up:standby
    [...]
```

23.4.4 手动将目录树关联到级别

在多个工作元数据服务器配置中，将会运行一个平衡器，用于在集群中均衡分配元数据负载。这种模式通常足以满足大多数用户的需求，但有时，用户需要使用元数据到特定级别的显式映射来覆盖动态平衡器。这样，管理员或用户便可以在整个集群上均衡地分配应用负载，或限制用户的元数据请求对整个集群的影响。

针对此目的提供的机制称为“导出关联”。它是目录的扩展属性。此扩展属性名为 `ceph.dir.pin`。用户可以使用标准命令设置此属性：

```
# setfattr -n ceph.dir.pin -v 2 /path/to/dir
```

扩展属性的值 (`-v`) 是要将目录子树指定到的级别。默认值 -1 表示不关联该目录。

目录导出关联继承自设置了导出关联的最近的父级。因此，对某个目录设置导出关联会影响该目录的所有子级。但是，可以通过设置子目录导出关联来覆盖父级的关联。例如：

```
# mkdir -p a/b                       # "a" and "a/b" start with no export pin
 set.
setfattr -n ceph.dir.pin -v 1 a/  # "a" and "b" are now pinned to rank 1.
setfattr -n ceph.dir.pin -v 0 a/b # "a/b" is now pinned to rank 0
                                  # and "a/" and the rest of its children
                                  # are still pinned to rank 1.
```

23.5 管理故障转移

如果 MDS 守护进程停止与 Monitor 通讯，Monitor 会等待 `mds_beacon_grace` 秒（默认为 15 秒），然后将守护进程标记为 **laggy**。可以配置一个或多个“待机”守护进程，用于在 MDS 守护进程故障转移期间接管工作。

23.5.1 配置待机重放

可以配置每个 CephFS 文件系统以添加待机重放守护进程。这些待机守护进程会跟随活动 MDS 的元数据日志，以缩短当活动 MDS 变为不可用状态时的故障转移时间。每个活动 MDS 只能有一个跟随它的待机重放守护进程。

可以使用以下命令在文件系统上配置待机重放：

```
cephuser@adm > ceph fs set FS-NAME allow_standby_replay BOOL
```

设置待机重放后，Monitor 将会指定该文件系统中可用于跟随活动 MDS 的待机守护进程。

当某个 MDS 进入待机重放状态后，它只会用作所跟随级别的待机守护进程。如果另一个级别发生故障，此待机重放守护进程不会作为替代者，即使没有其他待机守护进程可用也是如此。因此，如果使用了待机重放，则建议每个活动 MDS 都应有一个待机重放守护进程。

23.6 设置 CephFS 配额

您可以对 Ceph 文件系统的任何子目录设置配额。配额可限制目录层次结构中指定点下所存储的**字节**或**文件**数。

23.6.1 CephFS 配额限制

CephFS 的配额设置具有以下限制：

配额是合作性而非竞争性的。

达到限制时，Ceph 配额依赖挂载文件系统的客户端来停止向其写入数据。服务器端无法阻止恶意客户端写入所需数量的数据。在客户端完全不受信任的环境中，请勿使用配额来阻止填充文件系统。

配额并不精确。

在达到配额限制不久后，系统便会停止向文件系统写入数据的进程。这样便会不可避免地允许这些进程在超出配置的限制后又写入一定数量的数据。系统将会在超出所配置限制后的十分之一秒内停止客户端写入程序。

配额是在自 4.17 版本起的内核客户端中实施的。

配额受用户空间客户端（libcephfs、ceph-fuse）支持。4.17 及更高版本的 Linux 内核客户端支持 SUSE Enterprise Storage 7.1 集群上的 CephFS 配额。内核客户端（甚至最近的版本）无法处理较旧集群的配额，即便它们可以设置配额扩展属性也是如此。SLE12-SP3（及更高版本）内核始终包含必要的向后移植能力，以处理配额。

与基于路径的挂载限制搭配使用时，需谨慎配置配额。

客户端需要有权访问配置了配额的目录 Inode，才能强制实施这些配额。如果根据 MDS 用户权限，客户端对特定路径（例如 `/home/user`）的访问受到限制，并且配额是对其无权访问的祖先目录 (`/home`) 配置的，则客户端将无法强制实施该配额。使用基于路径的访问限制时，请务必对客户端可以访问的目录（例如 `/home/user` 或 `/home/user/quota_dir`）配置配额。

23.6.2 配置 CephFS 配额

您可以使用虚拟扩展属性来配置 CephFS 配额：

`ceph.quota.max_files`

配置**文件**限制。

`ceph.quota.max_bytes`

配置**字节**限制。

如果某个目录 Inode 存在这些属性，即表示该位置配置了配额。如果不存在，则表示未对该目录配置配额（即便可能对父目录配置了配额）。

要设置 100 MB 的配额，请运行以下命令：

```
cephuser@mds > setfattr -n ceph.quota.max_bytes -v 100000000 /SOME/DIRECTORY
```

要设置 10,000 个文件的配额，请运行以下命令：

```
cephuser@mds > setfattr -n ceph.quota.max_files -v 10000 /SOME/DIRECTORY
```

要查看配额设置，请运行以下命令：

```
cephuser@mds > getfattr -n ceph.quota.max_bytes /SOME/DIRECTORY
```

```
cephuser@mds > getfattr -n ceph.quota.max_files /SOME/DIRECTORY
```

注意：未设置配额

如果扩展属性的值为“0”，表示未设置配额。

要删除配额，请运行以下命令：

```
cephuser@mds > setfattr -n ceph.quota.max_bytes -v 0 /SOME/DIRECTORY
cephuser@mds > setfattr -n ceph.quota.max_files -v 0 /SOME/DIRECTORY
```

23.7 管理 CephFS 快照

在截取 CephFS 快照时，快照会创建此时间点文件系统的只读视图。您可以在任何目录中创建快照。快照将涵盖文件系统中指定目录下的所有数据。创建快照后，系统会从各客户端异步清理缓冲数据。因此，创建快照的速度十分快。

重要：多个文件系统

如果您有多个 CephFS 文件系统在共享一个存储池（通过名称空间），则这些文件系统的快照将会发生冲突，并且删除其中一个快照将会导致共享同一存储池的其他快照缺少文件数据。

23.7.1 创建快照

对于新文件系统，系统默认会启用 CephFS 快照特性。要对现有文件系统启用该特性，请运行以下命令：

```
cephuser@adm > ceph fs set CEPHFS_NAME allow_new_snaps true
```

启用快照后，CephFS 中的所有目录都将包含一个特殊的 `.snap` 子目录。

注意

这是个**虚拟**子目录。它不会出现在父目录的目录列表中，但您不能使用 `.snap` 名称作为文件或目录名称。要访问 `.snap` 目录，需要采用显式访问方式，例如：

```
> ls -la /CEPHFS_MOUNT/.snap/
```

重要：内核客户端限制

CephFS 内核客户端存在一个限制：一旦文件系统中所含快照数量超过 400 个，它们将无法处理。快照的数量应始终低于此上限，无论使用哪个客户端均如此。如果使用较旧的 CephFS 客户端（例如 SLE12-SP3），请记住，快照数量超过 400 个对运行十分有害，因为这会导致客户端崩溃。

提示：自定义快照子目录名称

您可以通过设置 `client snapdir` 设置来为快照子目录配置其他名称。

要创建快照，请以自定义名称在 `.snap` 目录下创建子目录。例如，要创建目录 `/CEPHFS_MOUNT/2/3/` 的快照，请运行以下命令：

```
> mkdir /CEPHFS_MOUNT/2/3/.snap/CUSTOM_SNAPSHOT_NAME
```

23.7.2 删除快照

要删除某个快照，请在 `.snap` 目录中删除该快照的子目录：

```
> rmdir /CEPHFS_MOUNT/2/3/.snap/CUSTOM_SNAPSHOT_NAME
```

24 通过 Samba 导出 Ceph 数据

本章介绍如何通过 Samba/CIFS 共享导出 Ceph 集群中存储的数据，以便您可从 Windows* 客户端计算机轻松访问这些数据。另外还介绍了有助于您将 Ceph Samba 网关配置为加入到 Windows* 域中的 Active Directory 以进行用户身份验证和授权的信息。

注意：Samba 网关性能

由于客户端与存储区之间的额外网络跃点会导致协议开销增加并产生额外的延迟，因此与使用本机 Ceph 客户端相比，通过 Samba 网关访问 CephFS 可能会使应用性能大幅降低。

24.1 通过 Samba 共享导出 CephFS

警告：跨协议访问

本机 CephFS 和 NFS 客户端不受通过 Samba 获取的文件锁定限制，反之亦然。如果通过其他方式访问 CephFS 支持的 Samba 共享路径，则依赖跨协议文件锁定的应用可能会出现数据损坏情况。

24.1.1 配置和导出 Samba 软件包

要配置和导出 Samba 共享，需要安装以下软件包：`samba-ceph` 和 `samba-winbind`。如尚未安装这些软件包，请运行以下命令进行安装：

```
cephuser@smb > zypper install samba-ceph samba-winbind
```

24.1.2 单个网关示例

在导出 Samba 共享的准备过程中，需要选择用于充当 Samba 网关的合适节点。该节点必须能访问 Ceph 客户端网络，同时要有足够的 CPU、内存和网络资源。

可以通过 CTDB 和 SUSE Linux Enterprise High Availability Extension 提供故障转移功能。有关 HA 设置的详细信息，请参见第 24.1.3 节 “配置高可用性”。

1. 请确保集群中已存在一个正常工作的 CephFS。

2. 在 Ceph 管理节点上创建一个特定于 Samba 网关的密钥环，并将其复制到两个 Samba 网关节点：

```
cephuser@adm > ceph auth get-or-create client.samba.gw mon 'allow r' \
 osd 'allow *' mds 'allow *' -o ceph.client.samba.gw.keyring
cephuser@adm > scp ceph.client.samba.gw.keyring SAMBA_NODE:/etc/ceph/
```

   将 SAMBA_NODE 替换为 Samba 网关节点的名称。

3. 在 Samba 网关节点上执行以下步骤。将 Samba 连同 Ceph 集成软件包一起安装：

```
cephuser@smb > sudo zypper in samba samba-ceph
```

4. 将 /etc/samba/smb.conf 文件的默认内容替换为以下内容：

```
[global]
  netbios name = SAMBA-GW
  clustering = no
  idmap config * : backend = tdb2
  passdb backend = tdbsam
  # disable print server
  load printers = no
  smbd: backgroundqueue = no

[SHARE_NAME]
  path = CEPHFS_MOUNT
  read only = no
  oplocks = no
  kernel share modes = no
```

   在启动具有内核 CephFS 共享配置的 Samba 之前，必须先挂载上面的 CEPHFS_MOUNT 路径。请参见第 23.3 节 “在 /etc/fstab 中挂载 CephFS”。

上面的共享配置使用 Linux 内核 CephFS 客户端，这是出于性能原因建议使用的客户端。作为替代方案，也可以使用 Samba `vfs_ceph` 模块与 Ceph 集群通讯。以下说明仅适用于旧版，不建议用于新的 Samba 部署：

```
[SHARE_NAME]
  path = /
  vfs objects = ceph
  ceph: config_file = /etc/ceph/ceph.conf
  ceph: user_id = samba.gw
  read only = no
  oplocks = no
  kernel share modes = no
```

提示：Oplocks 和共享模式

`oplocks`（也称为 SMB2+ 租用）可通过加速客户端缓存来提升性能，不过如果将其他 CephFS 客户端（例如内核 `mount.ceph`、FUSE 或 NFS Ganesha）与 Samba 一起部署，该机制目前并不安全。

如果所有 CephFS 文件系统路径访问都专由 Samba 处理，则可安全启用 `oplocks` 参数。

目前，在使用 CephFS vfs 扩展模块运行的共享中，需要禁用 `kernel share modes`，文件处理工作才能正常进行。

重要：允许访问

Samba 可将 SMB 用户和组映射到本地帐户。可以通过以下命令为本地用户指定 Samba 共享访问的密码：

```
# smbpasswd -a USERNAME
```

对于成功的 I/O，共享路径的访问控制列表 (ACL) 需要允许访问通过 Samba 连接的用户。您可以通过 CephFS 内核客户端暂时挂载，并对共享路径使用 **chmod**、**chown** 或 **setfacl** 实用程序来修改 ACL。例如，要允许所有用户进行访问，请运行以下命令：

```
# chmod 777 MOUNTED_SHARE_PATH
```

24.1.2.1 启动 Samba 服务

使用以下命令可启动或重启动独立的 Samba 服务：

```
# systemctl restart smb.service
# systemctl restart nmb.service
# systemctl restart winbind.service
```

要确保会在引导时启动 Samba 服务，请通过以下命令将其启用：

```
# systemctl enable smb.service
# systemctl enable nmb.service
# systemctl enable winbind.service
```

提示：可选的 nmb 和 winbind 服务

如果不需要浏览网络共享，则无需启用和启动 `nmb` 服务。

仅当配置为 Active Directory 域成员时才需要 `winbind` 服务。请参见第 24.2 节 “加入 Samba 网关和 Active Directory”。

24.1.3 配置高可用性

重要：不支持透明故障转移

尽管多节点 Samba + CTDB 部署比单节点部署的高可用性更佳（请参见第 24 章 “通过 Samba 导出 Ceph 数据”），但它并不支持客户端透明故障转移。应用可能会在 Samba 网关节点发生故障时出现短暂的中断。

本节提供一个示例来说明如何设置 Samba 服务器的双节点高可用性配置。该设置需要 SUSE Linux Enterprise High Availability Extension。两个节点名为 `earth` (`192.168.1.1`) 和 `mars` (`192.168.1.2`)。

有关 SUSE Linux Enterprise High Availability Extension 的详细信息，请参见 https://documentation.suse.com/sle-ha/15-SP1/。

此外，使用两个浮动虚拟 IP 地址可让客户端连接到服务，不管该服务在哪个物理节点上运行均如此。`192.168.1.10` 用于通过 Hawk2 进行集群管理，`192.168.2.1` 专门用于 CIFS 导出。这样，以后便可更轻松地应用安全限制。

以下过程介绍示例安装。https://documentation.suse.com/sle-ha/15-SP1/single-html/SLE-HA-install-quick/ 上提供了更多详细信息。

1. 在管理节点上创建一个特定于 Samba 网关的密钥环，并将其复制到上述两个节点上：

```
cephuser@adm > ceph auth get-or-create client.samba.gw mon 'allow r' \
    osd 'allow *' mds 'allow *' -o ceph.client.samba.gw.keyring
cephuser@adm > scp ceph.client.samba.gw.keyring earth:/etc/ceph/
cephuser@adm > scp ceph.client.samba.gw.keyring mars:/etc/ceph/
```

2. SLE-HA 设置需要一个隔离设备，以避免在活动集群节点变为未同步时出现**节点分裂**情况。为此，您可以将 Ceph RBD 映像与 Stonith 块设备 (SBD) 搭配使用。有关更多详细信息，请参见https://documentation.suse.com/sle-ha/15-SP1/single-html/SLE-HA-guide/#sec-ha-storage-protect-fencing-setup。
   如果该映像尚不存在，则创建一个名为 `rbd` 的 RBD 存储池（参见第 18.1 节 “创建存储池”），并将其与 `rbd` 相关联（参见第 18.5.1 节 “将存储池与应用关联”）。然后创建一个名为 `sbd01` 的相关 RBD 映像：

```
cephuser@adm > ceph osd pool create rbd
cephuser@adm > ceph osd pool application enable rbd rbd
cephuser@adm > rbd -p rbd create sbd01 --size 64M --image-shared
```

3. 准备好 `earth` 和 `mars`，以托管 Samba 服务：

   a. 在继续下一步之前，请确保已安装以下软件包：`ctdb`、`tdb-tools` 和 `samba`。

   ```
   # zypper in ctdb tdb-tools samba samba-ceph
   ```

   b. 确保已停止并禁用 Samba 和 CTDB 服务：

   ```
   # systemctl disable ctdb
   # systemctl disable smb
   # systemctl disable nmb
   # systemctl disable winbind
   # systemctl stop ctdb
   # systemctl stop smb
   # systemctl stop nmb
   # systemctl stop winbind
   ```

   c. 在所有节点上打开防火墙的端口 4379。这是为了使 CTDB 能够与其他集群节点通讯。

4. 在 `earth` 上创建 Samba 的配置文件。这些文件稍后将自动同步到 `mars`。

   a. 在 `/etc/ctdb/nodes` 文件中插入 Samba 网关节点的专用 IP 地址列表。有关更多详细信息，请参见 ctdb 手册页 (**`man 7 ctdb`**)。

   ```
   192.168.1.1
   192.168.1.2
   ```

   b. 配置 Samba。在 `/etc/samba/smb.conf` 的 `[global]` 部分中添加下面几行。使用所选的主机名取代 `CTDB-SERVER`（集群中的所有节点将显示为一个使用此名称的大节点，以方便操作）。另外，添加一个共享定义，以 `SHARE_NAME` 为例：

   ```
   [global]
     netbios name = SAMBA-HA-GW
   ```

```
  clustering = yes
  idmap config * : backend = tdb2
  passdb backend = tdbsam
  ctdbd socket = /var/lib/ctdb/ctdb.socket
  # disable print server
  load printers = no
  smbd: backgroundqueue = no

[SHARE_NAME]
  path = /
  vfs objects = ceph
  ceph: config_file = /etc/ceph/ceph.conf
  ceph: user_id = samba.gw
  read only = no
  oplocks = no
  kernel share modes = no
```

请注意，`/etc/ctdb/nodes` 和 `/etc/samba/smb.conf` 文件需要在所有 Samba 网关节点上都保持一致。

5. 安装并引导 SUSE Linux Enterprise High Availability 集群。

   a. 在 `earth` 和 `mars` 上注册 SUSE Linux Enterprise High Availability Extension：

   ```
   root@earth # SUSEConnect -r ACTIVATION_CODE -e E_MAIL
   ```

   ```
   root@mars # SUSEConnect -r ACTIVATION_CODE -e E_MAIL
   ```

   b. 在两个节点上安装 `ha-cluster-bootstrap`：

   ```
   root@earth # zypper in ha-cluster-bootstrap
   ```

   ```
   root@mars # zypper in ha-cluster-bootstrap
   ```

   c. 通过 `在两个 Samba 网关上映射 RBD 映像 sbd01rbdmap.service`。
   编辑 `/etc/ceph/rbdmap` 并添加 SBD 映像的条目：

   ```
   rbd/sbd01 id=samba.gw,keyring=/etc/ceph/ceph.client.samba.gw.keyring
   ```

启用并启动 `rbdmap.service`：

```
root@earth # systemctl enable rbdmap.service && systemctl start
 rbdmap.service
root@mars # systemctl enable rbdmap.service && systemctl start
 rbdmap.service
```

`/dev/rbd/rbd/sbd01` 设备应在两个 Samba 网关上都可用。

d. 在 `earth` 上初始化集群并让 `mars` 加入集群。

```
root@earth # ha-cluster-init
```

```
root@mars # ha-cluster-join -c earth
```

重要

在初始化和加入集群的过程中，会以交互方式询问您是否使用 SBD。点击 `y` 进行确认，然后将 `/dev/rbd/rbd/sbd01` 指定为存储设备的路径。

6. 检查集群的状态。您应该会看到两个节点都已添加到集群中：

```
root@earth # crm status
2 nodes configured
1 resource configured

Online: [ earth mars ]

Full list of resources:

 admin-ip       (ocf::heartbeat:IPaddr2):       Started earth
```

7. 在 `earth` 上执行以下命令，以配置 CTDB 资源：

```
root@earth # crm configure
crm(live)configure# primitive ctdb ocf:heartbeat:CTDB params \
    ctdb_manages_winbind="false" \
```

```
    ctdb_manages_samba="false" \
    ctdb_recovery_lock="!/usr/lib64/ctdb/ctdb_mutex_ceph_rados_helper
        ceph client.samba.gw cephfs_metadata ctdb-mutex"
    ctdb_socket="/var/lib/ctdb/ctdb.socket" \
        op monitor interval="10" timeout="20" \
        op start interval="0" timeout="200" \
        op stop interval="0" timeout="100"
crm(live)configure# primitive smb systemd:smb \
    op start timeout="100" interval="0" \
    op stop timeout="100" interval="0" \
    op monitor interval="60" timeout="100"
crm(live)configure# primitive nmb systemd:nmb \
    op start timeout="100" interval="0" \
    op stop timeout="100" interval="0" \
    op monitor interval="60" timeout="100"
crm(live)configure# primitive winbind systemd:winbind \
    op start timeout="100" interval="0" \
    op stop timeout="100" interval="0" \
    op monitor interval="60" timeout="100"
crm(live)configure# group g-ctdb ctdb winbind nmb smb
crm(live)configure# clone cl-ctdb g-ctdb meta interleave="true"
crm(live)configure# commit
```

提示：可选的 nmb 和 winbind 基元

如果不需要浏览网络共享，则无需添加 `nmb` 基元。

仅当配置为 Active Directory 域成员时才需要 `winbind` 基元。请参见第 24.2 节“加入 Samba 网关和 Active Directory”。

配置选项 `ctdb_recovery_lock` 中的二进制文件 **`/usr/lib64/ctdb/ctdb_mutex_ceph_rados_helper`** 中按如下顺序包含以下参数：`CLUSTER_NAME`、`CEPHX_USER`、`RADOS_POOL` 和 `RADOS_OBJECT`。

可追加一个额外的锁定超时参数来覆盖所用的默认值（10 秒）。使用更高的值将会增加 CTDB 恢复主节点故障转移时间，然而使用更低的值可能会导致不正确地将恢复主节点检测为停用状态，以致触发摆动故障转移。

8. 添加集群 IP 地址：

```
crm(live)configure# primitive ip ocf:heartbeat:IPaddr2
    params ip=192.168.2.1 \
    unique_clone_address="true" \
    op monitor interval="60" \
    meta resource-stickiness="0"
crm(live)configure# clone cl-ip ip \
    meta interleave="true" clone-node-max="2" globally-unique="true"
crm(live)configure# colocation col-with-ctdb 0: cl-ip cl-ctdb
crm(live)configure# order o-with-ctdb 0: cl-ip cl-ctdb
crm(live)configure# commit
```

如果 `unique_clone_address` 设置为 `true`，IPaddr2 资源代理将向指定的地址添加一个克隆 ID，从而导致出现三个不同的 IP 地址。这些地址通常是不需要的，但有助于实现负载平衡。有关此主题的更多信息，请参见 https://documentation.suse.com/sle-ha/15-SP1/single-html/SLE-HA-guide/#cha-ha-lb。

9. 检查结果：

```
root@earth # crm status
Clone Set: base-clone [dlm]
     Started: [ factory-1 ]
     Stopped: [ factory-0 ]
 Clone Set: cl-ctdb [g-ctdb]
     Started: [ factory-1 ]
     Started: [ factory-0 ]
 Clone Set: cl-ip [ip] (unique)
     ip:0       (ocf:heartbeat:IPaddr2):       Started factory-0
     ip:1       (ocf:heartbeat:IPaddr2):       Started factory-1
```

10. 从客户端计算机进行测试。在 Linux 客户端上运行以下命令，以检查能否从系统复制文件以及将文件复制到系统：

```
# smbclient //192.168.2.1/myshare
```

24.1.3.1 重启动 HA Samba 资源

发生任何 Samba 或 CTDB 配置更改后，可能需要重启动 HA 资源才能使更改生效。这可通过以下命令实现：

```
# crm resource restart cl-ctdb
```

24.2 加入 Samba 网关和 Active Directory

您可以将 Ceph Samba 网关配置为支持 Active Directory (AD) 的 Samba 域的成员。作为 Samba 域成员，您可以针对来自导出 CephFS 的文件和目录在本地访问列表 (ACL) 中使用域用户和组。

24.2.1 准备 Samba 安装

本节介绍在配置 Samba 自身之前，您需要执行的一些准备步骤。首先，您需要准备一个干净的环境，这样有助于防止混淆，并可确认以前所安装的 Samba 系统中的文件没有与新安装的域成员混用。

提示：同步时钟

所有 Samba 网关节点的时钟都需要与 Active Directory 域控制器保持同步。时钟偏差可能会导致身份验证失败。

确认没有正在运行的 Samba 或名称缓存进程：

```
cephuser@smb > ps ax | egrep "samba|smbd|nmbd|winbindd|nscd"
```

如果输出列出了任何 `samba`、`smbd`、`nmbd`、`winbindd` 或 `nscd` 进程，请将其停止。

如果您之前在此主机上运行过 Samba 安装，请删除 `/etc/samba/smb.conf` 文件。另外，请删除所有 Samba 数据库文件（例如 `*.tdb` 和 `*.ldb` 文件）。要列出包含 Samba 数据库的目录，请运行以下命令：

```
cephuser@smb > smbd -b | egrep "LOCKDIR|STATEDIR|CACHEDIR|PRIVATE_DIR"
```

24.2.2 确认 DNS

Active Directory (AD) 使用 DNS 来查找域控制器 (DC) 和服务（例如 Kerberos）。因此，AD 域成员和服务器需要能够解析 AD DNS 区域。

确认已正确配置 DNS 且正向和反向查找均可正确解析，例如：

```
cephuser@adm > nslookup DC1.domain.example.com
Server:         10.99.0.1
Address:        10.99.0.1#53

Name:   DC1.domain.example.com
Address: 10.99.0.1
```

```
cephuser@adm > 10.99.0.1
Server:        10.99.0.1
Address: 10.99.0.1#53

1.0.99.10.in-addr.arpa name = DC1.domain.example.com.
```

24.2.3 解析 SRV 记录

AD 使用 SRV 记录查找服务（例如 Kerberos 和 LDAP）。要确认能否正确解析 SRV 记录，请使用 **`nslookup`** 交互外壳，例如：

```
cephuser@adm > nslookup
Default Server:  10.99.0.1
Address:  10.99.0.1

> set type=SRV
> _ldap._tcp.domain.example.com.
Server:  UnKnown
Address:  10.99.0.1

_ldap._tcp.domain.example.com   SRV service location:
          priority       = 0
          weight         = 100
```

```
          port            = 389
          svr hostname    = dc1.domain.example.com
domain.example.com      nameserver = dc1.domain.example.com
dc1.domain.example.com  internet address = 10.99.0.1
```

24.2.4 配置 Kerberos

Samba 支持 Heimdal 和 MIT Kerberos 后端。要对域成员配置 Kerberos，请在您的 `/etc/krb5.conf` 文件中进行以下设置：

```
[libdefaults]
 default_realm = DOMAIN.EXAMPLE.COM
 dns_lookup_realm = false
 dns_lookup_kdc = true
```

上面的示例为 DOMAIN.EXAMPLE.COM 领域配置了 Kerberos。我们不建议在 `/etc/krb5.conf` 文件中设置任何进一步的参数。如果您的 `/etc/krb5.conf` 中包含 `include` 行，则您**必须**删除此行，否则该文件将无法工作。

24.2.5 解析本地主机名

当您将主机加入域中时，Samba 会尝试在 AD DNS 区域中注册其主机名。为此，**`net`** 实用程序需要能够使用 DNS 或 `/etc/hosts` 文件中的正确条目来解析主机名。

要确认您的主机名解析正确，请使用 **`getent hosts`** 命令：

```
cephuser@adm > getent hosts example-host
10.99.0.5       example-host.domain.example.com    example-host
```

主机名和 FQDN 不得解析为 127.0.0.1 IP 地址或与域成员的 LAN 接口上所用 IP 地址不同的任何 IP 地址。如果未显示输出或主机解析为错误的 IP 地址，而您未在使用 DHCP，请在 `/etc/hosts` 文件中设置正确的条目：

```
127.0.0.1       localhost
10.99.0.5       example-host.samdom.example.com    example-host
```

提示：DHCP 和 /etc/hosts

如果您在使用 DHCP，请检查 `/etc/hosts` 是否仅包含“127.0.0.1”行。如果仍然有问题，请联系 DHCP 服务器的管理员。

如果您需要添加计算机主机名的别名，请将别名添加到以计算机 IP 地址开头的行的末尾，切勿添加到“127.0.0.1”行。

24.2.6 配置 Samba

本节说明有关您需要在 Samba 配置中包含的特定配置选项的信息。

Active Directory 域成员资格主要通过在 `/etc/samba/smb.conf` 的 `[global]` 部分中设置 `security = ADS` 以及相应的 Kerberos 领域和 ID 映射参数进行配置。

```
[global]
  security = ADS
  workgroup = DOMAIN
  realm = DOMAIN.EXAMPLE.COM
  ...
```

24.2.6.1 选择 winbindd 中用于 ID 映射的后端

如果您需要让您的用户使用不同的登录外壳和/或 Unix 主目录路径，或者让他们在任何地方都使用相同的 ID，您需要使用 winbind“ad”后端并将 RFC2307 属性添加到 AD 中。

重要：RFC2307 属性和 ID 号

创建用户或组时，系统不会自动添加 RFC2307 属性。

DC 上找到的 ID 号（3000000 范围内的编号）**不是** RFC2307 属性，将不会在 Unix 域成员上使用。如果您需要在任何地方都使用相同的 ID 号，请将 `uidNumber` 和 `gidNumber` 属性添加到 AD 中，并在 Unix 域成员上使用 winbind“ad”后端。如果您决定将 `uidNumber` 和 `gidNumber` 属性添加到 AD 中，请不要使用 3000000 范围内的编号。

如果您的用户仅将 Samba AD DC 用于身份验证目的，而不会在其上存储数据或登录到其中，则您可使用 winbind“rid”后端。如此，系统会依据 Windows* RID 来计算用户和组 ID。如果您在每个 Unix 域成员上的 `smb.conf` 中都使用相同的 `[global]` 段落，您将会获得相同的 ID。如果您使用“rid”后端，则不需要向 AD 添加任何内容，系统将会忽略 RFC2307 属性。使用“rid”后端时，请在 `smb.conf` 中设置 `template shell` 和 `template homedir` 参数。它们是全局设置，会为所有用户设置相同的登录外壳和 Unix 主目录路径（RFC2307 属性则不同，可用来分别设置不同的 Unix 主目录路径和外壳）。

如果您需要让您的用户和组在任何地方都使用相同的 ID，但只需要为他们设置相同的登录外壳和相同的 Unix 主目录路径，还可使用另一种方法来设置 Samba。您可以通过使用 winbind“ad”后端并在 `smb.conf` 中使用模板行来实现此目的。使用此方法时，您仅需将 `uidNumber` 和 `gidNumber` 属性添加到 AD 中。

提示：有关 ID 映射后端的更多信息

有关可用 ID 映射后端的详细信息，请参见下列相关手册页：**`man 8 idmap_ad`**、**`man 8 idmap_rid`** 和 **`man 8 idmap_autorid`**。

24.2.6.2 设置用户和组 ID 范围

决定好使用哪个 winbind 后端后，您需要在 `smb.conf` 中设置 `idmap config` 选项来指定要使用的范围。默认情况下，Unix 域成员上预留有多个用户和组 ID 块：

表 24.1： 默认用户和组 ID 块

ID	范围
0-999	本地系统用户和组。
从 1000 开始	本地 Unix 用户和组。
从 10000 开始	DOMAIN 用户和组。

根据上述范围，您不应将“*”或“DOMAIN”范围设置为 999 以内，因为它们会与本地系统用户和组发生冲突。您还应为任何本地 Unix 用户和组留出余地，因此将 `idmap config` 范围设置为从 3000 开始是不错的折衷方法。

您需要确定“DOMAIN”可能会增长到多大，并决定是否打算建立任何受信任的域。然后，便可按如下所示来设置 `idmap config` 范围：

表 24.2： ID 范围

域	范围
*	3000-7999
DOMAIN	10000-999999
TRUSTED	1000000-9999999

24.2.6.3 将域管理员帐户映射到本地 root 用户

Samba 可让您将域帐户映射到本地帐户。通过此功能，您可以用不同于客户端上请求执行操作的帐户的用户身份在域成员的文件系统上执行文件操作。

提示：映射域管理员（可选）

将域管理员映射到本地 `root` 帐户属于可选操作。请仅在域管理员需要能够使用 `root` 权限在域成员上执行文件操作时配置该映射。请注意，将 Administrator 映射到 `root` 帐户后，其便不能以“Administrator”身份登录 Unix 域成员。

要将域管理员映射到本地 `root` 帐户，请执行以下步骤：

1. 将以下参数添加到 `smb.conf` 文件的 `[global]` 段落：

```
username map = /etc/samba/user.map
```

2. 创建包含以下内容的 `/etc/samba/user.map` 文件：

```
!root = DOMAIN\Administrator
```

重要

使用“ad”ID 映射后端时，请不要为域管理员帐户设置 `uidNumber` 属性。如果为域管理员帐户设置了该属性，其值会覆盖 `root` 用户的本地 UID“0”，因而会导致映射失败。

有关详细信息，请参见 `smb.conf` 手册页 (**`man 5 smb.conf`**) 中的 `username map` 参数。

24.2.7 加入 Active Directory 域

要将主机加入 Active Directory，请运行以下命令：

```
cephuser@smb > net ads join -U administrator
Enter administrator's password: PASSWORD
Using short domain name -- DOMAIN
Joined EXAMPLE-HOST to dns domain 'DOMAIN.example.com'
```

24.2.8 配置名称服务开关

要使域用户和组可供本地系统使用，您需要启用名称服务开关 (NSS) 库。在 `/etc/nsswitch.conf` 文件中，将 `winbind` 条目追加到以下数据库：

```
passwd: files winbind
group:  files winbind
```

重要：需考虑的要点

- 将 `files` 条目指定为这两个数据库的第一个源。这可让 NSS 在查询 `winbind` 服务之前，先从 /etc/passwd 和 /etc/group 文件中查找域用户和组。
- 不要将 `winbind` 条目添加到 NSS `shadow` 数据库中。这样做可能会导致 **`wbinfo`** 实用程序失败。
- 不要在本地 `/etc/passwd` 文件中使用与域中相同的用户名。

24.2.9 启动服务

在配置更改后，根据第 24.1.2.1 节 “启动 Samba 服务”或第 24.1.3.1 节 “重启动 HA Samba 资源”中所述重启动 Samba 服务。

24.2.10 测试 winbindd 连接性

24.2.10.1 发送 winbindd ping

要验证 `winbindd` 服务能否连接到 AD 域控制器 (DC) 或主域控制器 (PDC)，请输入以下命令：

```
cephuser@smb > wbinfo --ping-dc
checking the NETLOGON for domain[DOMAIN] dc connection to
 "DC.DOMAIN.EXAMPLE.COM" succeeded
```

如果前一条命令失败，请确认 `winbindd` 服务正在运行且 `smb.conf` 文件设置正确。

24.2.10.2 查找域用户和组

您可以使用 `libnss_winbind` 库查找域用户和组。例如，要查找域用户“DOMAIN\demo01”，请运行以下命令：

```
cephuser@smb > getent passwd DOMAIN\\demo01
DOMAIN\demo01:*:10000:10000:demo01:/home/demo01:/bin/bash
```

要查找域组“Domain Users”，请运行以下命令：

```
cephuser@smb > getent group "DOMAIN\\Domain Users"
DOMAIN\domain users:x:10000:
```

24.2.10.3 为域用户和组指定文件权限

借助名称服务开关 (NSS) 库，您可以在命令中使用域用户帐户和组。例如，要将文件的所有者设置为“demo01”域用户，并将组设置为“Domain Users”域组，请输入以下命令：

```
cephuser@smb > chown "DOMAIN\\demo01:DOMAIN\\domain users" file.txt
```

25 NFS Ganesha

NFS Ganesha 是一种 NFS 服务器，它在用户地址空间中运行，而不是作为操作系统内核的一部分运行。借助 NFS Ganesha，您可以插入自己的存储机制（例如 Ceph），并从任何 NFS 客户端访问它。有关安装说明，请参阅《部署指南》, 第 8 章 “使用 cephadm 部署其余核心服务”，第 8.3.6 节 “部署 NFS Ganesha”。

注意：NFS Ganesha 性能

由于客户端与存储区之间的额外网络跃点会导致协议开销增加并产生额外的延迟，因此与使用本机 CephFS 相比，通过 NFS 网关访问 Ceph 可能会大幅降低应用性能。

每个 NFS Ganesha 服务都含有一个配置层次结构，其中包含：

- 引导 `ganesha.conf`
- 每个服务的 RADOS 通用配置对象
- 每个导出的 RADOS 配置对象

引导配置是要在容器中启动 `nfs-ganesha` 守护进程的最低配置。每个引导配置都将包含一个 `%url` 指令，该指令中包含来自 RADOS 通用配置对象的任何额外配置。通用配置对象可以为导出 RADOS 配置对象中定义的每个 NFS 导出包含额外的 `%url` 指令。

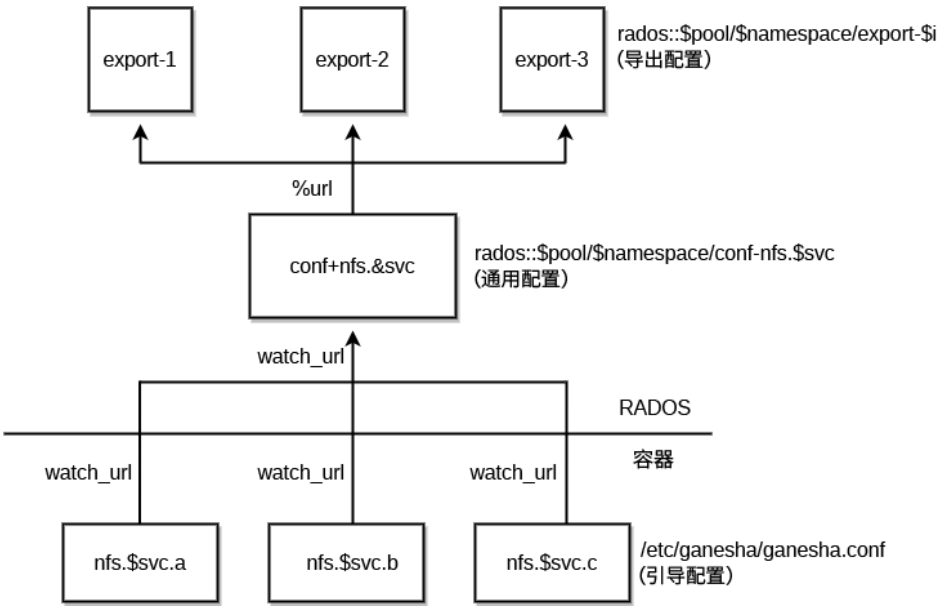


图 25.1： NFS GANESHA 结构

25.1 创建 NFS 服务

指定 Ceph 服务部署的推荐方法是创建一个 YAML 格式的文件，其中包含所要部署服务的规范。您可以为每种类型的服务创建单独的规范文件，也可以在一个文件中指定多个（或所有）服务类型。

根据您的选择，您将需要更新或创建相关的 YAML 格式文件来创建 NFS Ganesha 服务。有关创建该文件的详细信息，请参见《部署指南》, 第 8 章 “使用 cephadm 部署其余核心服务”，第 8.2 节 “服务和归置规范”。

更新或创建该文件之后，请执行以下命令以创建 `nfs-ganesha` 服务：

```
cephuser@adm > ceph orch apply -i FILE_NAME
```

25.2 启动或重启动 NFS Ganesha

重要

启动 NFS Ganesha 服务不会自动导出 CephFS 文件系统。要导出 CephFS 文件系统，请长假导出配置文件。有关更多详细信息，请参见第 25.4 节 “创建 NFS 导出”。

要启动 NFS Ganesha 服务，请运行：

```
cephuser@adm > ceph orch start nfs.SERVICE_ID
```

要重启动 NFS Ganesha 服务，请运行：

```
cephuser@adm > ceph orch restart nfs.SERVICE_ID
```

如果您只想重启动单个 NFS Ganesha 守护进程，请运行：

```
cephuser@adm > ceph orch daemon restart nfs.SERVICE_ID
```

启动或重启动 NFS Ganesha 时，NFS v4 会有 90 秒的超时宽限期。在宽限期内，会主动拒绝来自客户端的新请求。因此，当 NFS 处于宽限期内，客户端可能会发生请求处理速度变慢的情况。

25.3 列出 NFS 恢复存储池中的对象

执行以下命令以列出 NFS 恢复存储池中的对象：

```
cephuser@adm > rados --pool POOL_NAME --namespace NAMESPACE_NAME ls
```

25.4 创建 NFS 导出

您可以在 Ceph Dashboard 中创建 NFS 导出，也可以在命令行上手动创建。要使用 Ceph Dashboard 创建 NFS 导出，请参见第 7 章 “管理 NFS Ganesha”，具体而言，请参见第 7.1 节 “创建 NFS 导出项”。

要手动创建 NFS 导出，请创建导出的配置文件。例如，包含以下内容的 `/tmp/export-1` 文件：

```
EXPORT {
    export_id = 1;
    path = "/";
    pseudo = "/";
    access_type = "RW";
    squash = "no_root_squash";
    protocols = 3, 4;
    transports = "TCP", "UDP";
    FSAL {
        name = "CEPH";
        user_id = "admin";
        filesystem = "a";
        secret_access_key = "SECRET_ACCESS_KEY";
    }
}
```

创建并保存新导出的配置文件后，运行以下命令来创建该导出：

```
rados --pool POOL_NAME --namespace NAMESPACE_NAME
 put EXPORT_NAME EXPORT_CONFIG_FILE
```

例如：

```
cephuser@adm > rados --pool example_pool --namespace example_namespace put
 export-1 /tmp/export-1
```

注意

应修改 FSAL 块以包含所需的 `cephx` 用户 ID 和秘密访问密钥。

25.5 确认 NFS 导出

NFS v4 将在伪文件系统的根目录下构建导出列表。您可以通过挂载 NFS Ganesha 服务器节点的 `/` 来确认 NFS 共享是否已导出：

```
# mount -t nfs nfs_ganesha_server_hostname:/ /path/to/local/mountpoint
# ls /path/to/local/mountpoint cephfs
```

注意：NFS Ganesha 仅限 v4

默认情况下，cephadm 将配置 NFS v4 服务器。NFS v4 不会与 `rpcbind` 或 `mountd` 守护进程进行交互。NFS 客户端工具（例如 **`showmount`**）将不会显示任何已配置的导出。

25.6 挂载 NFS 导出

要在客户端主机上挂载导出的 NFS 共享，请运行：

```
# mount -t nfs nfs_ganesha_server_hostname:/ /path/to/local/mountpoint
```

25.7 多个 NFS Ganesha 集群

可以定义多个 NFS Ganesha 集群。如此便可：

- 分隔 NFS Ganesha 集群，以便访问 CephFS。

V 与虚拟化工具集成

26 libvirt 和 Ceph 356

27 Ceph 用作 QEMU KVM 实例的后端 362

26 libvirt 和 Ceph

libvirt 库在超级管理程序接口与使用这些接口的软件应用之间建立了一个虚拟机抽象层。使用 libvirt，开发人员和系统管理员可将工作重心放在通用管理框架、通用 API、通用外壳接口 (**virsh**) 以及诸多不同的超级管理程序（包括 QEMU/KVM、Xen、LXC 或 VirtualBox）上。

Ceph 块设备支持 QEMU/KVM。您可以通过与 libvirt 连接的软件来使用 Ceph 块设备。云解决方案使用 libvirt 来与 QEMU/KVM 交互，而 QEMU/KVM 通过 librbd 来与 Ceph 块设备交互。

要创建使用 Ceph 块设备的 VM，请按以下各节中所述的过程操作。在示例中，我们分别使用了 libvirt-pool、client.libvirt 和 new-libvirt-image 作为存储池名称、用户名和映像名称。您可以根据个人喜好使用任何值，但在执行后续过程中的命令时，请务必替换这些值。

26.1 将 Ceph 配置为与 libvirt 搭配使用

要将 Ceph 配置为与 libvirt 搭配使用，请执行以下步骤：

1. 创建存储池。下面的示例使用存储池名称 libvirt-pool 和 128 个归置组。

   ```
   cephuser@adm > ceph osd pool create libvirt-pool 128 128
   ```

   校验该存储池是否存在。

   ```
   cephuser@adm > ceph osd lspools
   ```

2. 创建 Ceph 用户。下面的示例使用 Ceph 用户名 client.libvirt 并引用 libvirt-pool。

   ```
   cephuser@adm > ceph auth get-or-create client.libvirt mon 'profile rbd' osd
    \
    'profile rbd pool=libvirt-pool'
   ```

   校验该名称是否存在。

```
cephuser@adm > ceph auth list
```

注意：用户名或 ID

`libvirt` 将使用 ID `libvirt`，而不是 Ceph 名称 `client.libvirt` 来访问 Ceph。有关 ID 与名称之间的差别的详细说明，请参见第 30.2.1.1 节 “用户”。

3. 使用 QEMU 在 RBD 池中创建映像。下面的示例使用映像名称 `new-libvirt-image` 并引用 `libvirt-pool`。

提示：密钥环文件位置

`libvirt` 用户密钥存储在 `/etc/ceph` 目录下的密钥环文件中。需要为密钥环文件指定适当的名称，其中应包含其所属 Ceph 集群的名称。如果集群名称为默认名称“ceph”，则密钥环文件名为 `/etc/ceph/ceph.client.libvirt.keyring`。

如果该密钥环不存在，请使用以下命令创建它：

```
cephuser@adm > ceph auth get client.libvirt > /etc/ceph/
ceph.client.libvirt.keyring
```

```
# qemu-img create -f raw rbd:libvirt-pool/new-libvirt-image:id=libvirt 2G
```

校验该映像是否存在。

```
cephuser@adm > rbd -p libvirt-pool ls
```

26.2 准备 VM 管理器

虽然您可以单独使用 `libvirt`，而不借助 VM 管理器，但您可能会发现，使用 **`virt-manager`** 创建第一个域会更简单。

1. 安装虚拟机管理器。

```
# zypper in virt-manager
```

2. 准备/下载要运行虚拟化的系统的 OS 映像。
3. 启动虚拟机管理器。

```
virt-manager
```

26.3 创建 VM

要使用 **virt-manager** 创建 VM，请执行以下步骤：

1. 从列表中选择连接，右键点击该连接，然后选择新建。
2. 通过提供现有存储的路径来导入现有的磁盘映像。指定 OS 类型和内存设置，并给虚拟机命名，例如 `libvirt-virtual-machine`。
3. 完成配置并启动 VM。
4. 使用 **sudo virsh list** 校验新建的域是否存在。如果需要，请指定连接字符串，例如

```
virsh -c qemu+ssh://root@vm_host_hostname/system list
Id    Name                           State
----------------------------------------------
[...]
 9     libvirt-virtual-machine       running
```

5. 在将 VM 配置为与 Ceph 搭配使用前，登录 VM 并将其停止。

26.4 配置 VM

本章重点介绍如何使用 **virsh** 配置 VM，以与 Ceph 集成。**virsh** 命令通常需要 root 特权 (**sudo**)，它不会返回相应的结果，也不会告知您需要 root 特权。有关 **virsh** 命令的参考信息，请参见 **man 1 virsh**（需要安装 `libvirt-client` 软件包）。

1. 使用 **`virsh edit`** `vm-domain-name` 打开配置文件。

```
# virsh edit libvirt-virtual-machine
```

2. <devices> 下面应有一个 <disk> 项。

```
<devices>
    <emulator>/usr/bin/qemu-system-SYSTEM-ARCH</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw'/>
      <source file='/path/to/image/recent-linux.img'/>
      <target dev='vda' bus='virtio'/>
      <address type='drive' controller='0' bus='0' unit='0'/>
    </disk>
```

将 `/path/to/image/recent-linux.img` 替换为 OS 映像的路径。

重要

请使用 **`sudo virsh edit`**，不要使用文本编辑器。如果使用文本编辑器编辑 `/etc/qemu` 下的配置文件，`libvirt`libvirt 可能无法识别更改。如果 `/etc/libvirt/qemu` 下的 XML 文件内容与 **`sudo virsh dumpxml`** `vm-domain-name` 返回的结果有差异，则表示 VM 可能没有正常工作。

3. 将之前创建的 Ceph RBD 映像添加为 <disk> 项。

```
<disk type='network' device='disk'>
        <source protocol='rbd' name='libvirt-pool/new-libvirt-image'>
                <host name='monitor-host' port='6789'/>
        </source>
        <target dev='vda' bus='virtio'/>
</disk>
```

将 `monitor-host` 替换为主机的名称，并根据需要替换存储池名称和/或映像名称。可为 Ceph Monitor 添加多个 <host> 项。`dev` 属性是逻辑设备名称，将显示在 VM 的 `/dev` 目录下。可选的 bus 属性表示要模拟的磁盘设备类型。有效的设置都特定于驱动程序（例如 ide、scsi、virtio、xen、usb 或 sata）。

4. 保存文件。

5. 如果 Ceph 集群已启用身份验证（默认会启用），则您必须生成机密。打开所选的编辑器，并创建包含以下内容的 `secret.xml` 文件：

```
<secret ephemeral='no' private='no'>
        <usage type='ceph'>
                <name>client.libvirt secret</name>
        </usage>
</secret>
```

6. 定义机密。

```
# virsh secret-define --file secret.xml
<uuid of secret is output here>
```

7. 获取 `client.libvirt` 密钥，并将密钥字符串保存到某个文件中。

```
cephuser@adm > ceph auth get-key client.libvirt | sudo tee
 client.libvirt.key
```

8. 设置机密的 UUID。

```
# virsh secret-set-value --secret uuid of secret \
--base64 $(cat client.libvirt.key) && rm client.libvirt.key secret.xml
```

此外，必须通过将以下 `<auth>` 项添加到前面输入的 `<disk>` 元素（请将 uuid 值替换为上述命令行示例的结果），来手动设置机密。

```
# virsh edit libvirt-virtual-machine
```

然后，在域配置文件中添加 `<auth></auth>` 元素：

```
...
</source>
<auth username='libvirt'>
        <secret type='ceph' uuid='9ec59067-fdbc-a6c0-03ff-df165c0587b8'/>
</auth>
<target ...
```

注意

示例 ID 为 `libvirt`，而不是在第 26.1 节 “将 Ceph 配置为与 libvirt 搭配使用”的步骤 2 中生成的 Ceph 名称 `client.libvirt`。请务必使用所生成的 Ceph 名称的 ID 组成部分。如果出于某种原因需要重新生成机密，则在再次执行 **`sudo virsh secret-set-value`** 之前，需要执行 **`sudo virsh secret-undefine`** `uuid`。

26.5 摘要

配置要与 Ceph 搭配使用的 VM 之后，便可启动该 VM。要校验 VM 与 Ceph 是否可相互通讯，可执行以下过程。

1. 检查 Ceph 是否在运行：

```
cephuser@adm > ceph health
```

2. 检查 VM 是否在运行：

```
# virsh list
```

3. 检查 VM 是否在与 Ceph 通讯。将 `vm-domain-name` 替换为 VM 域的名称：

```
# virsh qemu-monitor-command --hmp vm-domain-name 'info block'
```

4. 检查 `/dev` 或 `/proc/partitions` 下是否存在 `&target dev='hdb' bus='ide'/>` 中的设备：

```
> ls /dev
> cat /proc/partitions
```

27 Ceph 用作 QEMU KVM 实例的后端

最常见的 Ceph 用例涉及到向虚拟机提供块设备映像。例如，在理想的配置中，用户可以创建包含 OS 和所有相关软件的“黄金”映像。然后，用户可以创建该映像的快照。最后，用户可以克隆该快照（通常要克隆多次，有关详细信息，请参见第 20.3 节 “快照”）。能够创建快照的写入时复制克隆，就意味着 Ceph 能够快速向虚拟机供应块设备映像，因为客户端不需要在每次运转新的虚拟机时都下载整个映像。

Ceph 块设备可与 QEMU 虚拟机集成。有关 QEMU KVM 的详细信息，请参见 https://documentation.suse.com/sles/15-SP1/single-html/SLES-virtualization/#part-virt-qemu。

27.1 安装 qemu-block-rbd

要使用 Ceph 块设备，需在 QEMU 上安装相应的驱动程序。请检查是否已安装 `qemu-block-rbd` 软件包，并根据需要予以安装：

```
# zypper install qemu-block-rbd
```

27.2 使用 QEMU

使用 QEMU 命令行时，您需要指定存储池名称和映像名称。您也可以指定快照名称。

```
qemu-img command options \
rbd:pool-name/image-name@snapshot-name:option1=value1:option2=value2...
```

例如，可按如下所示指定 `id` 和 `conf` 选项：

```
qemu-img command options \
rbd:pool_name/image_name:id=glance:conf=/etc/ceph/ceph.conf
```

27.3 使用 QEMU 创建映像

可以通过 QEMU 创建块设备映像。必须指定 rbd、存储池名称，以及要创建的映像的名称。此外，必须指定映像的大小。

```
qemu-img create -f raw rbd:pool-name/image-name size
```

例如：

```
qemu-img create -f raw rbd:pool1/image1 10G
Formatting 'rbd:pool1/image1', fmt=raw size=10737418240 nocow=off cluster_size=0
```

重要

事实上，raw 数据格式是可对 RBD 使用的唯一合理格式选项。从技术上讲，您也可以使用 QEMU 支持的其他格式（例如 qcow2），但这会增加额外的开销，如果启用了缓存，还会在实时迁移虚拟机时让卷变得不安全。

27.4 使用 QEMU 调整映像大小

可以通过 QEMU 调整块设备映像的大小。必须指定 rbd、存储池名称，以及要调整大小的映像的名称。此外，必须指定映像的大小。

```
qemu-img resize rbd:pool-name/image-name size
```

例如：

```
qemu-img resize rbd:pool1/image1 9G
Image resized.
```

27.5 使用 QEMU 检索映像信息

可以通过 QEMU 检索块设备映像的信息。必须指定 rbd、存储池名称和映像名称。

```
qemu-img info rbd:pool-name/image-name
```

例如：

```
qemu-img info rbd:pool1/image1
image: rbd:pool1/image1
file format: raw
virtual size: 9.0G (9663676416 bytes)
disk size: unavailable
cluster_size: 4194304
```

27.6 使用 RBD 运行 QEMU

QEMU 可以通过 `librbd` 直接将映像作为虚拟块设备来访问。这可以避免额外的环境切换，并可利用 RBD 缓存的优势。

您可以使用 **`qemu-img`** 将现有的虚拟机映像转换成 Ceph 块设备映像。例如，如果您有一个 qcow2 映像，则可以运行：

```
qemu-img convert -f qcow2 -O raw sles12.qcow2 rbd:pool1/sles12
```

要运行从该映像引导的虚拟机，可以运行：

```
# qemu -m 1024 -drive format=raw,file=rbd:pool1/sles12
```

RBD 缓存可大幅提高性能。QEMU 的缓存选项可控制 `librbd` 缓存：

```
# qemu -m 1024 -drive format=rbd,file=rbd:pool1/sles12,cache=writeback
```

有关 RBD 缓存的详细信息，请参见第 20.5 节 “缓存设置”。

27.7 启用丢弃和 TRIM

Ceph 块设备支持丢弃操作。这意味着，guest 可以发送 TRIM 请求，让 Ceph 块设备回收未使用的空间。可以通过结合 discard 选项挂载 XFS，在 guest 中启用此功能。

要让 guest 可使用此功能，必须为块设备显式启用此功能。为此，您必须指定与驱动器关联的 `discard_granularity`：

```
# qemu -m 1024 -drive format=raw,file=rbd:pool1/sles12,id=drive1,if=none \
-device driver=ide-hd,drive=drive1,discard_granularity=512
```

注意

上面的示例使用 IDE 驱动程序。Virtio 驱动程序不支持丢弃功能。

如果使用 `libvirt`，请使用 **`virsh edit`** 编辑 libvirt 域的配置文件，以包含 `xmlns:qemu` 值。然后，将 `qemu:commandline block` 添加为该域的子级。下面的示例说明如何将包含 `qemu id=` 的两个设备设置为不同的 `discard_granularity` 值。

```
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
 <qemu:commandline>
  <qemu:arg value='-set'/>
  <qemu:arg value='block.scsi0-0-0.discard_granularity=4096'/>
  <qemu:arg value='-set'/>
  <qemu:arg value='block.scsi0-0-1.discard_granularity=65536'/>
 </qemu:commandline>
</domain>
```

27.8 设置 QEMU 缓存选项

QEMU 的缓存选项与以下 Ceph RBD 缓存设置对应。

写回:

```
rbd_cache = true
```

直写：

```
rbd_cache = true
rbd_cache_max_dirty = 0
```

无:

```
rbd_cache = false
```

QEMU 的缓存设置会覆盖 Ceph 的默认设置（未在 Ceph 配置文件中显式指定的设置）。如果在 Ceph 配置文件中明确指定了 RBD 缓存设置（请参见第 20.5 节 “缓存设置”），您的 Ceph 设置将会覆盖 QEMU 缓存设置。如果在 QEMU 命令行中指定了缓存设置，则 QEMU 命令行设置会覆盖 Ceph 配置文件设置。

VI 配置集群

28 Ceph 集群配置 368

29 Ceph Manager 模块 387

30 cephx 身份验证 392

28 Ceph 集群配置

本章介绍如何通过配置选项配置 Ceph 集群。

28.1 配置 ceph.conf 文件

cephadm 使用基本 `ceph.conf` 文件，该文件中仅包含用于连接到 MON、身份验证和提取配置信息的最小选项集。在大多数情况下，这仅限于 `mon_host` 选项（虽然可以通过使用 DNS SRV 记录来避免该限制）。

重要

`ceph.conf` 文件不再充当存储集群配置的中心位置，而是支持配置数据库（参见第 28.2 节 “配置数据库”）。

如果您仍然需要通过 `ceph.conf` 文件更改集群配置（例如，由于您使用的客户端不支持从配置数据库读取选项），则需要运行以下命令，并负责在整个集群维护和分发 `ceph.conf` 文件：

```
cephuser@adm > ceph config set mgr mgr/cephadm/manage_etc_ceph_ceph_conf
 false
```

28.1.1 访问容器映像内部的 ceph.conf

虽然 Ceph 守护进程在容器内运行，您仍然可以访问其 `ceph.conf` 配置文件。它作为主机系统上的以下文件**绑定挂载**：

```
/var/lib/ceph/CLUSTER_FSID/DAEMON_NAME/config
```

将 `CLUSTER_FSID` 替换为 **`ceph fsid`** 命令返回的运行中集群的唯一 FSID，并将 `DAEMON_NAME` 替换为 **`ceph orch ps`** 命令列出的特定守护进程的名称。例如：

```
/var/lib/ceph/b4b30c6e-9681-11ea-ac39-525400d7702d/osd.2/config
```

要修改某个守护进程的配置，请编辑其 `config` 文件并重启动该守护进程：

```
# systemctl restart ceph-CLUSTER_FSID-DAEMON_NAME
```

例如：

```
# systemctl restart ceph-b4b30c6e-9681-11ea-ac39-525400d7702d-osd.2
```

重要

cephadm 重新部署守护进程后，所有自定义设置都将丢失。

28.2 配置数据库

Ceph Monitor 负责管理影响整个集群行为的配置选项的中央数据库。

28.2.1 配置部分和掩码

MON 存储的配置选项可以位于**全局**部分、**守护进程类型**部分或**特定守护进程**部分中。此外，选项还可以具有与其关联的**掩码**，以进一步限制选项所适用的守护进程或客户端范围。掩码具有以下两种格式：

- `TYPE:LOCATION`，其中 `TYPE` 为 `rack` 或 `host` 等 CRUSH 属性，而 `LOCATION` 为该属性的值。
  例如，`host:example_host` 会将该选项仅限用于在某个特定主机上运行的守护进程或客户端。

- `CLASS:DEVICE_CLASS`，其中 `DEVICE_CLASS` 为 `hdd` 或 `ssd` 等 CRUSH 设备类型的名称。例如，`class:ssd` 会将该选项仅限用于由 SSD 支持的 OSD。此掩码对非 OSD 守护进程或客户端无效。

28.2.2 设置和读取配置选项

使用以下命令可设置或读取集群配置选项。`WHO` 参数可以是部分名称、掩码或两者的组合，用斜杠 (/) 字符分隔。例如，`osd/rack:foo` 表示机架中名为 `foo` 的所有 OSD 守护进程。

`ceph config dump`

转储整个集群的整个配置数据库。

`ceph config get WHO`

转储存储在配置数据库中的特定守护进程或客户端（例如，`mds.a`）的配置。

`ceph config set WHO OPTION VALUE`

将配置选项设置为配置数据库中的指定值。

`ceph config show WHO`

显示某个运行中守护进程的已报告运行中配置。如果还使用了本地配置文件，或者在命令行上或运行时覆盖了选项，则这些设置可能与 Monitor 所存储的设置有所不同。选项值的来源会作为输出的一部分进行报告。

`ceph config assimilate-conf -i INPUT_FILE -o OUTPUT_FILE`

导入指定为 `INPUT_FILE` 的配置文件，并将任何有效选项存储到配置数据库中。任何无法识别、无效或 Monitor 无法控制的设置都将在存储为 `OUTPUT_FILE` 的缩写文件中返回。此命令对于从旧配置文件转换为基于 Monitor 的集中式配置非常有用。

28.2.3 在运行时配置守护进程

在大多数情况下，Ceph 允许您在运行时更改守护进程的配置。例如，当您需要增加或减少日志记录输出量时，或者在执行运行时集群优化时，这非常有用。

您可以使用以下命令更新配置选项的值：

```
cephuser@adm > ceph config set DAEMON OPTION VALUE
```

例如，要调整某个特定 OSD 上的调试日志级别，请运行：

```
cephuser@adm > ceph config set osd.123 debug_ms 20
```

注意

如果在本地配置文件中还自定义了相同的选项，则将忽略 Monitor 设置，因为它的优先级低于配置文件。

28.2.3.1 覆盖值

您可以使用 **tell** 或 **daemon** 子命令暂时修改选项值。此类修改只影响运行中进程，并会在守护进程或进程重启动后被丢弃。

有两种方法可以覆盖值：

- 使用 **tell** 子命令从任何集群节点向某个特定守护进程发送消息：

```
cephuser@adm > ceph tell DAEMON config set OPTION VALUE
```

例如：

```
cephuser@adm > ceph tell osd.123 config set debug_osd 20
```

提示

tell 子命令接受使用通配符作为守护进程标识符。例如，要调整所有 OSD 守护进程上的调试级别，请运行：

```
cephuser@adm > ceph tell osd.* config set debug_osd 20
```

- 使用 **daemon** 子命令通过 `/var/run/ceph` 中的套接字从运行进程的节点连接到某个特定守护进程：

```
cephuser@adm > cephadm enter --name osd.ID -- ceph daemon DAEMON config
 set OPTION VALUE
```

例如：

```
cephuser@adm > cephadm enter --name osd.4 -- ceph daemon osd.4 config set
 debug_osd 20
```

提示

使用 **`ceph config show`** 命令查看运行时设置（参见第 28.2.3.2 节 “查看运行时设置”）时，显示的暂时覆盖值将带有来源 `override`。

28.2.3.2 查看运行时设置

查看某个守护进程的所有选项设置：

```
cephuser@adm > ceph config show-with-defaults osd.0
```

查看某个守护进程的的所有非默认选项设置：

```
cephuser@adm > ceph config show osd.0
```

检查某个特定选项：

```
cephuser@adm > ceph config show osd.0 debug_osd
```

您还可以从运行其进程的节点连接到某个运行中守护进程，并查看其配置：

```
cephuser@adm > cephadm enter --name osd.0 -- ceph daemon osd.0 config show
```

仅查看非默认设置：

```
cephuser@adm > cephadm enter --name osd.0 -- ceph daemon osd.0 config diff
```

检查某个特定选项：

```
cephuser@adm > cephadm enter --name osd.0 -- ceph daemon osd.0 config get
 debug_osd
```

28.3 config-key 储存

config-key 是 Ceph Monitor 提供的一项通用服务。它通过永久存储键值对简化了配置键的管理。config-key 主要由 Ceph 工具和守护进程使用。

提示

添加新键或修改现有键后，重启动受影响的服务，以便使更改生效。有关操作 Ceph 服务的更多详细信息，请参见第 14 章 “Ceph 服务的操作”。

使用 命令可操作 config-keyconfig-key 存储。**config-key** 命令使用以下子命令：

ceph config-key rm KEY

删除指定键。

ceph config-key exists KEY

检查指定键是否存在。

ceph config-key get KEY

检索指定键的值。

ceph config-key ls

列出所有键。

ceph config-key dump

转储所有键及其值。

ceph config-key set KEY VALUE

存储具有给定值的指定键。

28.3.1 iSCSI 网关

iSCSI 网关使用 config-key 存储来保存或读取其配置选项。所有与 iSCSI 网关相关的键都以 iscsi 字符串作为前缀，例如：

```
iscsi/trusted_ip_list
```

```
iscsi/api_port
iscsi/api_user
iscsi/api_password
iscsi/api_secure
```

例如，如果需要两组配置选项，请使用另一个描述性关键字（例如 `datacenterA` 和 `datacenterB`）扩展前缀：

```
iscsi/datacenterA/trusted_ip_list
iscsi/datacenterA/api_port
[...]
iscsi/datacenterB/trusted_ip_list
iscsi/datacenterB/api_port
[...]
```

28.4 Ceph OSD 和 BlueStore

28.4.1 配置自动缓存大小调整

如果 `tc_malloc` 配置为内存分配器并且启用了 `bluestore_cache_autotune` 设置，可将 BlueStore 配置为自动调整其缓存大小。当前，此选项默认处于启用状态。BlueStore 会尝试将 OSD 堆内存使用率维持在 `osd_memory_target` 配置选项指定的目标大小之下。这是一种最优算法，而且缓存不会缩小到小于 `osd_memory_cache_min` 指定的大小。系统将根据优先级的层次选择缓存率。如果无法获取优先级信息，将使用 `bluestore_cache_meta_ratio` 和 `bluestore_cache_kv_ratio` 选项作为备选。

bluestore_cache_autotune

在遵循最小值的要求下，自动调整为不同 BlueStore 缓存指定的比率。默认值是 `true`。

osd_memory_target

如果启用了 `tc_malloc` 和 `bluestore_cache_autotune`，该选项将尝试在内存中映射此数量的字节。

注意

此数量可能与进程的 RSS 内存使用率并不完全匹配。虽然进程映射的堆内存总量一般应该保持为接近此目标的数量，但并不能保证内核会真正回收已取消映射的内存。

osd_memory_cache_min

如果启用了 `tc_malloc` 和 `bluestore_cache_autotune`，该选项可设置用于缓存的最小内存量。

注意

将此值设置得过小可能会导致严重的缓存抖动。

28.5 Ceph 对象网关

您可以通过多个选项来影响对象网关行为。如果未指定选项，将使用默认值。完整的对象网关选项列表如下：

28.5.1 一般设置

rgw_frontends

配置 HTTP 前端。请以逗号分隔的列表指定多个前端。每个前端配置可能包含一组以空格分隔的选项，每个选项的格式均为“键=值”或“键”。默认值为 `beast port=7480`。

rgw_data

为对象网关设置数据文件的位置。默认值为 `/var/lib/ceph/radosgw/CLUSTER_ID`。

rgw_enable_apis

启用指定的 API。默认值为“s3, swift, swift_auth, admin All APIs”。

rgw_cache_enabled

启用或禁用对象网关缓存。默认值是 `true`。

rgw_cache_lru_size

对象网关缓存中的条目数。默认值为 10000。

rgw_socket_path

域套接字的套接字路径。`FastCgiExternalServer` 使用此套接字。如果您未指定套接字路径，对象网关将不会作为外部服务器运行。您在此处指定的路径须与 `rgw.conf` 文件中指定的路径相同。

rgw_fcgi_socket_backlog

fcgi 的套接字积压数据。默认值为 1024。

rgw_host

对象网关实例的主机，可以是 IP 地址或主机名。默认值为 `0.0.0.0`。

rgw_port

实例用于侦听请求的端口号。如果未指定，对象网关将运行外部 FastCGI。

rgw_dns_name

所处理的域的 DNS 名称。

rgw_script_uri

如果请求中未设置 SCRIPT_URI，则作为其备选值。

rgw_request_uri

如果请求中未设置 REQUEST_URI，则作为其备选值。

rgw_print_continue

启用 100-continue（如果可正常工作）。默认值是 `true`。

rgw_remote_addr_param

远程地址参数。例如，包含远程地址的 HTTP 字段或 X-Forwarded-For 地址（如果反向代理可正常工作）。默认值为 `REMOTE_ADDR`。

rgw_op_thread_timeout

活动线程的超时（以秒为单位）。默认值为 600。

rgw_op_thread_suicide_timeout

对象网关进程停止前超时的时间（以秒为单位）。如果设置为 0（默认值），则表示禁用。

rgw_thread_pool_size

Beast 服务器的线程数。如果需要处理更多请求，请设置更高的值。默认为 100 个线程。

rgw_num_rados_handles

对象网关的 RADOS 集群句柄数。现在，每个对象网关工作线程都可以选择一个 RADOS 句柄并在其有效期内予以保留。未来版本中可能会弃用并移除此选项。默认值为 1。

rgw_num_control_oids

不同对象网关实例之间用于缓存同步的通知对象数量。默认值为 8。

rgw_init_timeout

对象网关放弃初始化前经过的秒数。默认值为 30。

rgw_mime_types_file

MIME 类型的路径和位置。用于对象类型的 Swift 自动检测。默认值为 `/etc/mime.types`。

rgw_gc_max_objs

垃圾回收在一个垃圾回收处理周期内可处理的最大对象数量。默认值为 32。

rgw_gc_obj_min_wait

垃圾回收处理可删除和处理对象前需等待的最短时间。默认值为 2*3600。

rgw_gc_processor_max_time

两次连续的垃圾回收处理周期开始时所间隔的最长时间。默认值为 3600。

rgw_gc_processor_period

垃圾回收处理的周期时间。默认值为 3600。

rgw_s3_success_create_obj_status

`create-obj` 的备选成功状态响应。默认值为 0。

rgw_resolve_cname

对象网关是否应使用请求主机名字段的 DNS CNAME 记录（如果主机名与对象网关 DNS 名称不同）。默认为 `false`。

rgw_obj_stripe_size

对象网关对象的对象条带大小。默认值为 `4 << 20`。

rgw_extended_http_attrs

添加可对实体（例如，用户、存储桶或对象）设置的一组新属性。使用 POST 方法放置或修改实体时，可通过 HTTP 报头字段设置这些额外的属性。如果设置了这些属性，当针对实体请求 GET/HEAD 时，它们将作为 HTTP 字段返回。默认值为 `content_foo, content_bar, x-foo-bar`。

rgw_exit_timeout_secs

无条件退出前等待进程的秒数。默认值为 120。

rgw_get_obj_window_size

单个对象请求的窗口大小（以字节为单位）。默认值为 `16 << 20`。

rgw_get_obj_max_req_size

发送到 Ceph 存储集群的单次 GET 操作的请求大小上限。默认值为 `4 << 20`。

rgw_relaxed_s3_bucket_names

为 US 区域存储桶启用宽松 S3 存储桶名称规则。默认为 `false`。

rgw_list_buckets_max_chunk

列出用户存储桶时，在一次操作中检索的存储桶数量上限。默认值为 1000。

rgw_override_bucket_index_max_shards

表示存储桶索引对象的分片数量。设为 0（默认值）表示没有分片。不建议将此值设置得太大（例如 1000），因为这样会增加列出存储桶的成本。应在客户端或全局段落设置此变量，以便将其自动应用到 **`radosgw-admin`** 命令。

rgw_curl_wait_timeout_ms

某些 **`curl`** 调用的超时时间（以毫秒为单位）。默认值为 1000。

rgw_copy_obj_progress

启用在费时较长的复制操作期间输出对象进度的功能。默认值是 `true`。

rgw_copy_obj_progress_every_bytes

两次复制进度输出相隔的最小字节数。默认值为 1024*1024。

rgw_admin_entry

管理请求 URL 的入口点。默认值为 `admin`。

rgw_content_length_compat

为设置了 CONTENT_LENGTH 和 HTTP_CONTENT_LENGTH 的 FCGI 请求启用兼容性处理。默认为 `false`。

rgw_bucket_quota_ttl

已缓存配额信息的可信时长（以秒为单位）。经过此超时时间后，将从集群重新提取配额信息。默认值为 600。

rgw_user_quota_bucket_sync_interval

同步到集群前，存储桶配额信息的累计时长（以秒为单位）。在此期间，其他对象网关实例将不会看到与此实例上的操作相关的存储桶配额统计数据发生的更改。默认值为 180。

rgw_user_quota_sync_interval

同步到集群前，用户配额信息的累计时长（以秒为单位）。在此期间，其他对象网关实例将不会看到与此实例上的操作相关的用户配额统计数据发生的更改。默认值为 180。

rgw_bucket_default_quota_max_objects

每个存储桶的默认对象数量上限。如果未指定其他配额，此上限将对新用户设置，对现有用户无效。应在客户端或全局段落设置此变量，以便将其自动应用到 **`radosgw-admin`** 命令。默认值为 -1。

rgw_bucket_default_quota_max_size

每个存储桶的默认容量上限（以字节为单位）。如果未指定其他配额，此上限将对新用户设置，对现有用户无效。默认值为 -1。

rgw_user_default_quota_max_objects

用户的默认对象数量上限。这个数量包括该用户拥有的所有存储桶中的全部对象。如果未指定其他配额，此上限将对新用户设置，对现有用户无效。默认值为 -1。

rgw_user_default_quota_max_size

如果未指定其他配额，用户配额大小上限值（以字节为单位）将对新用户设置，对现有用户无效。默认值为 -1。

rgw_verify_ssl

发出请求时验证 SSL 证书。默认值是 `true`。

rgw_max_chunk_size

将在单个操作中读取的数据块的最大大小。将值增至 4MB (4194304) 可以在处理大型对象时提高性能。默认值为 128kB (131072)。

多站点设置

rgw_zone

网关实例所在区域的名称。如果未设置区域，可使用 `radosgw-admin zone default` 命令配置集群范围的默认值。

rgw_zonegroup

网关实例所在区域组的名称。如果未设置区域组，可使用 `radosgw-admin zonegroup default` 命令配置集群范围的默认值。

rgw_realm

网关实例所在领域的名称。如果未设置领域，可使用 `radosgw-admin realm default` 命令配置集群范围的默认值。

rgw_run_sync_thread

如果领域中有其他要依据其同步的区域，则生成线程来处理数据和元数据的同步。默认值是 `true`。

rgw_data_log_window

数据日志条目时段（以秒为单位）。默认值为 30。

rgw_data_log_changes_size

要为数据更改日志保存的内存内条目数。默认值为 1000。

rgw_data_log_obj_prefix

数据日志的对象名称前缀。默认值为“data_log”。

rgw_data_log_num_shards

保存数据更改日志的分片（对象）的数量。默认值为 128。

rgw_md_log_max_shards

元数据日志的最大分片数量。默认值为 64。

SWIFT 设置

rgw_enforce_swift_acls

强制实施 Swift 访问控制列表 (ACL) 设置。默认值是 `true`。

rgw_swift_token_expiration

Swift 令牌过期时间（以秒为单位）。默认值为 24*3600。

rgw_swift_url

Ceph Object Gateway Swift API 的 URL。

rgw_swift_url_prefix

Swift StorageURL 中位于“/v1”部分前的 URL 前缀。此设置允许在同一主机上运行多个网关实例。出于兼容性考虑，请将此配置变量设为空，以便使用默认值“/swift”。使用显式前缀“/”会在根目录启动 StorageURL。

警告

如果启用了 S3 API，将此选项设为“/”将不起作用。请注意，如果禁用 S3，将无法在多站点配置中部署对象网关。

rgw_swift_auth_url

未使用内部 Swift 身份验证时，用于验证 v1 身份验证令牌的默认 URL。

rgw_swift_auth_entry

Swift 身份验证 URL 的入口点。默认值为 `auth`。

rgw_swift_versioning_enabled

启用 OpenStack Object Storage API 的对象版本控制。此选项可让客户端对应进行版本控制的容器设置 `X-Versions-Location` 属性。该属性指定存储已存档版本的容器的名称。出于访问控制验证（ACL **不在**考虑范围之内）的原因，该属性必须由受版本控制的容器所属的同一用户所有。这些容器无法通过 S3 对象版本控制机制进行版本控制。默认为 `false`。

日志记录设置

rgw_log_nonexistent_bucket

允许对象网关记录针对不存在的存储桶的请求。默认为 `false`。

rgw_log_object_name

对象名称的日志记录格式。有关格式说明符的详细信息，请参见 `man 1 date` 手册页。默认值为 `%Y-%m-%d-%H-%i-%n`。

rgw_log_object_name_utc

记录的对象名称是否包含 UTC 时间。如果设置为 `false`（默认值），则会使用本地时间。

rgw_usage_max_shards

用于使用率日志记录的最大分片数量。默认值为 32。

rgw_usage_max_user_shards

用于单个用户的使用率日志记录的最大分片数量。默认值为 1。

rgw_enable_ops_log

启用记录每个成功的对象网关操作的功能。默认为 `false`。

rgw_enable_usage_log

启用使用率日志。默认为 `false`。

rgw_ops_log_rados

是否应将操作日志写入 Ceph 存储集群后端。默认值是 `true`。

rgw_ops_log_socket_path

用于写入操作日志的 Unix 域套接字。

rgw_ops_log_data_backlog

写入 Unix 域套接字的操作日志的数据积压大小上限。默认值为 5 << 20。

rgw_usage_log_flush_threshold

同步刷新前使用率日志中已合并的脏条目数量。默认值为 1024。

rgw_usage_log_tick_interval

每隔“n”秒刷新待处理使用率日志数据。默认值为 30。

rgw_log_http_headers

要包含在日志条目中的 HTTP 报头的逗号分隔列表。报头名称不区分大小写，使用完整报头名称格式，各单词间以下划线分隔。例如“http_x_forwarded_for”、“http_x_special_k”。

rgw_intent_log_object_name

要记录的日志对象名称的日志记录格式。有关格式说明符的详细信息，请参见 **man 1 date** 手册页。默认值为“%Y-%m-%d-%i-%n”。

rgw_intent_log_object_name_utc

要记录的日志对象名称是否包含 UTC 时间。如果设置为 `false`（默认值），则会使用本地时间。

KEYSTONE 设置

rgw_keystone_url

Keystone 服务器的 URL。

rgw_keystone_api_version

应用于与 Keystone 服务器通讯的 OpenStack Identity API 的版本（2 或 3）。默认值为 2。

rgw_keystone_admin_domain

使用 OpenStack Identity API v3 时具有管理员权限的 OpenStack 域的名称。

rgw_keystone_admin_project

使用 OpenStack Identity API v3 时具有管理员权限的 OpenStack 项目的名称。如果未设置，将使用 **rgw keystone admin tenant** 的值替代。

rgw_keystone_admin_token

Keystone 管理员令牌（共享密钥）。在对象网关中，使用管理员令牌的身份验证的优先级高于使用管理员身份凭证的身份验证（选项 `rgw keystone admin user`、`rgw keystone admin password`、`rgw keystone admin tenant`、`rgw keystone admin project` 和 `rgw keystone admin domain`）。管理员令牌功能被视为已弃用。

rgw_keystone_admin_tenant

使用 OpenStack Identity API v2 时具有管理员权限的 OpenStack 租户（服务租户）的名称。

rgw_keystone_admin_user

使用 OpenStack Identity API v2 时具有管理员权限的 OpenStack 用户（服务用户）的名称。

rgw_keystone_admin_password

使用 OpenStack Identity API v2 时 OpenStack 管理员用户的密码。

rgw_keystone_accepted_roles

处理请求需要具有的角色。默认值为“Member, admin”。

rgw_keystone_token_cache_size

每个 Keystone 令牌缓存中的条目数上限。默认值为 10000。

rgw_keystone_revocation_interval

令牌撤消检查间隔的时间（以秒为单位）。默认值为 15*60。

rgw_keystone_verify_ssl

向 Keystone 发出令牌请求时，验证 SSL 证书。默认值是 `true`。

28.5.1.1 补充说明

rgw_dns_name

允许客户端使用 `vhost` 样式的存储桶。

`vhost` 样式访问表示使用 `bucketname.s3-endpoint/object-path`。与之相对，`path` 样式访问使用：`s3-endpoint/bucket/object`

如果设置了 `rgw dns name`，请确认 S3 客户端是否配置为将请求定向到 `rgw dns name` 指定的端点。

28.5.2 配置 HTTP 前端

28.5.2.1 Beast

port、ssl_port

IPv4 和 IPv6 侦听端口号。您可以指定多个端口号：

```
port=80 port=8000 ssl_port=8080
```

默认值为 80。

endpoint、ssl_endpoint

侦听地址采用“地址[:端口]”格式，其中，地址为采用点分隔十进制数字格式的 IPv4 地址字符串，或是采用以方括号括住的十六进制表示格式的 IPv6 地址。指定将仅侦听 IPv6 的 IPv6 端点。`endpoint` 和 `ssl_endpoint` 的可选端口号分别默认为 80 和 443。您可以指定多个地址：

```
endpoint=[::1] endpoint=192.168.0.100:8000 ssl_endpoint=192.168.0.100:8080
```

ssl_private_key

此为可选设置，指定用于启用了 SSL 的端点的私用密钥文件路径。如果未指定，将使用 `ssl_certificate` 文件充当私用密钥。

tcp_nodelay

如果指定该选项，套接字选项将针对连接禁用 Nagle 的算法。这表示将会尽快发送数据包，而不会等到发生全缓冲或超时才发送。

“1”会对所有套接字禁用 Nagle 的算法。

“0”会使 Nagle 的算法保持启用状态（默认值）。

例 28.1： BEAST 配置示例

```
cephuser@adm > ceph config set rgw.myrealm.myzone.ses-min1.kwwazo \
 rgw_frontends beast port=8000 ssl_port=443 \
 ssl_certificate=/etc/ssl/ssl.crt \
 error_log_file=/var/log/radosgw/beast.error.log
```

28.5.2.2 CivetWeb

port

侦听端口号。对启用了 SSL 的端口添加“s”后缀（例如，“443s”）。要绑定特定 IPv4 或 IPv6 地址，请使用“地址:端口”格式。您可以通过使用“+”联接多个端点或通过提供多个选项，来指定多个端点：

```
port=127.0.0.1:8000+443s
port=8000 port=443s
```

默认值为 7480。

num_threads

Civetweb 所生成以处理传入 HTTP 连接的线程数。此设置可有效限制前端可处理的并发连接数。

默认为 `rgw_thread_pool_size` 选项所指定的值。

request_timeout_ms

Civetweb 在放弃前需等待更多传入数据的时长（以毫秒为单位）。

默认值为 30000 毫秒。

access_log_file

访问日志文件的路径。您可以指定完整路径，或当前工作目录的相对路径。如果未指定（默认设置），将不会记录访问。

error_log_ file

错误日志文件的路径。您可以指定完整路径，或当前工作目录的相对路径。如果未指定（默认设置），将不会记录错误。

例 28.2： `/etc/ceph/ceph.conf` 中的示例 CIVETWEB 配置

```
cephuser@adm > ceph config set rgw.myrealm.myzone.ses-min2.ingabw \
 rgw_frontends civetweb port=8000+443s request_timeout_ms=30000 \
 error_log_file=/var/log/radosgw/civetweb.error.log
```

28.5.2.3 通用选项

ssl_certificate

用于启用了 SSL 的端点的 SSL 证书文件路径。

prefix

要插入到所有请求的 URI 中的前缀字符串。例如，仅支持 Swift 的前端可能会提供一个 `/swift` 的 URI 前缀。

29 Ceph Manager 模块

Ceph Manager 的体系结构（请参见《部署指南》, 第 1 章 “SES 和 Ceph”, 第 1.2.3 节 “Ceph 节点和守护进程”中的简短介绍）允许您通过“仪表盘”（请参见第 I 部分 “Ceph Dashboard”）、“prometheus”（请参见第 16 章 “监控和告警”）或“平衡器”等**扩展模块**来扩展它的功能。

要列出所有可用的扩展模块，请运行：

```
cephuser@adm > ceph mgr module ls
{
        "enabled_modules": [
                "restful",
                "status"
        ],
        "disabled_modules": [
                "dashboard"
        ]
}
```

要启用或禁用特定扩展模块，请运行：

```
cephuser@adm > ceph mgr module enable MODULE-NAME
```

例如：

```
cephuser@adm > ceph mgr module disable dashboard
```

要列出已启用扩展模块所提供的服务，请运行：

```
cephuser@adm > ceph mgr services
{
        "dashboard": "http://myserver.com:7789/",
        "restful": "https://myserver.com:8789/"
}
```

29.1 平衡器

平衡器扩展模块可优化各 OSD 之间的归置组 (PG) 分布，以实现更平衡的部署。尽管默认会激活该扩展模块，但它并不会处于工作状态。它支持 `crush-compat` 和 `upmap` 这两种模式。

提示：当前的平衡器状态和配置

要查看当前的平衡器状态和配置信息，请运行：

```
cephuser@adm > ceph balancer status
```

29.1.1 “crush-compat”模式

在“crush-compat”模式下，平衡器会调整 OSD 的重设权重集，从而改进数据分布。它会将 PG 在 OSD 之间移动，因此会暂时出现由 PG 错放所导致的 `HEALTH_WARN` 集群状态。

提示：模式激活

尽管“crush-compat”为默认模式，但我们仍建议明确将其激活：

```
cephuser@adm > ceph balancer mode crush-compat
```

29.1.2 数据平衡的规划和执行

您可以使用平衡器扩展模块创建数据平衡计划。随后，您可以手动执行该计划，或让平衡器持续平衡 PG。

要决定是以手动还是自动模式来运行平衡器，需要考虑诸如当前数据的不平衡情况、集群大小、PG 个数或 I/O 活动等多项因素。建议您选择在集群中的 I/O 负载较低时创建和执行初始计划。原因是初始的不平衡情况可能会非常严重，因此最好将对客户端的影响保持在较低程度。初始的手动运行完成后，建议您激活自动模式并监控正常 I/O 负载下的重新平衡流量。需要对照平衡器所导致的重新平衡流量来权衡 PG 分布的改进。

提示：归置组 (PG) 的可移动部分

在平衡过程中，平衡器扩展模块会限制 PG 的移动，以便仅移动可配置的 PG 部分。该部分默认为 5%，您可通过运行以下命令调整该部分（例如，调整为 9%）：

```
cephuser@adm > ceph config set mgr target_max_misplaced_ratio .09
```

要创建和执行平衡计划，请执行以下步骤：

1. 查看当前集群分数：

   ```
   cephuser@adm > ceph balancer eval
   ```

2. 创建计划，例如“great_plan”：

   ```
   cephuser@adm > ceph balancer optimize great_plan
   ```

3. 查看“great_plan”将引发的更改：

   ```
   cephuser@adm > ceph balancer show great_plan
   ```

4. 查看潜在的集群分数（如果您决定应用“great_plan”）：

   ```
   cephuser@adm > ceph balancer eval great_plan
   ```

5. 执行“great_plan”一次：

   ```
   cephuser@adm > ceph balancer execute great_plan
   ```

6. 使用 **ceph -s** 命令观察集群平衡情况。如果您对结果满意，请激活自动平衡：

   ```
   cephuser@adm > ceph balancer on
   ```

   如果您稍后决定停用自动平衡，请运行：

   ```
   cephuser@adm > ceph balancer off
   ```

提示：在不执行初始计划的情况下激活自动平衡

您可以不执行初始计划，直接激活自动平衡。在此情况下，归置组的重新平衡用时可能会非常长。

29.2 启用遥测模块

遥测插件可发送有关运行它的集群的 Ceph 项目匿名数据。

此（选择性接受）组件包含有关集群部署方式、Ceph 版本、主机分布以及其他有助于项目更好地了解 Ceph 使用方式的参数的计数器和统计数据。它不包含任何诸如存储池名称、对象名称、对象内容或主机名等敏感数据。

遥测扩展模块用于为开发人员提供自动化反馈环，以帮助其量化采用率、跟踪情况，或指出为避免出现不良后果而需在配置期间以更佳方式进行解释或验证的内容。

注意

遥测扩展模块需要 Ceph Manager 节点能够通过 HTTPS 将数据推送到上游服务器。请确保您的公司防火墙允许此操作。

1. 要启用遥测扩展模块，请运行以下命令：

```
cephuser@adm > ceph mgr module enable telemetry
```

   ### 注意

   此命令只能让您在本地查看您的数据。它并不会将您的数据共享到 Ceph 社区。

2. 要允许遥测扩展模块开始共享数据，请运行以下命令：

```
cephuser@adm > ceph telemetry on
```

3. 要禁用遥测数据共享，请运行以下命令：

```
cephuser@adm > ceph telemetry off
```

4. 要生成可打印的 JSON 报告，请运行以下命令：

```
cephuser@adm > ceph telemetry show
```

5. 要向报告添加联系信息和描述，请运行以下命令：

```
cephuser@adm > ceph config set mgr mgr/telemetry/contact John Doe
 john.doe@example.com
cephuser@adm > ceph config set mgr mgr/telemetry/description 'My first Ceph
 cluster'
```

6. 该扩展模块默认每 24 小时编译并发送一次新报告。要调整此间隔，请运行以下命令：

```
cephuser@adm > ceph config set mgr mgr/telemetry/interval HOURS
```

30 cephx 身份验证

为了识别客户端并防范中间人攻击，Ceph 提供了 cephx 身份验证系统。在此环境中，**客户端**表示人类用户（例如 admin 用户）或 Ceph 相关的服务/守护进程（例如 OSD、Monitor 或对象网关）。

注意

cephx 协议不会处理 TLS/SSL 之类的传输中数据加密。

30.1 身份验证体系结构

cephx 使用共享秘密密钥进行身份验证，这意味着客户端和 Ceph Monitor 都有客户端秘密密钥的副本。身份验证协议可让双方互相证明各自持有密钥的副本，且无需真正透露密钥。这样就实现了相互身份验证，即，集群可确保用户拥有秘密密钥，而用户亦可确保集群持有秘密密钥的副本。

Ceph 的一项重要可伸缩性功能就是可免于通过集中式界面与 Ceph 对象存储交互。这意味着，Ceph 客户端可直接与 OSD 交互。为了保护数据，Ceph 提供了 cephx 身份验证系统来对 Ceph 客户端进行身份验证。

每个 Monitor 都可对客户端进行身份验证并分发密钥，因此，在使用 cephx 时，不会出现单一故障点或瓶颈。Monitor 会返回身份验证数据结构，其中包含获取 Ceph 服务时要用到的会话密钥。此会话密钥本身已使用客户端的永久秘密密钥进行了加密，因此，只有客户端才能向 Ceph Monitor 请求服务。然后，客户端使用会话密钥向 Monitor 请求所需的服务，Monitor 会为客户端提供一个票据，用于向实际处理数据的 OSD 验证客户端身份。Ceph Monitor 和 OSD 共享一个机密，因此，客户端使用 Monitor 提供的票据向集群中的任何 OSD 或元数据服务器表明身份。cephx 票据有失效时间，因此，攻击者无法使用已失效的票据或以不当方式获取的会话密钥。

要使用 cephx，管理员必须先设置客户端/用户。在下图中，`client.admin` 用户从命令行调用 **`ceph auth get-or-create-key`** 来生成用户名和秘密密钥。Ceph 的 **`auth`** 子系统会生成该用户名和密钥，在 Monitor 中存储一个副本，并将该用户的机密传回给 `client.admin` 用户。这意味着，客户端和 Monitor 共享一个秘密密钥。

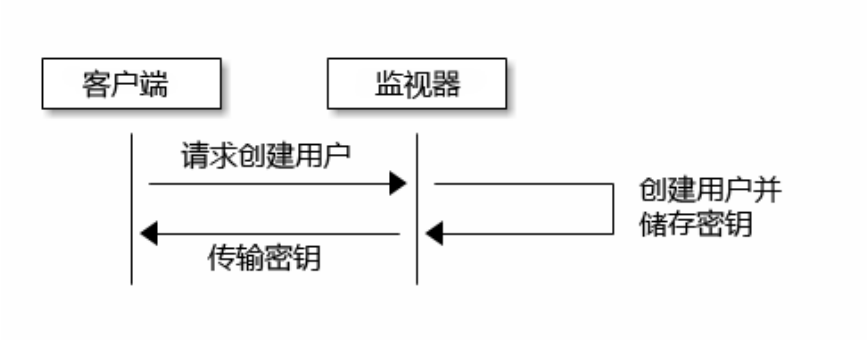


图 30.1： 基本的 cephx 身份验证

要在 Monitor 中进行身份验证，客户端需将用户名传递给 Monitor。Monitor 会生成一个会话密钥，并使用与该用户名关联的秘密密钥来加密该会话密钥，然后将加密的票据传回给客户端。之后，客户端会使用共享的秘密密钥解密数据，以获取会话密钥。会话密钥可识别当前会话的用户。然后，客户端请求与该用户相关、由会话密钥签名的票据。Monitor 会生成一个票据，使用用户的秘密密钥进行加密，然后将其传回给客户端。客户端解密该票据，并使用它对发往整个集群中的 OSD 和元数据服务器的请求进行签名。

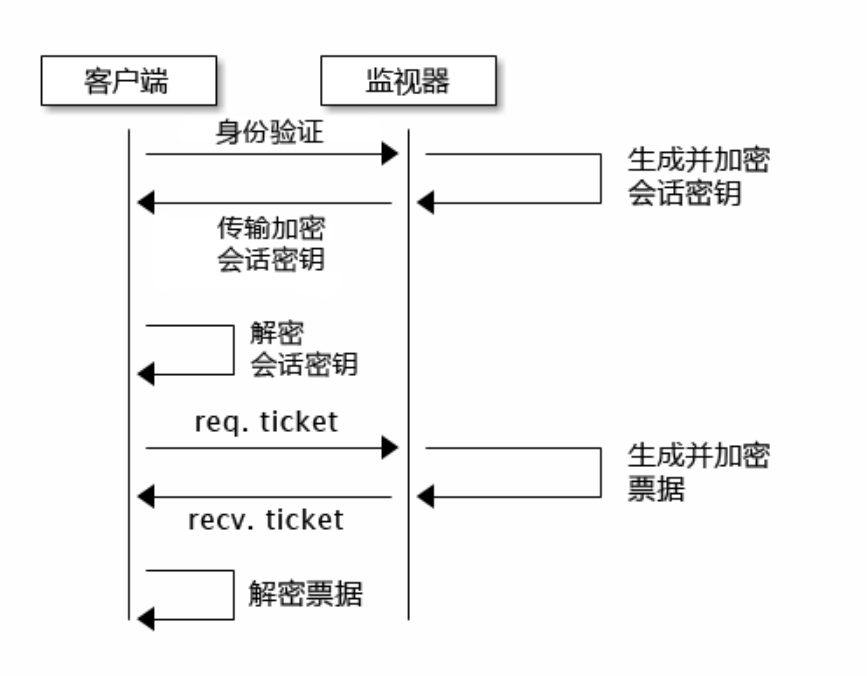


图 30.2： cephx AUTHENTICATION

cephx 协议会对客户端计算机与 Ceph 服务器之间进行的通讯进行身份验证。完成初始身份验证后，将使用 Monitor、OSD 和元数据服务器可通过共享密钥进行校验的票据，来对客户端与服务器之间发送的每条消息进行签名。

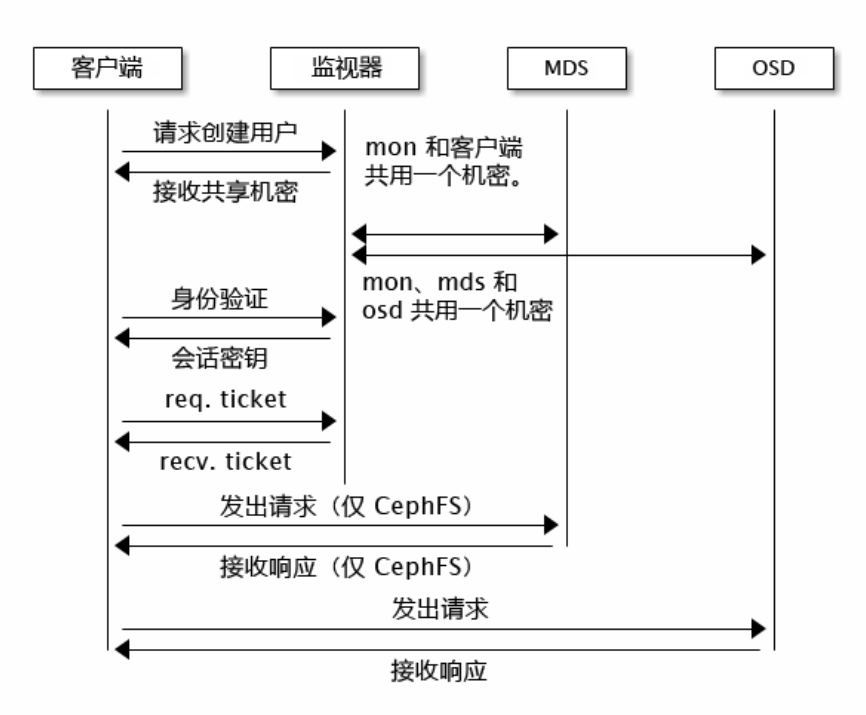


图 30.3： cephx 身份验证 - MDS 和 OSD

重要

这种身份验证提供的保护仅限于 Ceph 客户端与 Ceph 集群主机之间，身份验证不会扩展到 Ceph 客户端以外。如果用户从远程主机访问 Ceph 客户端，则不会对用户主机与客户端主机之间的连接应用 Ceph 身份验证。

30.2 主要管理

本节介绍 Ceph 客户端用户，以及如何在 Ceph 存储集群中对其进行身份验证和授权。**用户**是指使用 Ceph 客户端来与 Ceph 存储集群守护进程交互的个人或系统参与者（例如应用）。

Ceph 在启用身份验证和授权（默认启用）的情况下运行时，您必须指定一个用户名，以及包含所指定用户的秘密密钥的密钥环（通常通过命令行指定）。如果您未指定用户名，Ceph 将使用 `client.admin` 作为默认用户名。如果您未指定密钥环，Ceph 将通过 Ceph 配置文件中的密钥环设置来查找密钥环。例如，如果您在未指定用户名或密钥环的情况下执行 **`ceph health`** 命令，Ceph 将按如下所示解释该命令：

```
cephuser@adm > ceph -n client.admin --keyring=/etc/ceph/
ceph.client.admin.keyring health
```

或者，您可以使用 `CEPH_ARGS` 环境变量来避免重复输入用户名和机密。

30.2.1 背景信息

无论 Ceph 客户端是何类型（例如，块设备、对象存储、文件系统、本机 API），Ceph 都会在**存储池**中将所有数据存储为对象。Ceph 用户需要拥有存储池访问权限才能读取和写入数据。此外，Ceph 用户必须拥有执行权限才能使用 Ceph 的管理命令。以下概念可帮助您理解 Ceph 用户管理。

30.2.1.1 用户

用户是指个人或系统参与者（例如应用）。通过创建用户，可以控制谁（或哪个参与者）能够访问您的 Ceph 存储集群、其存储池及存储池中的数据。

Ceph 使用多种**类型**的用户。进行用户管理时，将始终使用 `client` 类型。Ceph 通过句点 (.) 分隔格式来标识用户，该格式由用户类型和用户 ID 组成。例如，`TYPE.ID`、`client.admin` 或 `client.user1`。区分用户类型的原因在于，Ceph Monitor、OSD 和元数据服务器也使用 cephx 协议，但它们并非客户端。区分用户类型有助于将客户端用户与其他用户区分开来，从而简化访问控制、用户监视和可追溯性。

有时，Ceph 的用户类型可能会令人混淆，因为根据命令行用法的不同，Ceph 命令行允许您指定具有类型的用户或不具有类型的用户。如果您指定 `--user` 或 `--id`，则可以省略类型。因此，可将 `client.user1` 简单地输入为 `user1`。如果您指定的是 `--name` 或 `-n`，则必须指定类型和名称，例如 `client.user1`。作为最佳实践，我们建议尽可能使用类型和名称。

注意

Ceph 存储集群用户与 Ceph 对象存储用户或 Ceph 文件系统用户均不同。Ceph 对象网关使用 Ceph 存储集群用户在网关守护进程与存储集群之间通讯，但网关有自己的用户管理功能来管理最终用户。Ceph 文件系统使用 POSIX 语义。与它关联的用户空间与 Ceph 存储集群用户不同。

30.2.1.2 授权和用户权限

Ceph 使用“用户权限”(caps) 术语来描述对已经过身份验证的用户授权，允许其运用 Monitor、OSD 和元数据服务器的功能。用户权限还可限制对存储池或存储池名称空间内的数据的访问。Ceph 管理用户可在创建或更新用户时设置用户的权限。

用户权限语法的格式如下：

```
daemon-type 'allow capability' [...]
```

下面是每个服务类型的用户权限列表：

Monitor 用户权限

包括 `r`、`w`、`x` 和 `allow profile cap`。

```
mon 'allow rwx'
mon 'allow profile osd'
```

OSD 用户权限

包括 `r`、`w`、`x`、`class-read`、`class-write` 和 `profile osd`。此外，OSD 用户权限还允许进行存储池和名称空间设置。

```
osd 'allow capability' [pool=poolname] [namespace=namespace-name]
```

MDS 用户权限

只需要 `allow`，或留空。

```
mds 'allow'
```

以下各项描述了每项用户权限：

allow

需先于守护进程的访问设置指定。仅对 MDS 表示 `rw`。

r

向用户授予读取权限。访问 Monitor 以检索 CRUSH 索引时需具有此用户权限。

w

向用户授予针对对象的写入权限。

x

授予用户调用类方法（包括读取和写入）的能力，以及在 Monitor 中执行 `auth` 操作的能力。

class-read

授予用户调用类读取方法的能力。`x` 的子集。

class-write

授予用户调用类写入方法的能力。`x` 的子集。

*****

授予用户对特定守护进程/存储池的读取、写入和执行权限，以及执行管理命令的能力。

profile osd

授予用户以某个 OSD 身份连接到其他 OSD 或 Monitor 的权限。授予 OSD 权限，使 OSD 能够处理复制检测信号流量和状态报告。

profile mds

授予用户以某个 MDS 身份连接到其他 MDS 或 Monitor 的权限。

profile bootstrap-osd

授予用户引导 OSD 的权限。授权给部署工具，使其在引导 OSD 时有权添加密钥。

profile bootstrap-mds

授予用户引导元数据服务器的权限。授权给部署工具，使其在引导元数据服务器时有权添加密钥。

30.2.1.3 存储池

存储池是指用户在其中存储数据的逻辑分区。在 Ceph 部署中，常见的做法是为相似类型的数据创建一个存储池作为逻辑分区。例如，将 Ceph 部署为 OpenStack 的后端时，典型的部署方式是为卷、映像、备份和虚拟机以及用户（如 `client.glance` 或 `client.cinder`）创建相应的存储池。

30.2.2 管理用户

用户管理功能可让 Ceph 集群管理员能够直接在 Ceph 集群中创建、更新和删除用户。

在 Ceph 集群中创建或删除用户时，可能需要将密钥分发到客户端，以便将密钥添加到密钥环。有关详细信息，请参见第 30.2.3 节 “管理密钥环”。

30.2.2.1 列出用户

要列出集群中的用户，请执行以下命令：

```
cephuser@adm > ceph auth list
```

Ceph 将列出您集群中的所有用户。例如，在包含两个节点的集群中，**`ceph auth list`** 输出类似下方所示：

```
installed auth entries:

osd.0
        key: AQCvCbtToC6MDhAATtuT70Sl+DymPCfDSsyV4w==
        caps: [mon] allow profile osd
        caps: [osd] allow *
osd.1
        key: AQC4CbtTCFJBChAAVq5spj0ff4eHZICxIOVZeA==
        caps: [mon] allow profile osd
        caps: [osd] allow *
client.admin
        key: AQBHCbtT6APDHhAA5W00cBchwkQjh3dkKsyPjw==
        caps: [mds] allow
        caps: [mon] allow *
        caps: [osd] allow *
client.bootstrap-mds
        key: AQBICbtTOK9uGBAAdbe5zcIGHZL3T/u2g6EBww==
        caps: [mon] allow profile bootstrap-mds
client.bootstrap-osd
        key: AQBHCbtT4GxqORAADE5u7RkpCN/oo4e5W0uBtw==
        caps: [mon] allow profile bootstrap-osd
```

注意：TYPE.ID 表示法

请注意，针对用户采用 `TYPE.ID` 表示法，例如，`osd.0` 指定 `osd` 类型的用户，其 ID 为 `0`。`client.admin` 是 `client` 类型的用户，其 ID 为 `admin`。另请注意，每个项包含一个 `key: value` 项，以及一个或多个 `caps:` 项。

可以结合使用 `-o 文件名`选项和 **`ceph auth list`** 将输出保存到某个文件。

30.2.2.2 获取有关用户的信息

要检索特定的用户、密钥和用户权限，请执行以下命令：

```
cephuser@adm > ceph auth get TYPE.ID
```

例如：

```
cephuser@adm > ceph auth get client.admin
exported keyring for client.admin
[client.admin]
 key = AQA19uZUqIwkHxAAFuUwvq0eJD4S173oFRxe0g==
 caps mds = "allow"
 caps mon = "allow *"
 caps osd = "allow *"
```

开发人员也可以执行以下命令：

```
cephuser@adm > ceph auth export TYPE.ID
```

`auth export` 命令与 **`auth get`** 相同，不过它还会列显内部身份验证 ID。

30.2.2.3 添加用户

添加一个用户会创建用户名 (`TYPE.ID`)、秘密密钥，以及包含在命令中用于创建该用户的所有用户权限。

用户可使用其密钥向 Ceph 存储集群进行身份验证。用户的用户权限授予该用户在 Ceph Monitor (mon)、Ceph OSD (osd) 或 Ceph 元数据服务器 (mds) 上进行读取、写入或执行的能力。

可以使用以下几个命令来添加用户：

`ceph auth add`

此命令是添加用户的规范方法。它会创建用户、生成密钥，并添加所有指定的用户权限。

`ceph auth get-or-create`

此命令往往是创建用户的最简便方式，因为它会返回包含用户名（在方括号中）和密钥的密钥文件格式。如果该用户已存在，此命令只以密钥文件格式返回用户名和密钥。您可以使用 `-o filename` 选项将输出保存到某个文件。

`ceph auth get-or-create-key`

此命令是创建用户并仅返回用户密钥的简便方式。对于只需要密钥的客户端（例如 `libvirt`），此命令非常有用。如果该用户已存在，此命令只返回密钥。您可以使用 `-o filename` 选项将输出保存到某个文件。

创建客户端用户时，可以创建不具有用户权限的用户。不具有用户权限的用户可以进行身份验证，但不能执行其他操作。此类客户端无法从 Monitor 检索集群索引。但是，如果您希望稍后再添加用户权限，可以使用 **`ceph auth caps`** 命令创建一个不具有用户权限的用户。

典型的用户至少对 Ceph Monitor 具有读取功能，并对 Ceph OSD 具有读取和写入功能。此外，用户的 OSD 权限通常限制为只能访问特定的存储池。

```
cephuser@adm > ceph auth add client.john mon 'allow r' osd \
 'allow rw pool=liverpool'
cephuser@adm > ceph auth get-or-create client.paul mon 'allow r' osd \
 'allow rw pool=liverpool'
cephuser@adm > ceph auth get-or-create client.george mon 'allow r' osd \
 'allow rw pool=liverpool' -o george.keyring
cephuser@adm > ceph auth get-or-create-key client.ringo mon 'allow r' osd \
 'allow rw pool=liverpool' -o ringo.key
```

重要

如果您为某个用户提供了对 OSD 的用户权限，但**未**限制只能访问特定存储池，则该用户将有权访问集群中的**所有**存储池。

30.2.2.4 修改用户权限

使用 **`ceph auth caps`** 命令可以指定用户以及更改该用户的权限。设置新用户权限会重写当前的用户权限。要查看当前用户权限，请运行 **`ceph auth get`** `USERTYPE.USERID`。要添加用户权限，使用以下格式时还需要指定现有用户权限：

```
cephuser@adm > ceph auth caps USERTYPE.USERID daemon 'allow [r|w|x|*|...] \
     [pool=pool-name] [namespace=namespace-name]' [daemon 'allow [r|w|x|*|...] \
     [pool=pool-name] [namespace=namespace-name]']
```

例如：

```
cephuser@adm > ceph auth get client.john
cephuser@adm > ceph auth caps client.john mon 'allow r' osd 'allow rw
 pool=prague'
cephuser@adm > ceph auth caps client.paul mon 'allow rw' osd 'allow r
 pool=prague'
cephuser@adm > ceph auth caps client.brian-manager mon 'allow *' osd 'allow *'
```

要删除某个用户权限，可重设置该用户权限。如果希望用户无权访问以前设置的特定守护进程，请指定一个空字符串：

```
cephuser@adm > ceph auth caps client.ringo mon ' ' osd ' '
```

30.2.2.5 删除用户

要删除用户，请使用 **`ceph auth del`**：

```
cephuser@adm > ceph auth del TYPE.ID
```

其中，`TYPE` 是 `client`、`osd`、`mon` 或 `mds` 之一，`ID` 是用户名或守护进程的 ID。

如果您创建了仅对不再存在的存储池具有许可权限的用户，则应该考虑也删除那些用户。

30.2.2.6 列显用户的密钥

要将用户的身份验证密钥列显到标准输出，请执行以下命令：

```
cephuser@adm > ceph auth print-key TYPE.ID
```

其中，TYPE 是 client、osd、mon 或 mds 之一，ID 是用户名或守护进程的 ID。

需要在客户端软件（例如 libvirt）中填充某个用户的密钥时，列显用户的密钥非常有帮助，如以下示例所示：

```
# mount -t ceph host:/ mount_point \
-o name=client.user,secret=`ceph auth print-key client.user`
```

30.2.2.7 导入用户

要导入一个或多个用户，请使用 **ceph auth import** 并指定密钥环：

```
cephuser@adm > ceph auth import -i /etc/ceph/ceph.keyring
```

注意

Ceph 存储集群将添加新用户及其密钥和用户权限，并更新现有用户及其密钥和用户权限。

30.2.3 管理密钥环

当您通过 Ceph 客户端访问 Ceph 时，该客户端会查找本地密钥环。默认情况下，Ceph 会使用以下四个密钥环名称预设置密钥环设置，因此，除非您要覆盖默认值，否则无需在 Ceph 配置文件中设置这些名称：

```
/etc/ceph/cluster.name.keyring
/etc/ceph/cluster.keyring
/etc/ceph/keyring
/etc/ceph/keyring.bin
```

cluster 元变量是根据 Ceph 配置文件名称定义的 Ceph 集群名称。ceph.conf 表示集群名称为 ceph，因此密钥环名称为 ceph.keyring。name 元变量是用户类型和用户 ID（例如 client.admin），因此密钥环名称为 ceph.client.admin.keyring。

创建用户（例如 `client.ringo`）之后，必须获取密钥并将其添加到 Ceph 客户端上的密钥环，以使该用户能够访问 Ceph 存储集群。

第 30.2 节 “主要管理”详细介绍了如何直接在 Ceph 存储集群中列出、获取、添加、修改和删除用户。不过，Ceph 还提供了 **`ceph-authtool`** 实用程序，可让您从 Ceph 客户端管理密钥环。

30.2.3.1 创建密钥环

当您执行第 30.2 节 “主要管理”中的过程创建用户时，需要向 Ceph 客户端提供用户密钥，以使客户端能检索指定用户的密钥，并向 Ceph 存储集群进行身份验证。Ceph 客户端将访问钥环，以查找用户名并检索用户的密钥：

```
cephuser@adm > ceph-authtool --create-keyring /path/to/keyring
```

创建包含多个用户的密钥环时，我们建议使用集群名称（例如 `cluster`.keyring）作为密钥环文件名，并将其保存在 `/etc/ceph` 目录中，如此，您无需在 Ceph 配置文件的本地副本中指定文件名，密钥环配置默认设置就会选取正确的文件名。例如，可执行以下命令创建 `ceph.keyring`：

```
cephuser@adm > ceph-authtool -C /etc/ceph/ceph.keyring
```

创建包含单个用户的密钥环时，我们建议使用集群名称、用户类型和用户名，并将其保存在 `/etc/ceph` 目录中。例如，为 用户创建 ceph.client.admin.keyring`client.admin`。

30.2.3.2 将用户添加到密钥环

将某个用户添加到 Ceph 存储集群时（请参见第 30.2.2.3 节 “添加用户”），可以检索该用户、密钥和用户权限，并将该用户保存到密钥环。

如果您只想对每个密钥环使用一个用户，可以结合 `-o` 选项使用 **`ceph auth get`** 命令以密钥环文件格式保存输出。例如，要为 `client.admin` 用户创建密钥环，请执行以下命令：

```
cephuser@adm > ceph auth get client.admin -o /etc/ceph/ceph.client.admin.keyring
```

想要将用户导入到密钥环时，可以使用 **`ceph-authtool`** 指定目标密钥环和源密钥环：

```
cephuser@adm > ceph-authtool /etc/ceph/ceph.keyring \
  --import-keyring /etc/ceph/ceph.client.admin.keyring
```

重要

如果您的密钥环遭到入侵，请从 `/etc/ceph` 目录中删除您的密钥，并使用第 30.2.3.1 节 “创建密钥环”中的相同说明重新创建新密钥。

30.2.3.3 创建用户

Ceph 提供 **`ceph auth add`** 命令用于直接在 Ceph 存储集群中创建用户。但是，您也可以直接在 Ceph 客户端密钥环中创建用户、密钥和用户权限。然后，可将用户导入到 Ceph 存储集群：

```
cephuser@adm > ceph-authtool -n client.ringo --cap osd 'allow rwx' \
  --cap mon 'allow rwx' /etc/ceph/ceph.keyring
```

您也可以在创建密钥环的同时将新用户添加到该密钥环：

```
cephuser@adm > ceph-authtool -C /etc/ceph/ceph.keyring -n client.ringo \
  --cap osd 'allow rwx' --cap mon 'allow rwx' --gen-key
```

在前面的方案中，新用户 `client.ringo` 仅存放在密钥环中。要将该新用户添加到 Ceph 存储集群，仍必须手动添加：

```
cephuser@adm > ceph auth add client.ringo -i /etc/ceph/ceph.keyring
```

30.2.3.4 修改用户

要修改密钥环中某条用户记录的用户权限，请指定该密钥环和用户，然后指定用户权限：

```
cephuser@adm > ceph-authtool /etc/ceph/ceph.keyring -n client.ringo \
  --cap osd 'allow rwx' --cap mon 'allow rwx'
```

要在 Ceph 集群环境中更新已修改的用户，必须将密钥环中的更改导入到 Ceph 集群中的用户项：

```
cephuser@adm > ceph auth import -i /etc/ceph/ceph.keyring
```

请参见第 30.2.2.7 节 “导入用户”，了解有关根据密钥环更新 Ceph 存储集群用户的详细信息。

30.2.4 命令行用法

ceph 命令支持以下与用户名和机密操作相关的选项：

`--id` **或** `--user`

Ceph 使用类型和 ID（TYPE.ID，例如 `client.admin` 或 `client.user1`）来标识用户。使用 `id`、`name` 和 `-n` 选项可以指定用户名的 ID 部分（例如 `admin` 或 `user1`）。可以使用 --id 指定用户，并省略类型。例如，要指定用户 client.foo，请输入以下命令：

```
cephuser@adm > ceph --id foo --keyring /path/to/keyring health
cephuser@adm > ceph --user foo --keyring /path/to/keyring health
```

`--name` **或** `-n`

Ceph 使用类型和 ID（TYPE.ID，例如 `client.admin` 或 `client.user1`）来标识用户。使用 `--name` 和 `-n` 选项可以指定完全限定的用户名。必须指定用户类型（通常是 `client`）和用户 ID：

```
cephuser@adm > ceph --name client.foo --keyring /path/to/keyring health
cephuser@adm > ceph -n client.foo --keyring /path/to/keyring health
```

`--keyring`

包含一个或多个用户名和机密的密钥环的路径。`--secret` 选项提供相同的功能，但它不适用于对象网关，该网关将 `--secret` 用于其他目的。可以使用 **`ceph auth get-or-create`** 检索密钥环并将其存储在本地。这是首选的做法，因为无需切换密钥环路径就能切换用户名：

```
cephuser@adm > rbd map --id foo --keyring /path/to/keyring mypool/myimage
```

A 基于上游“Pacific”小版本的 Ceph 维护更新

SUSE Enterprise Storage 7.1 中的几个关键软件包均基于 Ceph 的 Pacific 版本系列。当 Ceph 项目 (https://github.com/ceph/ceph) 在 Pacific 系列中发布新的小版本时，SUSE Enterprise Storage 7.1 即会更新，以确保产品应用最新的上游 Bug 修复并可进行功能向后移植。

本章简要介绍有关已经或计划在产品中包含的每个上游小版本中的重要更改。

词汇表

一般

Alertmanager

用于处理 Prometheus 服务器发送的告警并通知最终用户的单个二进制文件。

Ceph Dashboard

一个内置的基于 Web 的 Ceph 管理和监控应用，负责管理集群的各个方面和对象。仪表盘作为 Ceph Manager 模块实施。

Ceph Manager

Ceph Manager 或 MGR 是 Ceph Manager 软件，会将整个集群的所有状态收集到一处。

Ceph Monitor

Ceph Monitor 或 MON 是 Ceph Monitor 软件。

Ceph OSD 守护进程

`ceph-osd` 守护进程是 Ceph 的组件，负责在本地文件系统上存储对象并在网络中提供对它们的访问。

Ceph 存储集群

用于存储用户数据的存储软件的核心集合。此类集合由 Ceph Monitor 和 OSD 组成。

Ceph 客户端

可以访问 Ceph 存储集群的 Ceph 组件集合。其中包括对象网关、Ceph 块设备、CephFS 及其相应的库、内核模块和 FUSE 客户端。

Ceph 对象存储

对象存储“产品”、服务或功能，由 Ceph 存储集群和 Ceph 对象网关组成。

`ceph-salt`

提供使用 Salt 部署由 cephadm 管理的 Ceph 集群的工具。

cephadm

cephadm 用于部署和管理 Ceph 集群，它会通过 SSH 从 Manager 守护进程连接到主机以添加、删除或更新 Ceph 守护进程容器。

CephFS

Ceph 文件系统。

CephX

Ceph 身份验证协议。Cephx 操作与 Kerberos 类似，但它没有单一故障点。

CRUSH 规则

适用于某个特定存储池或多个存储池的 CRUSH 数据归置规则。

CRUSH、CRUSH 索引

基于可缩放哈希的受控复制：通过计算数据存储位置来确定如何存储和检索数据的算法。CRUSH 需要获取集群的索引来以伪随机的方式在 OSD 中存储和检索数据，并以一致的方式在整个集群中分布数据。

DriveGroups

DriveGroups 是可映射到物理驱动器的一个或多个 OSD 布局的声明。OSD 布局定义了 Ceph 如何在符合指定准则的媒体上实际分配 OSD 存储。

Grafana

数据库分析和监视解决方案。

OSD

对象存储设备：一个物理或逻辑存储单元。

OSD 节点

一个集群节点，用于存储数据、处理数据复制、恢复、回填、重新平衡以及通过检查其他 Ceph OSD 守护进程为 Ceph Monitor 提供某些监视信息。

PG

归置组：**存储池**的细分，用于性能调节。

Prometheus

系统监视和警告工具箱。

RADOS 块设备 (RBD)

Ceph 的块存储组件。也称为 Ceph 块设备。

Samba

Windows 集成软件。

Samba 网关

Samba 网关加入到 Windows 域中的 Active Directory，以进行用户身份验证和授权。

元数据服务器

元数据服务器或 MDS 是 Ceph 元数据软件。

区域组

可靠自主分布式对象存储 (RADOS)

用于存储用户数据的存储软件的核心集合 (MON+OSD)。

多区域

存储桶

将其他节点聚合成物理位置层次结构的一个点。

存档同步扩展模块

允许创建对象网关区域以保留 S3 对象版本历史记录的扩展模块。

对象网关

Ceph 对象存储的 S3/Swift 网关组件。也称为 RADOS 网关 (RGW)。

小版本

任何只包含 bug 或安全修复的临时版本。

池

用于存储磁盘映像等对象的逻辑分区。

管理节点

在其上运行与 Ceph 相关的命令以管理集群主机的主机。

节点

Ceph 集群中的任何一台计算机或服务器。

规则组

用于确定存储池的数据归置的规则。

路由树

该术语指的是展示接收器可运行的不同路由的任何示意图。