Red Hat Ceph Storage 3

配置指南

Red Hat Ceph Storage 的配置设置

Last Updated: 2023-02-19

Red Hat Ceph Storage 3 配置指南

Red Hat Ceph Storage 的配置设置

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律通告

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

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本文档提供有关在引导时配置 Red Hat Ceph Storage 和运行时的信息。它还提供配置参考信息。

目录

第 1 章 配置参考

所有 Ceph 集群都有一个配置，它定义：

- 集群身份
- 身份验证设置
- 集群中的 Ceph 守护进程成员资格
- 网络配置
- 主机名和地址
- keyring 的路径
- 到数据的路径（包括日志）
- 其他运行时选项

红帽存储控制台或 Ansible 等部署工具通常会为您创建初始 Ceph 配置文件。但是，如果您想在不使用部署工具的情况下引导集群，您可以创建一个自己。

为方便起见，每个守护进程都有一系列默认值，即 **ceph/src/common/config_opts.h** 脚本。您可以通过 Ceph 配置文件或运行时覆盖这些设置，方法是使用 monitor **tell** 命令，或直接连接到 Ceph 节点上的守护进程套接字。

1.1. 常规建议

您可以在任何地方维护 Ceph 配置文件，但红帽建议您有一个管理节点，用来维护 Ceph 配置文件的主副本。

对 Ceph 配置文件进行更改时，最好将更新的配置文件推送到 Ceph 节点，以保持一致性。

1.2. 配置文件结构

Ceph 配置文件在启动时配置 Ceph 守护进程，覆盖默认值。Ceph 配置文件采用一种 *格式* 的语法。您可以在前面的注释中用井号 (#) 或分号 (;) 添加注释。例如：

```
# <--A number (#) sign precedes a comment.
; A comment may be anything.
# Comments always follow a semi-colon (;) or a pound (#) on each line.
# The end of the line terminates a comment.
# We recommend that you provide comments in your configuration file(s).
```

配置文件可以在 Ceph 存储集群中配置所有 Ceph 守护进程，或者在启动时配置特定类型的所有 Ceph 守护进程。要配置一系列守护进程，必须在接收配置的进程中包含设置，如下所示：

[global]

描述

[global] 下的设置会影响 Ceph Storage 集群中的所有守护进程。

示例

```
auth supported = cephx
```

[osd]

描述

[osd] 下的设置会影响 Ceph 存储集群中的所有 **ceph-osd** 守护进程，并覆盖 **[global]** 中的相同设置。

示例

OSD 日志大小 = 1000

[mon]

描述

[mon] 下的设置会影响 Ceph 存储集群中的所有 **ceph-mon** 守护进程，并覆盖 **[global]** 中的相同设置。

示例

mon host = hostname1,hostname2,hostname3 mon addr = 10.0.0.101:6789

[client]

描述

[client] 下的设置会影响所有 Ceph 客户端（例如，挂载的 Ceph 块设备、Ceph 对象网关等）。

示例

log file = /var/log/ceph/radosgw.log

全局设置会影响 Ceph 存储集群中所有守护进程的所有实例。将 **[global]** 设置用于 Ceph 存储集群中所有守护进程通用的值。您可以通过以下方法覆盖每个 **[global]** 设置：

1. 更改特定进程类型中的设置（例如 **[osd]**、**[mon]**）。
2. 更改特定进程的设置（例如，**[osd.1]** ）。

覆盖全局设置会影响所有子进程，除了您在特定守护进程中特别覆盖的子进程。

典型的全局设置涉及激活身份验证。例如：

```
[global]
#Enable authentication between hosts within the cluster.
auth_cluster_required = cephx
auth_service_required = cephx
auth_client_required = cephx
```

您可以指定适用于特定类型的守护进程的设置。当您指定 **[osd]** 或 **[mon]** 下的设置而不指定特定实例，则设置将分别应用于所有 OSD 或监控守护进程。

一个典型的守护进程范围内的设置是设置日志大小、文件存储设置等，例如：

```
[osd]
osd_journal_size = 1000
```

您可以为守护进程的特定实例指定设置。您可以通过输入类型，以句点 (.) 分隔来指定一个实例，或通过实例 ID 来指定实例。Ceph OSD 守护进程的实例 ID 始终是数字，但 Ceph 监视器可能会是字母数字。

```
[osd.1]
# settings affect osd.1 only.
```

```
[mon.a]
# settings affect mon.a only.
```

默认的 Ceph 配置文件位置按顺序包括：

1. **$CEPH_CONF** (跟随 **$CEPH_CONF** 环境变量的路径)
2. **-c path/path** （ **-c** 命令行参数）
3. **/etc/ceph/ceph.conf**
4. **~/.ceph/config**
5. **./Ceph.conf** （在当前工作目录中）

典型的 Ceph 配置文件至少具有以下设置：

```
[global]
fsid = {cluster-id}
mon_initial_members = {hostname}[, {hostname}]
mon_host = {ip-address}[, {ip-address}]

#All clusters have a front-side public network.
#If you have two NICs, you can configure a back side cluster
#network for OSD object replication, heart beats, backfilling,
#recovery, and so on
public_network = {network}[, {network}]
#cluster_network = {network}[, {network}]

#Clusters require authentication by default.
auth_cluster_required = cephx
auth_service_required = cephx
auth_client_required = cephx

#Choose reasonable numbers for your journals, number of replicas
#and placement groups.
osd_journal_size = {n}
osd_pool_default_size = {n}  # Write an object n times.
osd_pool_default_min_size = {n} # Allow writing n copy in a degraded state.
osd_pool_default_pg_num = {n}
osd_pool_default_pgp_num = {n}

#Choose a reasonable crush leaf type.
#0 for a 1-node cluster.
#1 for a multi node cluster in a single rack
#2 for a multi node, multi chassis cluster with multiple hosts in a chassis
#3 for a multi node cluster with hosts across racks, and so on
osd_crush_chooseleaf_type = {n}
```

1.3. METAVARIABLES

Metavariables 简化了 Ceph 存储集群配置。在配置值中设置 metavariable 时，Ceph 会将 metavariable 扩展为 Concrete 值。

在 Ceph 配置文件的 **[global]**, **[osd]**, **[mon]**, 或 **[client]** 部分中使用时，Mtavariables 非常强大。但是，您也可以在管理套接字中使用它们。Ceph 元变量与 Bash shell 扩展类似。

Ceph 支持以下元变量：

$cluster

描述

扩展至 Ceph 存储集群名称。在同一硬件上运行多个 Ceph 存储群集时很有用。

示例

/etc/ceph/$cluster.keyring

默认

ceph

$type

描述

根据即时守护进程的类型，扩展至 **osd** 或 **mon** 之一。

示例

/var/lib/ceph/$type

$id

描述

扩展至守护进程标识符。对于 **osd.0**，这将是 **0**。

示例

/var/lib/ceph/$type/$cluster-$id

$host

描述

扩展至即时守护进程的主机名。

$name

描述

扩展至 **$type.$id**。

示例

/var/run/ceph/$cluster-$name.asok

1.4. 查看 CEPH 运行时配置

要查看运行时配置，请登录 Ceph 节点并执行：

```
ceph daemon {daemon-type}.{id} config show
```

例如，如果要查看 **osd.0** 的配置，请登录到包含 **osd.0** 的节点并执行：

```
ceph daemon osd.0 config show
```

如需附加选项，指定守护进程和**帮助**。例如：

```
ceph daemon osd.0 help
```

1.5. 在运行时获取特定配置设置

要在运行时获取特定的配置设置，请登录 Ceph 节点并执行：

```
ceph daemon {daemon-type}.{id} config get {parameter}
```

例如，要检索 **osd.0** 的公共地址，请执行：

```
ceph daemon osd.0 config get public_addr
```

1.6. 在运行时设置特定配置

有两种常规方法来设置运行时配置：

- 使用 Ceph 监控器
- 使用管理套接字

您可以使用 **tell** 和 **injectargs** 命令联系 monitor 来设置 Ceph 运行时配置设置。要使用这个方法，您要修改的监控器和守护进程必须正在运行：

```
ceph tell {daemon-type}.{daemon id or *} injectargs --{name} {value} [--{name} {value}]
```

将 **{daemon-type}** 替换为 **osd** 或 **mon** 之一。您可以使用 * 将运行时设置应用到特定类型的所有守护进程，或者指定特定守护进程的 ID（即号码或名称）。例如，要将名为 **osd.0** 的 **ceph-osd** 守护进程的调试日志记录更改为 **0/5**，请执行以下命令：

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

tell 命令有多个参数，因此 **tell** 的每个参数都必须用单引号括起来，且配置前带有两个短划线 (**'--{config_opt} {opt-val}' ['-{config_opt} {opt-val}']**)。**守护进程** 命令不需要引号，因为它仅使用一个参数。

ceph tell 命令进入 monitor。如果您无法绑定到监控器，您仍然可以进行改变，登陆到需要修改配置的守护进程所在的主机，**ceph daemon** 进行修改。例如：

```
sudo ceph osd.0 config set debug_osd 0/5
```

1.7. 常规配置参考

常规设置通常由部署工具自动设置。

fsid

描述

文件系统 ID。每个集群一个。

类型

UUID

必需

No.

默认

不适用。通常由部署工具生成。

admin_socket

描述

在守护进程上执行管理命令的套接字，无论 Ceph 监视器是否建立了仲裁。

类型

字符串

必需

否

默认

/var/run/ceph/$cluster-$name.asok

pid_file

描述

监控或 OSD 将在其中写入其 PID 的文件。例如，**/var/run/$cluster/$type.$id.pid** 将为在 **ceph** 集群中运行的 id 为 **a** 的 **mon** 创建 /var/run/ceph/mon.a.pid。当守护进程安全停止时，将删除 **pid 文件**。如果进程不是守护进程化（使用 **-f** 或 **-d** 选项运行），则不会创建 **pid 文件**。

类型

字符串

必需

否

默认

否

chdir

描述

目录 Ceph 守护进程在启动并运行后会变为。建议使用默认 / 目录。

类型

字符串

必需

否

默认

/

max_open_files

描述

如果设置，当 Red Hat Ceph Storage 集群启动时，Ceph 会在 OS 级别设置 **max_open_fds**（即，最大文件描述符的数量）。它有助于防止 Ceph OSD 耗尽文件描述符。

类型

64 位整数

必填

否

默认

0

fatal_signal_handlers

描述

如果设置，我们将为 SEGV、ABRT、BUS、ILL、FPE、XCPU、XCPU、XZ、SYS 信号安装信号处理器，以生成有用的日志消息。

类型

布尔值

默认

true

1.8. OSD 内存目标

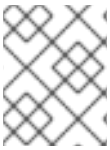
BlueStore 将 OSD 堆内存使用量保留在指定目标大小下，并使用 **osd_memory_target** 配置选项。

选项 **osd_memory_target** 根据系统中可用的 RAM 来设置 OSD 内存。默认情况下，Aisble 会将值设为 4 GB。在部署守护进程时，您可以在 **/usr/share/ceph-ansible/group_vars/all.yml** 文件中更改以字节表示的值。

示例：将 **osd_memory_target** 设置为 6000000000 字节

```
ceph_conf_overrides:
  osd:
    osd_memory_target=6000000000
```

当块设备速度较慢时（例如，传统的硬盘驱动器），Ceph OSD 内存缓存更为重要，因为缓存命中的好处要高于固态硬盘的情况。但是，这需要考虑 OSD 与其他服务共处的情况，比如在超融合基础架构 (HCI) 或其他应用程序中。



注意

osd_memory_target 的值对于传统硬盘设备是每个设备一个 OSD，对于 NVMe SSD 设备是每个设备两个 OSD。**osds_per_device** 在 **group_vars/osds.yml** 文件中定义。

其它资源

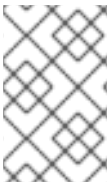
- 设置 **osd_memory_target** Setting OSD Memory Target

1.9. MDS 缓存内存限制

MDS 服务器将其元数据保留在一个单独的存储池中（名为 **cephfs_metadata**），并且是 Ceph OSD 的用户。对于 Ceph 文件系统，MDS 服务器必须支持整个 Red Hat Ceph Storage 集群，而不支持存储集群中的单个存储设备，因此它们的内存要求可能会非常显著，特别是当工作负载包含小时时，数据元数据的比例更大。

Example:Set the **mds_cache_memory_limit** to 2000000000 bytes

```
ceph_conf_overrides:
  osd:
    mds_cache_memory_limit=2000000000
```



注意

对于具有元数据密集型工作负载的大型 Red Hat Ceph Storage 集群，请不要将 MDS 服务器与其他内存密集型服务位于同一个节点上，这样做可让您将更多内存分配给 MDS，例如，大于 100 GB。

其它资源

- 请参阅了解 MDS 缓存大小限制

第 2 章 网络配置参考

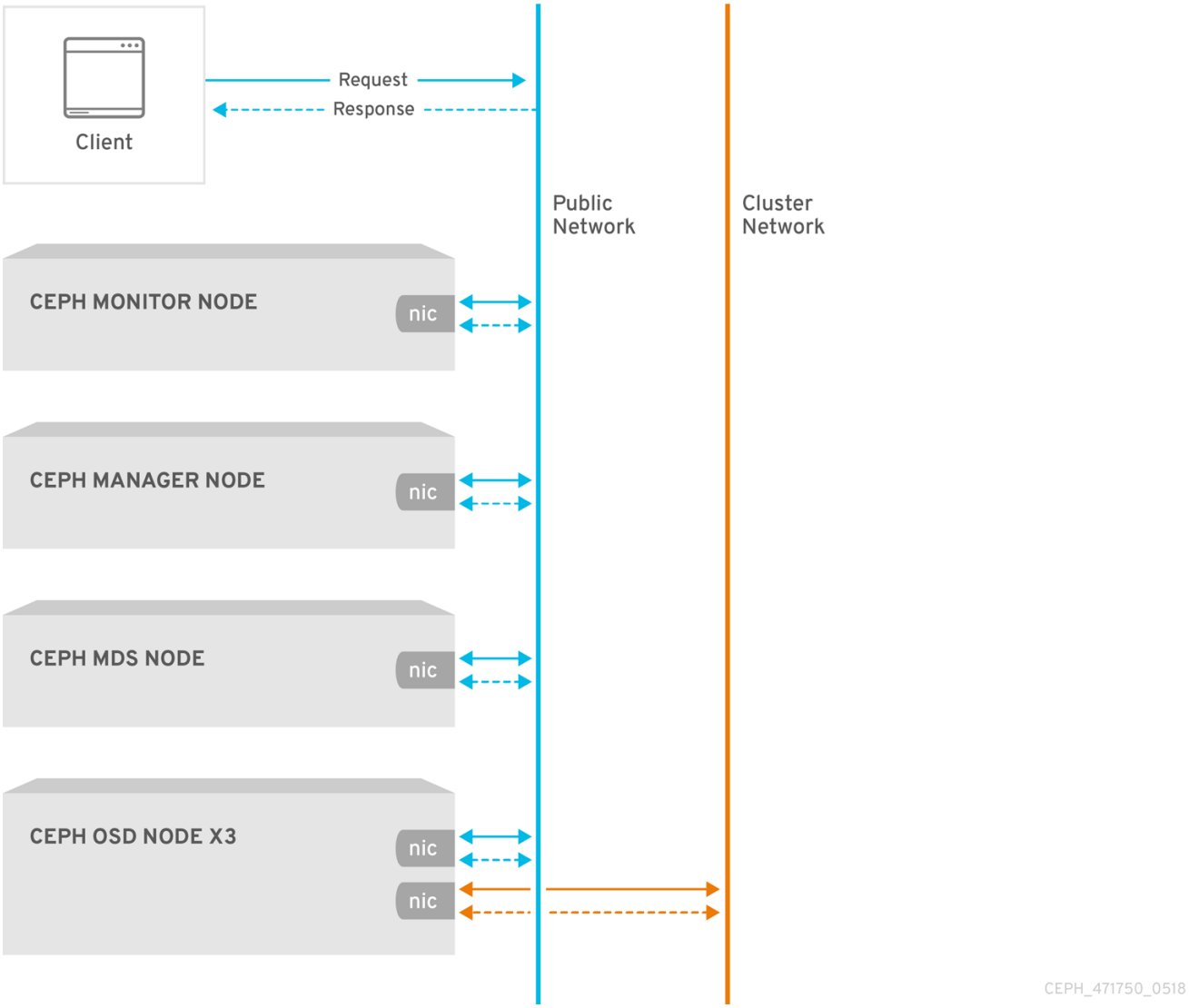
网络配置对于构建高性能 Red Hat Ceph Storage 集群至关重要。Ceph 存储集群不代表 Ceph 客户端执行请求路由或分配请求。相反，Ceph 客户端直接向 Ceph OSD 守护进程发出请求。Ceph OSD 代表 Ceph 客户端执行数据复制，这意味着复制和其他因素对 Ceph 存储集群的网络造成额外的负载。

所有 Ceph 集群都必须使用公共网络。但是，除非指定了一个集群（内部）网络，Ceph 假定有一个公共网络。Ceph 可以在只有一个公共网络的情况下运行，但对于大型集群，如果还有第二个 "cluster" 网络，则性能会显著提高。

红帽建议运行具有两个网络的 Ceph 存储集群：

- 公共网络
- 以及集群网络。

要支持两个网络，每个 Ceph 节点都需要有一个以上的网络接口卡 (NIC) 。



需要考虑操作两个独立网络的原因有很多：

- **性能：** Ceph OSD 处理 Ceph 客户端的数据复制。当 Ceph OSD 多次复制数据时，Ceph OSD 之间网络负载可轻松地在 Ceph 客户端和 Ceph 存储集群之间分配网络负载。这会引入延迟并创建性能问题。恢复和重新平衡还在公共网络上引入大量延迟。

- **安全** ：虽然大多数人用户都会正常使用资源，但有些人可能会参与所谓的拒绝服务 (DoS) 攻击。当 Ceph OSD 之间的流量中断时，peering 可能会失败，放置组可能无法反映 **活跃的 + clean** 状态，这可能会阻止用户读取和写入数据。缓解这类攻击的一个好方法是，维护一个完全独立的、不直接连接到互联网的集群网络。

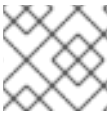
2.1. 网络配置设置

不需要网络配置设置。Ceph 可只用于公共网络，假定在运行 Ceph 守护进程的所有主机上都配置了公共网络。但是，Ceph 允许您建立更加具体的标准，包括用于公共网络的多个 IP 网络和子网掩码。您还可以建立一个单独的集群网络来处理 OSD 心跳、对象复制和恢复流量。

不要将您在配置中设置的 IP 地址与面向公共的 IP 地址网络客户端混淆。典型的内部 IP 网络通常为 **192.168.0.0** 或 **10.0.0.0**。

提示

如果您为公共或集群网络指定多个 IP 地址和子网掩码，则网络中的子网必须能够相互路由。另外，请确保在 IP 表中包括每个 IP 地址/子网，并根据需要打开端口。



注意

Ceph 使用 CIDR 表示法作为子网（例如，1**0.0.0.0/24**）。

当配置网络时，您可以重启集群或重启每个守护进程。Ceph 守护进程动态绑定，因此如果更改网络配置，不必一次重启整个集群。

2.1.1. 公共网络

要配置公共网络，请在 Ceph 配置文件的 **[global]** 部分添加以下选项：

```
[global]
    ...
    public_network = <public-network/netmask>
```

公共网络配置允许您为公共网络定义 IP 地址和子网。您可以使用特定守护进程的 **公共 addr** 设置特别分配静态 **IP 地址或覆盖公共网络** 设置。

public_network

描述

公共（前端）网络的 IP 地址和子网掩码（例如，**192.168.0.0/24**）。在 **[global]** 中设置。您可以指定以逗号分隔的子网。

类型

<ip-address>/<netmask> [, <ip-address>/<netmask>]

必填

否

默认

N/A

public_addr

描述

公共（前端）网络的 IP 地址。为每个守护进程设置。

类型

IP 地址

必填

否

默认

N/A

2.1.2. 集群网络

如果您创建集群网络，OSD 会通过集群网络路由心跳、对象复制和恢复流量。与使用单个网络相比，这可以提高性能。要配置集群网络，请在 Ceph 配置文件的 **[global]** 部分添加以下选项：

```
[global]
    ...
    cluster_network = <cluster-network/netmask>
```

最好的情况是，不能从公共网络或互联网访问集群网络以提高安全性。

集群网络配置允许您声明集群网络，并具体为集群网络定义 IP 地址和子网。您可以使用特定 OSD 守护进程的集群 **addr 设置来具体分配静态 IP 地址或覆盖 集群 网络设置**。

cluster_network

描述

集群网络的 IP 地址和子网掩码（例如 **10.0.0.0/24**）。在 **[global]** 中设置。您可以指定以逗号分隔的子网。

类型

<ip-address>/<netmask> [, <ip-address>/<netmask>]

必填

否

默认

N/A

cluster_addr

描述

集群网络的 IP 地址。为每个守护进程设置。

类型

地址

必填

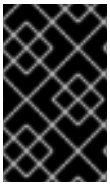
否

默认

N/A

2.1.3. 验证并配置 MTU 值

最大传输单元 (MTU) 值是链路层上发送的最大数据包的大小（以字节为单位）。默认的 MTU 值为 1500 字节。红帽建议将巨型帧（MTU 值为 9000 字节）用于 Red Hat Ceph Storage 集群。



重要

Red Hat Ceph Storage 在通信路径的所有网络设备中，公共和集群网络需要相同的 MTU 值。在在生产环境中使用 Red Hat Ceph Storage 集群之前，验证环境中所有节点和网络设备上的 MTU 值相同。

注意

将网络接口绑定在一起时，MTU 值只需要在绑定接口上设置。新的 MTU 值从绑定设备传播到底层网络设备。

先决条件

- 节点的根级别访问权限。

流程

1. 验证当前的 MTU 值：

   示例

   ```
   [root@mon ~]# ip link list
   1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
   DEFAULT group default qlen 1000
       link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   2: enp22s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
   mode DEFAULT group default qlen 1000
   ```

   在本例中，网络接口是 **enp22s0f0**，其 MTU 值为 **1500**。

2. **临时更改** 在线 MTU 值：

   语法

   ```
   ip link set dev NET_INTERFACE mtu NEW_MTU_VALUE
   ```

   示例

   ```
   [root@mon ~]# ip link set dev enp22s0f0 mtu 9000
   ```

3. **以永久更改** MTU 值。

   a. 打开针对特点网络接口的网络配置文件进行编辑：

      语法

      ```
      vim /etc/sysconfig/network-scripts/ifcfg-NET_INTERFACE
      ```

      示例

```
[root@mon ~]# vim /etc/sysconfig/network-scripts/ifcfg-enp22s0f0
```

b. 在新行中添加 **MTU=9000** 选项：

示例

```
NAME="enp22s0f0"
DEVICE="enp22s0f0"
MTU=9000 (1)
ONBOOT=yes
NETBOOT=yes
UUID="a8c1f1e5-bd62-48ef-9f29-416a102581b2"
IPV6INIT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
```

c. 重启网络服务：

示例

```
[root@mon ~]# systemctl restart network
```

其它资源

- 详情请查看 Red Hat Enterprise Linux 7 的 *网络指南*。

2.1.4. 消息传递

messenger 是 Ceph 网络层实施。红帽支持两种 messenger 类型：

- **simple**
- **async**

在 RHCS 2 及更早的版本中，**简单** 是默认的 mesenger 类型。在 RHCS 3 中，**async** 是默认的 messenger 类型。要更改 messenger 类型，请在 Ceph 配置文件的 **[global]** 部分中指定 **ms_type** 配置设置。

注意

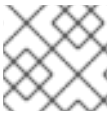
对于 **async** messenger，红帽支持 **posix** 传输类型，但目前不支持 **rdma** 或 **dpdk**。默认情况下，RHCS 3 中的 **ms_type** 设置应反映 **async+posix**，其中 **async** 是 messenger 类型，**posix** 传输类型。

关于 SimpleMessenger

SimpleMessenger 实施使用每个套接字有两个线程的 TCP 套接字。Ceph 将每个逻辑会话与连接相关联。管道处理连接，包括每个消息的输入和输出。尽管 **SimpleMessenger** 对 **posix** 传输类型有效，但它不适用于 **rdma** 或 **dpdk** 等其他传输类型。因此，**AsyncMessenger** 是 RHCS 3 及更新的版本的默认 mesenger 类型。

关于 AsyncMessenger

对于 RHCS 3，**AsyncMessenger** 实现使用带有固定大小的线程池的 TCP 套接字进行连接，这应该等于最大副本数或纠删代码区块。如果性能因为 CPU 数量较低或每个服务器有大量 OSD，可以将线程数设置为较低值。



注意

红帽目前不支持其他传输类型，如 **rdma** 或 **dpdk**。

messenger Type Settings

ms_type

描述

网络传输层的 messenger 类型。红帽支持使用 **posix** 语义的 **simple** 和 **async** messenger 类型。

类型

字符串.

必需

No.

默认

async+posix

ms_public_type

描述

公共网络的网络传输层的 messenger 类型。它的工作方式与 **ms_type** 相同，但仅适用于公共网络或前端网络。此设置可让 Ceph 为公共或前端和集群或后端网络使用不同的 mesenger 类型。

类型

字符串.

必需

No.

默认

无。

ms_cluster_type

描述

集群网络的网络传输层的 messenger 类型。它的工作方式与 **ms_type** 相同，但仅适用于集群或后端网络。此设置可让 Ceph 为公共或前端和集群或后端网络使用不同的 mesenger 类型。

类型

字符串.

必需

No.

默认

无。

2.1.5. AsyncMessenger 设置

ms_async_transport_type

描述

AsyncMessenger 使用的传输类型。红帽支持 **posix** 设置，但目前不支持 **dpdk** 或 **rdma** 设置。POSIX 使用标准 TCP/IP 网络，它是默认值。其他传输类型是实验性的，因此**不被**支持。

类型

字符串

必需

否

默认

posix

ms_async_op_threads

描述

每个 **AsyncMessenger** 实例使用的初始 worker 线程数。此配置设置 **SHOULD** 等于副本或纠删代码区块的数量。但如果 CPU 内核数量较低或单个服务器上的 OSD 数量很高，则可设置它。

类型

64-bit Unsigned 整数

必需

否

默认

3

ms_async_max_op_threads

描述

每个 **AsyncMessenger** 实例使用的最大 worker 线程数量。如果 OSD 主机具有有限 CPU 数量，并且如果 Ceph 利用率不足，则设置为较低值。

类型

64-bit Unsigned 整数

必需

否

默认

5

ms_async_set_affinity

描述

设置为 **true**，将 **AsyncMessenger** worker 绑定到特定的 CPU 内核。

类型

布尔值

必需

否

默认

true

ms_async_affinity_cores

描述

当 **ms_async_set_affinity** 为 **true** 时，该字符串指定了将 **AsyncMessenger** worker 绑定到 CPU 内核的方式。例如： **0,2** 会将 worker #1 和 #2 分别绑定到 CPU 内核 #0 和 #2。 **注意：** 在手动设置关联性时，确保不将 worker 分配给创建的虚拟 CPU，作为超线程或类似技术的影响，因为它们比物理 CPU 内核慢。

类型

字符串

必需

否

默认

(empty)

ms_async_send_inline

描述

直接从生成它们的线程中直接发送消息，而不是排队并从 **AsyncMessenger** 线程发送发送。这个选项已知可以降低具有大量 CPU 内核的系统的性能，因此默认禁用它。

类型

布尔值

必需

否

默认

false

2.1.6. 绑定

BIND 设置设置 Ceph OSD 守护进程使用的默认端口范围。默认范围为 **6800:7100**。确保防火墙配置允许您使用配置的端口范围。

您还可以启用 Ceph 守护进程来绑定到 IPv6 地址。

ms_bind_port_min

描述

OSD 守护进程要绑定到的最低端口号。

类型

32 位整数

默认

6800

必填

否

ms_bind_port_max

描述

OSD 守护进程要绑定到的最大端口号。

类型

32 位整数

默认

7300

必需

No.

ms_bind_ipv6

描述

启用 Ceph 守护进程来绑定到 IPv6 地址。

类型

布尔值

默认

false

必填

否

2.1.7. 主机

Ceph 预期在 Ceph 配置文件中至少声明了一个 monitor，每个声明的 monitor 下都有一个 **mon addr** 设置。Ceph 预期 Ceph 在 Ceph 配置文件中每个声明的 monitor、元数据服务器和 OSD 下的 **host** 设置。

mon_addr

描述

客户端可用于连接到 Ceph 监视器的 **<hostname>:<port>** 条目列表。如果没有设置，Ceph 会搜索 **[mon.*]** 部分。

类型

字符串

必需

否

默认

N/A

主机

描述

主机名。对特定的守护进程实例使用此设置（例如，**[osd.0]**）。

类型

字符串

必填

是，对于守护进程实例。

默认

localhost

提示

不要使用 **localhost**。要获得您的主机名，请执行 **hostname -s** 命令并使用您的主机名到第一个句点，而不是完全限定域名。

重要

在使用检索您的 **主机名** 的第三方部署系统时，不要为主机指定任何值。

2.1.8. TCP

Ceph 默认禁用 TCP 缓冲。

ms_tcp_nodelay

描述

Ceph 启用 **ms_tcp_nodelay**，使每个请求立即发送（无缓冲区）。禁用 Nagle 的算法会增加网络流量，它可以造成拥塞问题。如果您遇到大量小数据包，您可以尝试禁用 **ms_tcp_nodelay**，但请注意，禁用它通常会增加延迟。

类型

布尔值

必需

否

默认

true

ms_tcp_rcvbuf

描述

网络连接结尾的套接字缓冲区的大小。默认禁用。

类型

32 位整数

必填

否

默认

0

ms_tcp_read_timeout

描述

如果客户端或守护进程向另一个 Ceph 守护进程发出请求且不丢弃未使用的连接，则 **tcp read timeout** 会在指定秒数后将连接定义为 idle。

类型

unsigned 64 位整数

必填

否

默认

900 15 分钟。

2.1.9. firewall

默认情况下，守护进程绑定到 **6800:7100** 范围内的端口。您可自行配置此范围。在配置防火墙前，检查默认防火墙配置。您可自行配置此范围。

-

```
sudo iptables -L
```

对于 **firewalld** 守护进程，以 **root** 用户身份执行以下命令：

```
# firewall-cmd --list-all-zones
```

一些 Linux 发行版包含拒绝除来自所有网络接口的 SSH 之外的所有入站请求的规则。例如：

```
REJECT all -- anywhere anywhere reject-with icmp-host-prohibited
```

2.1.9.1. 监控防火墙

Ceph 监视器默认侦听端口 **6789**。此外，Ceph 监视器始终对公共网络运行。当使用以下示例添加规则时，请确保将 **<iface>** 替换为公共网络接口（如 **eth0**、**eth1** 等等），将 **<ip-address>** 替换为公共网络的 IP 地址，将 **<netmask>** 替换为公共网络的子网掩码。

```
sudo iptables -A INPUT -i <iface> -p tcp -s <ip-address>/<netmask> --dport 6789 -j ACCEPT
```

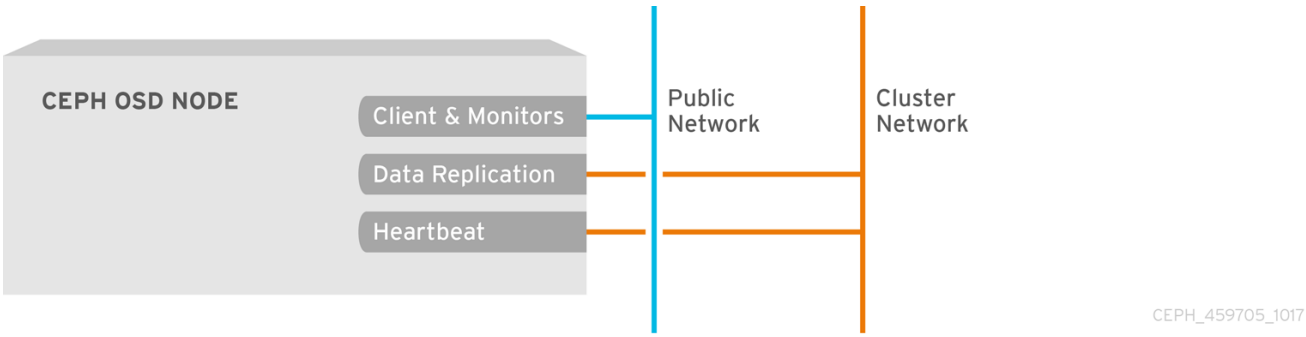
对于 **firewalld** 守护进程，以 **root** 用户身份执行以下命令：

```
# firewall-cmd --zone=public --add-port=6789/tcp
# firewall-cmd --zone=public --add-port=6789/tcp --permanent
```

2.1.9.2. OSD 防火墙

默认情况下，Ceph OSD 绑定到从 6800 端口开始的 Ceph 节点上第一个可用端口。确保为主机上运行的每个 OSD 最少打开三个端口，使其以端口 6800 开始：

1. 个用于与客户端和服务器（公共网络）对话。
2. 个用于将数据发送到其他 OSD（集群网络）。
3. 个用于发送 heartbeat 数据包（集群网络）。



端口特定于节点。但是，如果进程重启并且绑定端口没有释放绑定端口，您可能需要打开比该 Ceph 节点上运行的端口所需的端口数量。当守护进程失败并重启而不释放端口，以便重启的守护进程绑定到新端口时，请考虑打开一些额外的端口。另外，请考虑在每个 OSD 主机上打开 **6800:7300** 端口范围。

如果设置了单独的公共和集群网络，您必须同时为公共网络和集群网络添加规则，因为客户端将使用公共网络连接，其他 Ceph OSD 守护进程将使用集群网络进行连接。

当使用以下示例添加规则时，请确保将 < **iface&gt**; 替换为网络接口（如 **eth0** 或 **eth1**），**'<ip-address &** gt; 替换为公共或集群网络的子网掩码。例如：

```
sudo iptables -A INPUT -i <iface>  -m multiport -p tcp -s <ip-address>/<netmask> --dports 6800:6810
-j ACCEPT
```

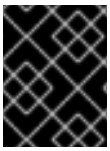
对于 **firewalld** 守护进程，以 **root** 用户身份执行以下命令：

```
# firewall-cmd --zone=public --add-port=6800-6810/tcp
# firewall-cmd --zone=public --add-port=6800-6810/tcp --permanent
```

如果将集群网络放在另一个区中，请根据需要在该区中打开端口。

2.2. CEPH 守护进程

Ceph 有一个网络配置要求适用于所有守护进程。Ceph 配置文件必须为每个守护进程指定 **host**。Ceph 不再要求 Ceph 配置文件指定监控器 IP 地址及其端口。



重要

有些部署实用程序可能会为您创建配置文件。如果部署实用程序为您设置这些值，则不要设置这些值。

提示

主机 设置是主机的短名称（即 FQDN）。它不是 IP 地址。使用 **hostname -s** 命令来检索主机的名称。

```
[mon.a]

    host = <hostname>
    mon addr = <ip-address>:6789

[osd.0]
    host = <hostname>
```

您不必为守护进程设置主机 IP 地址。如果您有一个静态 IP 配置和运行集群网络，Ceph 配置文件可能会为每个守护进程指定主机的 IP 地址。要为守护进程设置静态 IP 地址，以下选项应出现在 Ceph 配置文件的守护进程实例部分中。

```
[osd.0]
    public_addr = <host-public-ip-address>
    cluster_addr = <host-cluster-ip-address>
```

两个网络集群中的一个 NIC OSD

通常，红帽不推荐在有两个网络的集群中部署带有单个 NIC 的 OSD 主机。但是，您可以通过向 Ceph 配置文件的 **[osd.n]** 部分添加 **public addr** 条目来强制 OSD 主机在公共网络上运行，其中 **n** 代表带有一个 NIC 的 OSD 号。另外，公共网络和集群网络需要能够互相路由流量，因此处于安全原因，红帽不推荐这样做。

第 3 章 监控配置参考

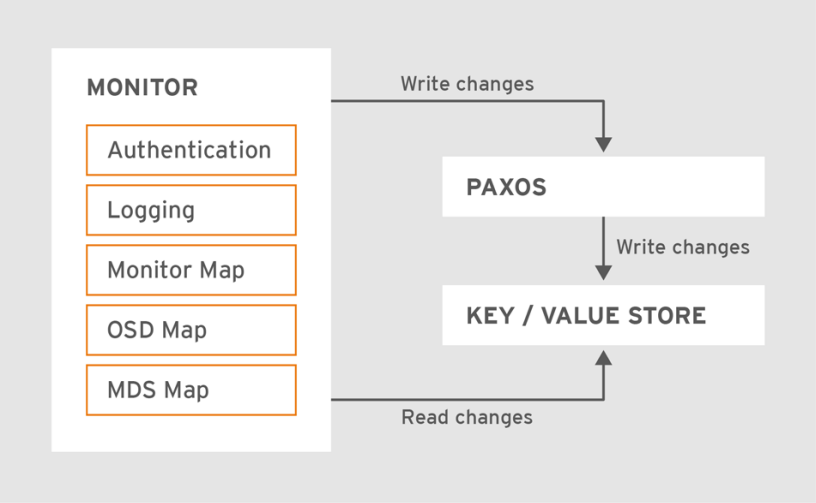
了解如何配置 Ceph 监控器是构建可靠的 Red Hat Ceph Storage 集群的重要部分。所有集群都至少有一个监控器。监控配置通常会保持大体一致，但您可以在集群中添加、删除或替换 monitor。

3.1. 背景信息

Ceph 监视器维护集群映射的"主副本"。这意味着，Ceph 客户端可以通过连接到一个 Ceph 监视器并检索当前 cluster map 来确定所有 Ceph 监视器和 Ceph OSD 的位置。

Ceph 客户端必须先连接到 Ceph 监视器，然后才能从 Ceph OSD 读取或写入到 Ceph OSD。使用 cluster map 的当前副本和 CRUSH 算法时，Ceph 客户端可以计算任何对象的位置。通过计算对象的位置，Ceph 客户端可以直接与 Ceph OSD 通信，这是 Ceph 高可扩展性和性能的一个重要方面。

Ceph 监视器的主要角色是维护 cluster map 的主副本。Ceph 监视器也提供身份验证和日志记录服务。Ceph 监视器将监控服务中的所有更改写入单个 Paxos 实例，Paxos 会将更改写入到键值存储，以实现强一致性。Ceph 监视器可以在同步操作期间查询 cluster map 的最新版本。Ceph Monitor 利用键值存储的快照和迭代器（使用 **leveldb** 数据库）来执行存储范围的同步。



3.1.1. cluster map

集群映射不同映射的一个组合，包括 monitor 映射、OSD 映射和放置组映射。集群映射跟踪多个重要事件：

- 哪些进程在 Red Hat Ceph Storage 集群中为 **in**
- 哪些进程在 Red Hat Ceph Storage 集群中为 **in**，状态为 **up** 并在运行，或为 **down**。
- 放置组是 **活动** 还是 **非活动**，以及 **清理** 或处于某些其他状态。
- 反映集群当前状态的其他详情，例如：
  - 存储空间总量或
  - 使用的存储量。

例如，当集群状态有重大改变时，一个 Ceph OSD 变为 down，放置组进入降级状态，等，集群映射会进行相应的更新以反映集群的当前状态。此外，Ceph 监控器也维护了群集之前状态的历史记录。monitor 映射、OSD 映射和放置组映射各自维护其映射版本的历史记录。每个版本称为一个 **epoch**。

运行 Red Hat Ceph Storage 集群时，要跟踪这些状态是集群管理的一个重要部分。

3.1.2. monitor Quorum

集群将通过单一监控器运行。但是，如果只有一个监控器，则代表有单一故障点。为确保生产 Ceph 存储群集中的高可用性，可运行具有多个监控器的 Ceph，因此当一个控制器出现故障时不会造成整个群集故障。

当 Ceph 存储集群运行多个 Ceph Monitor 以实现高可用性时，Ceph Monitor 使用 Paxos 算法来建立与主集群映射相关的共识。共识需要大多数运行的监控器建立一个仲裁（quorum）以达成对集群映射的共识（例如 1; 2 out of 3; 3 out of 5; 4 out of 6; 等）。

mon_force_quorum_join

描述

强制 monitor 加入仲裁，即使之前已从映射中删除

类型

布尔值

默认

False

3.1.3. 一致性

将监控设置添加到 Ceph 配置文件时，您需要了解 Ceph 监视器的一些架构方面。Ceph 在发现集群中的另一个 Ceph 监控器时，对 Ceph 监视器实施严格的一致性要求。Ceph 客户端和其他 Ceph 守护进程使用 Ceph 配置文件来发现 monitor，monitor 使用 monitor 映射 (**monmap**) 而不是 Ceph 配置文件互相发现。

在发现 Red Hat Ceph Storage 集群中的其他 Ceph 监控器时，Ceph Monitor 始终指 monitor map 的本地副本。使用 monitor map 而不是 Ceph 配置文件可避免可能破坏集群的错误，例如，在指定监控地址或端口时 Ceph 配置文件中的拼写错误。由于 monitor 使用 monitor map 进行发现，并且它们与客户端和其他 Ceph 守护进程共享 monitor 映射，monitor 映射为 monitor 提供严格保证其共识有效。

严格一致性也适用于 monitor 映射的更新。与 Ceph 监视器上的任何其他更新一样，对 monitor 映射的更改始终通过名为 Paxos 的分布式共识算法运行。Ceph monitor 必须同意对 monitor 映射的每个更新，如添加或移除 Ceph monitor，以确保仲裁中的每个 monitor 都有相同的监控器映射版本。monitor 映射的更新递增，以便 Ceph 监视器在版本上具有最新的同意，以及一组以前的版本。维护历史记录可让具有较老版本的 monitor 来获取 Red Hat Ceph Storage 集群的当前状态。

如果 Ceph 监控通过 Ceph 配置文件而不是监视器映射发现相互发现，它将带来额外的风险，因为 Ceph 配置文件没有被自动更新并分发。Ceph 监控可能会意外地使用旧的 Ceph 配置文件，无法识别 Ceph monitor，无法承担仲裁状态，或者开发 Paxos 无法准确确定系统当前状态的情况。

3.1.4. bootstrap 监控器

在大部分配置和部署情形中，部署 Ceph 的工具可以通过为您生成 monitor 映射（例如，Red Hat Storage Console 或 Ansible）帮助引导 Ceph 监视器。Ceph 监控需要一些显式设置：

- **文件系统 ID ：fsid** 是对象存储的唯一标识符。由于您可以在同一硬件上运行多个集群，所以您必须在引导 monitor 时指定对象存储的唯一 ID。使用部署工具，Red Hat Storage Console 或 Ansible 将生成文件系统标识符，但您也可以手动指定 **fsid**。

- **Monitor ID** ： monitor ID 是分配给集群中各个 monitor 的唯一 ID。它是一个字母数字值，按惯例标识符通常会采用一个字母顺序递增（如 **a**、**b** 等等）。您可以使用部署工具或使用 **ceph** 命令在 Ceph 配置文件中进行设置（例如 **[mon.a]**, **[mon.b]**, 等）。

- **Key**: monitor 必须具有 secret 键。

3.2. 配置监控器

要将配置设置应用到整个集群，请在 **[global]** 部分输入配置设置。要将配置设置应用到集群中的所有监控器，请在 **[mon]** 部分输入配置设置。要将配置设置应用到特定的监控器，请指定 monitor 实例（例如 **[mon.a]**）。按照惯例，监控实例名称使用 alpha 表示法。

```
[global]

[mon]

[mon.a]

[mon.b]

[mon.c]
```

3.2.1. 最小配置

如果 Ceph 配置文件中尚未配置 DNS 和 monitor 地址，则 Ceph Monitor 的裸机监控设置包括每个 monitor 的主机名。您可以在 **[mon]** 下或在特定监控器的条目下配置它们。

```
[mon]
mon_host = hostname1,hostname2,hostname3
mon_addr = 10.0.0.10:6789,10.0.0.11:6789,10.0.0.12:6789
```

或者

```
[mon.a]
host = hostname1
mon_addr = 10.0.0.10:6789
```

注意

此 monitor 的最小配置假定部署工具为您生成 **fsid** 和 **mon.** 键。

重要

部署 Ceph 集群后，请勿更改 monitor 的 IP 地址。

从 RHCS 2.4 开始，当集群被配置为通过 DNS 服务器查找监控器时，Ceph 不需要 **mon_host**。要配置 Ceph 集群以进行 DNS 查找，请在 Ceph 配置文件中设置 **mon_dns_srv_name** 设置。

mon_dns_srv_name

描述

用于查询监控主机/地址的 DNS 的服务名称。

类型

字符串

默认

ceph-mon

设置后，配置 DNS。为 DNS 区域中的 monitor 创建 IPv4 (A) 或 IPv6 (AAAA) 记录。例如：

```
#IPv4
mon1.example.com. A 192.168.0.1
mon2.example.com. A 192.168.0.2
mon3.example.com. A 192.168.0.3

#IPv6
mon1.example.com. AAAA 2001:db8::100
mon2.example.com. AAAA 2001:db8::200
mon3.example.com. AAAA 2001:db8::300
```

其中： **example.com** 是 DNS 搜索域。

然后，使用名称 **mon_dns_srv_name** 配置设置创建 SRV TCP 记录，指向三个监控器。以下示例使用默认的 **ceph-mon** 值。

```
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 6789 mon1.example.com.
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 6789 mon2.example.com.
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 6789 mon3.example.com.
```

在默认情况下，监控器在端口 **6789** 上运行，其优先级和权重在 foregoing 示例中分别设置为 **10** 和 **60**。

3.2.2. 集群 ID

每个 Red Hat Ceph Storage 集群都有一个唯一标识符 (**fsid**) 。如果指定，它通常出现在配置文件的 **[global]** 部分中。部署工具通常会生成 **fsid** 并将其存储在 monitor 映射中，因此该值可能不会出现在配置文件中。通过 **fsid**，可以在同一硬件上为多个集群运行守护进程。

fsid

描述

集群 ID。每个集群一个。

类型

UUID

必填

是。

默认

不适用。如果未指定，则由部署工具生成。

注意

如果使用部署工具，则不要设置这个值。

3.2.3. 初始成员

红帽建议运行至少具有三个 Ceph 监视器的生产环境 Red Hat Ceph Storage 集群，以确保高可用性。运行多个监视器时，您可以指定必须成为集群成员的初始监控器，才能建立仲裁。这可减少集群在线所需的时间。

```
[mon]
mon_initial_members = a,b,c
```

mon_initial_members

描述

启动期间集群中初始 monitor 的 ID。如果指定，Ceph 需要奇数个 monitor 来形成初始仲裁（例如 3）。

类型

字符串

默认

无

注意

集群 *中的大多数* monitor 必须能够相互连接，才能建立仲裁。您可以使用此设置减少初始数量的 monitor，以建立仲裁。

3.2.4. data

Ceph 提供了 Ceph 监视器存储数据的默认路径。为了在生产环境 Red Hat Ceph Storage 集群中获得最佳性能，红帽建议在独立于主机和 Ceph OSD 的驱动器上运行 Ceph 监视器。Ceph 监控器经常调用 **fsync（）** 函数，这可能会影响 Ceph OSD 工作负载。

Ceph 监视器将其数据存储为键值对。使用数据存储可防止恢复 Ceph 监视器通过 Paxos 运行损坏版本，而且它可在一个原子批处理中实现多次修改操作，以及其他优势。

注意

红帽不推荐修改默认数据位置。如果您修改默认位置，请通过在配置文件的 **[mon]** 部分中设置它，使它在 Ceph 监视器间统一。

mon_data

描述

monitor 的数据位置。

类型

字符串

默认

/var/lib/ceph/mon/$cluster-$id

mon_data_size_warn

描述

当监控数据存储达到这个阈值时，Ceph 在集群日志记录中发出 **HEALTH_WARN** 状态。默认值为 15GB。

类型

整数

默认

15*1024*1024*1024*`

mon_data_avail_warn

描述

当 monitor 数据存储的可用磁盘空间低于此百分比时，Ceph 会在集群日志记录中发出 **HEALTH_WARN** 状态。

类型

整数

默认

30

mon_data_avail_crit

描述

当 monitor 数据存储的可用磁盘空间较低或等于这个百分比时，Ceph 在集群日志中会发出 **HEALTH_ERR** 状态。

类型

整数

默认

5

mon_warn_on_cache_pools_without_hit_sets

描述

如果缓存池没有设置 **hit_set_type** paramater，Ceph 会在集群日志记录中发出 **HEALTH_WARN** 状态。如需了解更多详细信息，请参阅池值。

类型

布尔值

默认

True

mon_warn_on_crush_straw_calc_version_zero

描述

如果 CRUSH 的 **straw_calc_version** 为零，Ceph 会在集群日志记录中发出 **HEALTH_WARN** 状态。有关详细信息，请参阅 CRUSH 可调项。

类型

布尔值

默认

True

mon_warn_on_legacy_crush_tunables

描述

如果 CRUSH 可调项太旧（相对于 **mon_min_crush_required_version**而言，Ceph 会在集群日志记录中发出 **HEALTH_WARN** 状态）。

类型

布尔值

默认

True

mon_crush_min_required_version

描述

此设置定义集群所需的最小可调配置集版本。有关详细信息，请参阅 CRUSH 可调项。

类型

字符串

默认

firefly

mon_warn_on_osd_down_out_interval_zero

描述

如果 **mon_osd_down_out_interval** 设置为零，Ceph 在集群日志中会发出 **HEALTH_WARN** 状态，因为设置了 **noout** 标记时 Leader 的行为也类似。管理员通过设置 **noout** 标志来更轻松地对集群进行故障排除。Ceph 发出警告，以确保管理员知道该设置为零。

类型

布尔值

默认

True

mon_cache_target_full_warn_ratio

描述

当处于 **cache_target_full** 和 **target_max_object** 比率之间时，Ceph 会发出警告。

类型

浮点值

默认

0.66

mon_health_data_update_interval

描述

仲裁（以秒为单位）监控器与其对等状态共享其健康状况。负数会禁用运行状况更新。

类型

浮点值

默认

60

mon_health_to_clog

描述

此设置可让 Ceph 定期向集群日志发送健康摘要。

类型

布尔值

默认

True

mon_health_to_clog_tick_interval

描述

monitor 将健康摘要发送到集群日志记录的频率（以秒为单位）。一个非正数代表禁用。如果当前的健康摘要为空，或者与上一次相同，则 monitor 不会将状态发送到集群日志。

类型

整数

默认

3600

mon_health_to_clog_interval

描述

monitor 将健康摘要发送到集群日志记录的频率（以秒为单位）。一个非正数代表禁用。该监控器始终会将摘要发送到集群日志。

类型

整数

默认

60

3.2.5. 存储容量

当 Red Hat Ceph Storage 集群接近其最大容量时（通过 **mon_osd_full_ratio** 参数显示），Ceph 会阻止您写入或读取 Ceph OSD 的安全措施，以防止数据丢失。因此，让一个生产用 Red Hat Ceph Storage 集群方法使其全满比率不是一个好的做法，因为它降低了高可用性。默认全满比率为 **.95** 或 95% 的容量。对于具有多个 OSD 的测试集群来说，这是一个非常积极的设置。

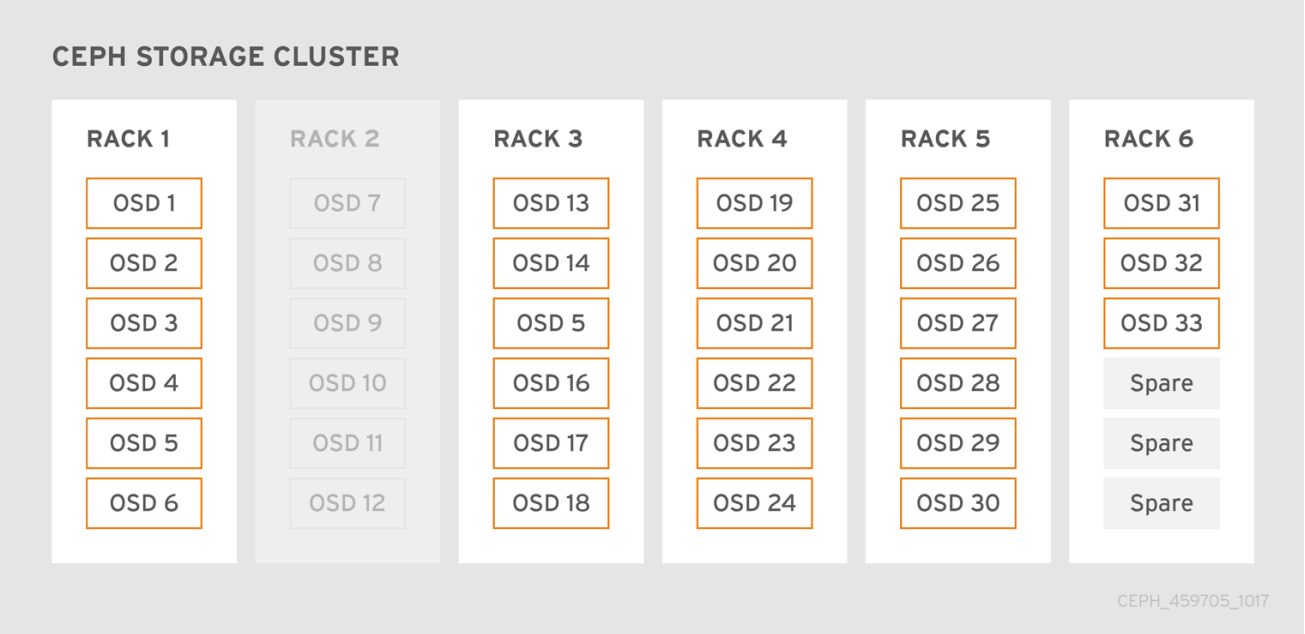
提示

监控集群时，请注意与 **nearfull** 比率相关的警告。这意味着，一些 OSD 故障可能会导致临时服务中断，如果一个或多个 OSD 出现故障。考虑添加更多 OSD 以增加存储容量。

测试集群的常见场景涉及系统管理员从 Red Hat Ceph Storage 集群中删除 Ceph OSD，以观察集群重新平衡。然后，删除另一个 Ceph OSD，直到 Red Hat Ceph Storage 集群最终达到完全的比例并锁定。

红帽建议在一个测试集群中仍然有点的容量规划。通过规划，您可以量化您需要的备用容量来保持高可用性。理想情况下，您要规划一系列 Ceph OSD 失败的情况，集群可以在不立即替换这些 Ceph OSD 的情况下恢复到 **active + clean** 状态。您可以运行状态为 **active + degraded** 的集群，但这对正常操作并不是一个理想的状态。

下图显示了一个简化的 Red Hat Ceph Storage 集群，其中包含每个主机有一个 Ceph OSD 的 33 Ceph 节点，每个 Ceph OSD 守护进程从中读取并写入 3TB 驱动器。因此，这一 exemplary Red Hat Ceph Storage 集群具有最大 99TB 的实际容量。当 **mon osd full ratio** 为 **0.95**，如果 Red Hat Ceph Storage 集群达到 5TB 的容量，集群不允许 Ceph 客户端读取和写入数据。因此，Red Hat Ceph Storage 集群的操作容量为 95 TB，而不是 99 TB。



在这样的集群中，一个或多个 OSD 无法正常使用。较为频繁但合理的方案涉及机架的路由器或电源故障，例如同时导致多个 OSD 下线，例如 OSDs 7-12。在这种情况下，保持集群正常运行并处于 **active + clean** 状态仍会为您带来更大益处，即使这需要在短时间内添加具有额外 OSD 的主机。如果您的容量利用率太高，可能不会丢失数据，但您仍然可能会牺牲数据可用性，同时在故障域内解决集群的容量利用率超过完整的比例。因此，红帽建议至少使用一些最小容量规划。

识别集群的两个值：

- OSD 数量
- 集群的总容量

要确定集群中的 OSD 的平均容量，请将集群的总容量除以集群中的 OSD 数量。考虑将这个数量乘以您希望在正常操作期间同时出现故障的 OSD 数量（相对较小的数）。最后，通过满比例将集群的容量乘以达到最大操作容量。然后，从 OSD 中减去您希望无法达到合理的全满比率的 OSD 的数据量。重复处理数量较高的 OSD 故障（例如，一个 OSD 机架），以达到接近的全满比率的合理数量。

```
[global]
...
mon_osd_full_ratio = .80
mon_osd_nearfull_ratio = .70
```

mon_osd_full_ratio

描述

在 OSD 被视为 **full** 之前使用的磁盘空间百分比。

类型

浮点值：

默认

.95

mon_osd_nearfull_ratio

描述

在 OSD 视为 **nearfull** 之前使用的磁盘空间百分比。

类型

浮点值

默认

.85

提示

如果某些 OSD 接近满，但其他 OSD 具有容量量，则可能对 **nearfull** OSD 的 CRUSH 权重出现问题。

3.2.6. heartbeat

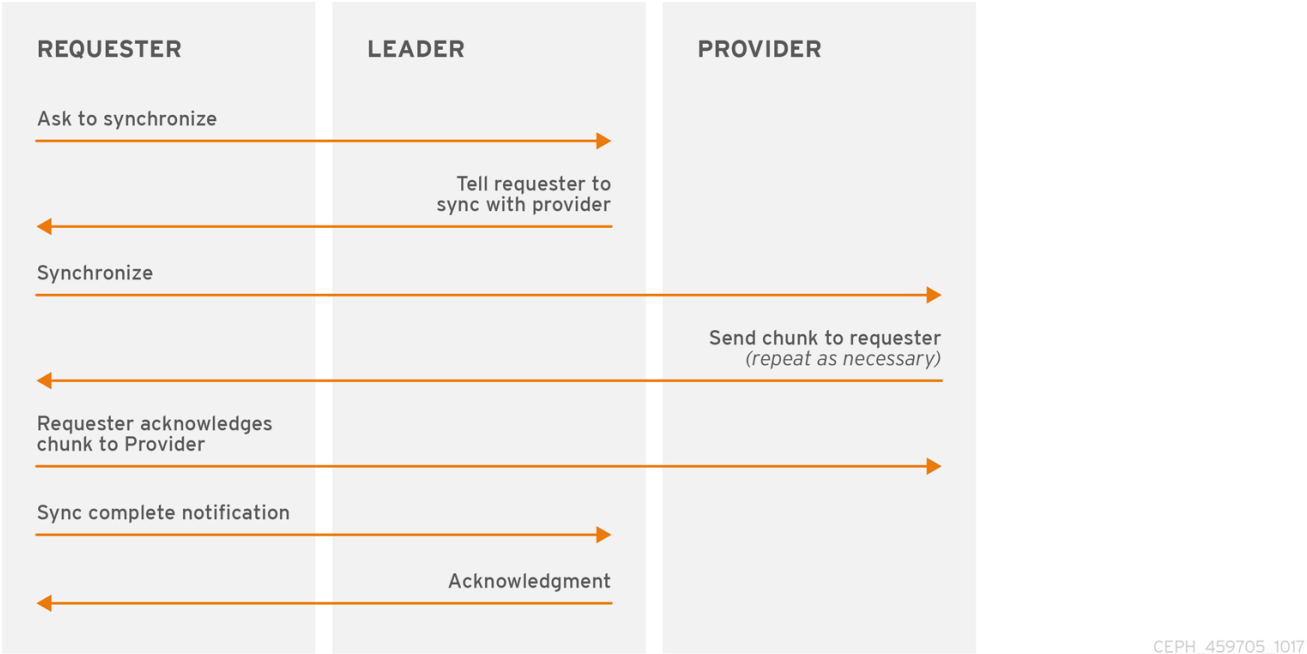
Ceph 监视器通过要求来自每个 OSD 的报告以及从 OSD 接收关于其邻居 OSD 状态的报告来了解集群。Ceph 为 monitor 和 OSD 之间的交互提供了合理的默认设置，但您可以根据需要修改它们。

3.2.7. monitor Store Synchronization

当您使用多个监控器运行生产集群时，每个 monitor 会检查邻居监视器是否有最新版本的 cluster map。例如，一个邻居监视器中的映射，其一个或多个 epoch 号高于即时监视器映射中当前 epoch 的数值。定期，集群中的一个监视器可能位于其他 monitor 后，它必须离开仲裁，同步来检索有关集群的最新信息，然后重新加入仲裁。出于同步目的，监视器可以假定以下三个角色之一：

- **Leader** ：Leader 是达到集群映射的最新 Paxos 版本的第一个 moitor。
- **Provider** ：Provider 是一个具有集群映射的最新版本的 monitor，但不是第一个。
- **Requester:** 请求者是一个监控器，它已落后于领导，必须同步来检索集群的最新信息，然后才能重新加入仲裁。

这些角色使领导能够将同步任务委派给提供程序，从而防止同步请求过载，并提高性能。在下图中，请求者已了解到它已位于其他 monitor 后。请求者要求领导要同步，并且领导者告诉请求者与提供程序同步。



当新 monitor 加入集群时，才会发生同步。在运行时操作期间，监控器可以在不同时间接收集群映射的的更新。这意味着领导和提供商角色可以从一个监控器迁移到另一个监视器。例如在同步时发生这种情况，例如，提供商落于领导，提供商可以与请求者终止同步。

同步完成后，Ceph 需要在集群中修剪。修剪要求放置组处于 **active + clean** 状态。

mon_sync_trim_timeout

描述, **类型**

双

默认

30.0

mon_sync_heartbeat_timeout

描述, **类型**

双

默认

30.0

mon_sync_heartbeat_interval

描述, **类型**

双

默认

5.0

mon_sync_backoff_timeout

描述, **类型**

双

默认

30.0

mon_sync_timeout

描述

在启动和引导前，监控器将等待其同步提供程序中下一次更新消息的秒数。

类型

双

默认

30.0

mon_sync_max_retries

描述, **类型**

整数

默认

5

mon_sync_max_payload_size

描述

同步有效负载（以字节为单位）的最大大小。

类型

32 位整数

默认

1045676

paxos_max_join_drift

描述

必须首先同步监控数据存储前，最大的 Paxos 迭代。当 monitor 发现其对等点比其太超前时，它将首先与数据存储同步，然后再继续。

类型

整数

默认

10

paxos_stash_full_interval

描述

（在 commits 时）stash PaxosService 状态的完整副本的频率。当前此设置仅影响 **mds**、**mon**、**auth** 和 **mgr** PaxosServices。

类型

整数

默认

25

paxos_propose_interval

描述

收集这个时间更新，然后再执行映射更新。

类型

双

默认

1.0

paxos_min

描述

要保留的最小 paxos 状态数量

类型

整数

默认

500

paxos_min_wait

描述

在不活跃的一段时间后，收集更新的最小时间。

类型

双

默认

0.05

paxos_trim_min

描述

在修剪前可以容忍的额外提议数

类型

整数

默认

250

paxos_trim_max

描述

一次要修剪的最大额外提议数

类型

整数

默认

500

paxos_service_trim_min

描述

触发修剪的最小版本数量（0 禁用它）

类型

整数

默认

250

paxos_service_trim_max

描述

单一提案期间要修剪的最大版本量（0 代表禁用它）

类型

整数

默认

500

mon_max_log_epochs

描述

单个提议期间要修剪的最大日志时期量

类型

整数

默认

500

mon_max_pgmap_epochs

描述

单个建议期间要修剪的最大 pgmap epoch 数量

类型

整数

默认

500

mon_mds_force_trim_to

描述

强制 monitor 在这点上修剪 mdsmaps（0 代表禁用。这个设置比较危险，请谨慎使用）

类型

整数

默认

0

mon_osd_force_trim_to

描述

强制 monitor 在这点上修剪 osdmaps，即使指定 epoch 中没有清理 PG（0 则禁用它。dangerous 则谨慎使用）

类型

整数

默认

0

mon_osd_cache_size

描述

osdmaps 缓存的大小，不依赖于底层存储的缓存

类型

整数

默认

10

mon_election_timeout

描述

在选举代理上，让所有 ACK 的最大等待时间（以秒为单位）。

类型

浮点值

默认

5

mon_lease

描述

监控版本中租期的长度（以秒为单位）。

类型

浮点值

默认

5

mon_lease_renew_interval_factor

描述

mon lease * **mon lease renew interval factor** 将是领导机更新其他 monitor 的租期的时间间隔。因素应小于 **1.0**。

类型

浮点值

默认

0.6

mon_lease_ack_timeout_factor

描述

领导机将会等待 **mon lease** * **mon lease ack timeout factor** 的时间来等待供应商确认租期的扩展。

类型

浮点值

默认

2.0

mon_accept_timeout_factor

描述

领导机将等待 **mon lease** * **mon accept timeout** 来等待 Requester 接受 Paxos 更新。它还在 Paxos 恢复阶段用于类似目的。

类型

浮点值

默认

2.0

mon_min_osdmap_epochs

描述

始终保留的最小 OSD map epoch 数。

类型

32 位整数

默认

500

mon_max_pgmap_epochs

描述

监视器应保留的最大 PG 映射 epoch 数。

类型

32 位整数

默认

500

mon_max_log_epochs

描述

监视器应保留的最大日志 epoch 数。

类型

32 位整数

默认

500

3.2.8. clock

Ceph 守护进程将关键消息传递到彼此，这必须在守护进程到达超时阈值前进行处理。如果 Ceph 监视器中的时钟没有同步，它可以导致一些异常。例如：

- 忽略收到的消息的守护进程（例如，时间戳过时）。
- 当没有收到信息时，超时会马上或晚末触发。

详情请参阅 Monitor Store Synchronization。

提示

在 Ceph 监控主机上安装 NTP，以确保监控集群与时钟同步运行。

时钟偏移可能仍然可以通过 NTP 发现，即使差异尚未有害。Ceph 时钟偏移和时钟偏移警告可能会触发，即使 NTP 维护合理的同步级别。在这种情况下，可以容忍时钟偏移。但是，很多因素，如工作负载、网络延迟、配置为默认超时，monitor 存储同步 设置可能会影响可接受的时钟偏移级别，而不会损害 Paxos 保障。

Ceph 提供以下可调选项，供您查找可接受值。

clock_offset

描述

系统时钟偏移量。详情请查看 **Clock.cc**。

类型

双

默认

0

mon_tick_interval

描述

监视器的空循环间隔（以秒为单位）。

类型

32 位整数

默认

5

mon_clock_drift_allowed

描述

监视器之间允许的时钟偏移（以秒为单位）。

类型

浮点值

默认

.050

mon_clock_drift_warn_backoff

描述

用于时钟偏移警告的指数 backoff.

类型

浮点值

默认

5

mon_timecheck_interval

描述

检查领导的时间间隔（时钟偏移检查）。

类型

浮点值

默认

300.0

mon_timecheck_skew_interval

描述

在领导存在偏差时（以秒为单位）的时间检查间隔（时钟偏移检查）。

类型

浮点值

默认

30.0

3.2.9. 客户端

mon_client_hunt_interval

描述

客户端每 **N** 秒尝试一次新监控器，直到连接建立为止。

类型

双

默认

3.0

mon_client_ping_interval

描述

客户端将每 **N** 秒 ping 监控器。

类型

双

默认

10.0

mon_client_max_log_entries_per_message

描述

监控的每个客户端将生成的最大日志条目数。

类型

整数

默认

1000

mon_client_bytes

描述

内存中允许的客户端消息数据量（以字节为单位）。

类型

64 位 Unsigned 整数

默认

100ul << 20

3.3. 其它

mon_max_osd

描述

集群中允许的最大 OSD 数量。

类型

32 位整数

默认

10000

mon_globalid_prealloc

描述

为集群中的客户端和守护进程预先分配的全局 ID 数量。

类型

32 位整数

默认

100

mon_sync_fs_threshold

描述

在编写指定对象数量时与文件系统同步。将它设置为 **0** 以禁用它。

类型

32 位整数

默认

5

mon_subscribe_interval

描述

订阅的刷新间隔（以秒为单位）。订阅机制允许获取集群映射和日志信息。

类型

双

默认

300

mon_stat_smooth_intervals

描述

Ceph 将在最后的 **N** PG 映射中平稳统计信息。

类型

整数

默认

2

mon_probe_timeout

描述

监视器在 bootstrap 前等待的对等点的秒数。

类型

双

默认

2.0

mon_daemon_bytes

描述

存储服务器和 OSD 消息的消息内存大写（以字节为单位）。

类型

64 位 Unsigned 整数

默认

400ul << 20

mon_max_log_entries_per_event

描述

每个事件的最大日志条目数。

类型

整数

默认

4096

mon_osd_prime_pg_temp

描述

当 OSD 返回到集群中时，启用或禁用带有之前 OSD 的 PGMap。当进行 **true** 设置时，客户端将继续使用前面的 OSD，直到 OSD 中新作为 PG 对等。

类型

布尔值

默认

true

mon_osd_prime_pg_temp_max_time

描述

当 OSD 返回到集群时，monitor 应该花费多少时间（以秒为单位）尝试推断 PGMap。

类型

浮点值

默认

0.5

mon_osd_prime_pg_temp_max_time_estimate

描述

在我们并行控制所有 PG 前，每个 PG 花费的最大估算时间。

类型

浮点值

默认

0.25

mon_osd_allow_primary_affinity

描述

允许在 osdmap 中设置 **primary_affinity**。

类型

布尔值

默认

False

mon_osd_pool_ec_fast_read

描述

是否启用对池的快速读取。如果在创建时没有指定 **fast_read**，它将用作新创建的池的默认设置。

类型

布尔值

默认

False

mon_mds_skip_sanity

描述

跳过 FSMap 上的安全断言（以防出现错误时我们想继续任何方式）。如果 FSMap sanity 检查失败，则 monitor 会终止，但您可以通过启用此选项来禁用它。

类型

布尔值

默认

False

mon_max_mdsmap_epochs

描述

单一建议期间要修剪的最大 mdsmap epoch 数。

类型

整数

默认

500

mon_config_key_max_entry_size

描述

config-key 条目的最大数量（以字节为单位）

类型

整数

默认

4096

mon_scrub_interval

描述

monitor 将存储清理其存储的频率（以秒为单位）与所有存储的密钥的计算方式进行比较。

类型

整数

默认

3600*24

mon_scrub_max_keys

描述

每次清理的最大键数。

类型

整数

默认

100

mon_compact_on_start

描述

在 **ceph-mon** 启动时，紧凑用作 Ceph Monitor 存储的数据库。手动压缩有助于缩小 monitor 数据库，并在常规压缩失败时提高它的性能。

类型

布尔值

默认

False

mon_compact_on_bootstrap

描述

紧凑用作 Ceph Monitor 存储在 bootstrap 上的数据库。monitor 会相互探测到 bootstrap 后创建仲裁。如果在加入仲裁前超时，它将再次启动并引导自身。

类型

布尔值

默认

False

mon_compact_on_trim

描述

在我们修剪旧状态时，压缩特定的前缀（包括 paxos）。

类型

布尔值

默认

True

mon_cpu_threads

描述

在监控中执行 CPU 密集型工作的线程数量。

类型

布尔值

默认

True

mon_osd_mapping_pgs_per_chunk

描述

我们以块的形式计算从放置组到 OSD 的 map。这个选项指定每个块的放置组数量。

类型

整数

默认

4096

mon_osd_max_split_count

描述

每个 "involved" OSD 的最大 PG 数量，以便进行拆分创建。当增加一个池的 **pg_num** 时，将把 PG 拆分到为这个池的所有 OSD 上。我们希望避免在 PG 分上的极倍。

类型

整数

默认

300

mon_session_timeout

描述

监控将终止非活动会话在此限制时处于闲置状态。

类型

整数

默认

300

rados_mon_op_timeout

描述

在从 RADOS 操作返回错误之前，RADOS 从 Ceph Monitor 等待响应的秒数。值为 0 表示没有限制。

类型

双

默认

0

第 4 章 CEPHX 配置参考

cephx 协议默认启用。加密身份验证具有一些计算成本，尽管它们通常很低。如果网络环境连接客户端和服务器主机非常安全，并且您无法承担身份验证，您可以禁用它。但是，红帽建议使用身份验证。

注意

如果您禁用身份验证，您将面临中间攻击的风险，改变客户端和服务器信息，这可能会导致严重的安全问题。

4.1. MANUAL（手动）

手动部署集群时，您必须手动引导 monitor 并创建 **client.admin** 用户和密钥环。要手动部署 Ceph， 请参阅知识库文章。监控 bootstrap 的步骤是使用 Chef、Puppet、Juju 等第三方部署工具时必须执行的逻辑步骤。

4.2. 启用和禁用 CEPHX

启用 Cephx 要求您已部署 monitor 和 OSD 的密钥。如果您简单地在 / off 上切换 Cephx，则不必重复引导过程。

4.2.1. 启用 Cephx

启用 **cephx** 后，Ceph 将在默认搜索路径中查找密钥环，其中包括 **/etc/ceph/$cluster.$name.keyring**。您可以通过在 Ceph 配置文件的 **[global]** 部分添加 **keyring** 选项来覆盖该位置，但不建议这样做。

执行以下步骤，在禁用身份验证的集群中启用 **cephx**。如果您或部署实用程序生成了密钥，您可以跳过与生成密钥相关的步骤。

1. 创建 **client.admin** 密钥，并为您的客户端主机保存密钥副本：

```
ceph auth get-or-create client.admin mon 'allow *' osd 'allow *' -o
/etc/ceph/ceph.client.admin.keyring
```

警告

这将擦除任何现有 **/etc/ceph/client.admin.keyring** 文件的内容。如果部署工具已为您完成，则不要执行此步骤。

2. 为 monitor 集群创建密钥环，并生成监控器 secret 密钥：

```
ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --cap mon 'allow *'
```

3. 将 monitor keyring 复制到每个 monitor mon **数据目录中的 ceph.mon.keyring** 文件。例如，要将其复制到 cluster **ceph** 中的 **mon.a** 中，请使用：

```
cp /tmp/ceph.mon.keyring /var/lib/ceph/mon/ceph-a/keyring
```

4. 为每个 OSD 生成 secret 密钥，其中 **{$id}** 是 OSD 号：

```
ceph auth get-or-create osd.{$id} mon 'allow rwx' osd 'allow *' -o /var/lib/ceph/osd/ceph-{$id}/keyring
```

5. 默认情况下启用 **cephx** 身份验证协议。

注意

如果以前通过将身份验证选项设置为 **none** 来禁用 **cephx** 身份验证协议，那么请在 Ceph 配置文件 **[global]** 部分 (**/etc/ceph/ceph.conf**) 下删除 [global] 部分下的以下行，则会重新启用 **cephx** 身份验证协议：

```
auth_cluster_required = none
auth_service_required = none
auth_client_required = none
```

6. 启动或重启 Ceph 集群。

重要

启用 **cephx** 需要停机，因为集群需要完全重启，或者在禁用客户端 I/O 时将其关闭并启动。

这些标记需要在重启或关闭存储集群前设置：

```
# ceph osd set noout
# ceph osd set norecover
# ceph osd set norebalance
# ceph osd set nobackfill
# ceph osd set nodown
# ceph osd set pause
```

启用 **cephx** 后，所有 PG 都活跃且干净，取消设置标记：

```
# ceph osd unset noout
# ceph osd unset norecover
# ceph osd unset norebalance
# ceph osd unset nobackfill
# ceph osd unset nodown
# ceph osd unset pause
```

4.2.2. 禁用 Cephx

以下流程描述了如何禁用 Cephx。如果您的集群环境相对安全，您可以降低运行身份验证的计算费用。红帽建议启用身份验证。但是，在设置或故障排除过程中可能会更容易地禁用身份验证。

1. 通过在 Ceph 配置文件的 **[global]** 部分设置以下选项来禁用 **cephx** 身份验证：

```
auth_cluster_required = none
auth_service_required = none
auth_client_required = none
```

2. 启动或重启 Ceph 集群。

4.3. 配置设置

4.3.1. 启用

auth_cluster_required

描述

如果启用，Red Hat Ceph Storage 集群守护进程（即 **ceph-mon** 和 **ceph-osd**）必须相互进行身份验证。有效的设置是 **cephx** 或 **none**。

类型

字符串

必需

否

默认

cephx。

auth_service_required

描述

如果启用，Red Hat Ceph Storage 集群守护进程需要 Ceph 客户端与 Red Hat Ceph Storage 集群进行身份验证，以便能访问 Ceph 服务。有效的设置是 **cephx** 或 **none**。

类型

字符串

必需

否

默认

cephx。

auth_client_required

描述

如果启用，Ceph 客户端需要 Red Hat Ceph Storage 集群与 Ceph 客户端进行身份验证。有效的设置是 **cephx** 或 **none**。

类型

字符串

必需

否

默认

cephx。

4.3.2. Keys

当您运行 Ceph 并启用了身份验证时，**ceph** 管理命令和 Ceph 客户端需要身份验证密钥来访问 Ceph 存储集群。

向 **ceph** 管理命令和客户端提供这些密钥的最常见方式是在 **/etc/ceph/** 目录下包含 Ceph 密钥环。文件名通常是 **ceph.client.admin.keyring** 或 **$cluster.client.admin.keyring**。如果您在 **/etc/ceph/** 目录下包含密钥环，则不需要在 Ceph 配置文件中指定 **keyring** 条目。

红帽建议将 Red Hat Ceph Storage 集群 keyring 文件复制到您要运行管理命令的节点，因为它包含 **client.admin** 密钥。要做到这一点，以 **root** 用户身份执行以下命令：

```
# scp <user>@<hostname>:/etc/ceph/ceph.client.admin.keyring /etc/ceph/ceph.client.admin.keyring
```

将 **<user>** 替换为主机上的用户名，使用 **client.admin** 键，将 **<hostname>** 替换为该主机的主机名。

注意

确保 **ceph.keyring** 文件已在客户端计算机上设置适当的权限。

您可以使用 **key** 设置在 Ceph 配置文件中（不推荐）指定密钥本身，或使用 **keyfile** 设置来指定到密钥文件的路径。

keyring

描述

keyring 文件的路径。

类型

字符串

必需

否

默认

/etc/ceph/$cluster.$name.keyring,/etc/ceph/$cluster.keyring,/etc/ceph/keyring,/etc/ceph/keyring.bin

keyfile

描述

密钥文件的路径（即仅包含密钥的文件）。

类型

字符串

必需

否

默认

无

key

描述

密钥（即密钥本身，它是一个文本字符串）。不建议。

类型

字符串

必需

否

默认

无

4.3.3. 守护进程密钥环

管理用户或部署工具可能会像生成用户密钥环一样生成守护进程密钥环。默认情况下，Ceph 将守护进程密钥环存储在其数据目录中。默认密钥环位置以及守护进程正常工作所需的功能如下所示。

ceph-mon

位置

$mon_data/keyring

功能

mon 'allow *'

ceph-osd

位置

$osd_data/keyring

功能

mon 'allow profile osd' osd 'allow *'

radosgw

位置

$rgw_data/keyring

功能

mon 'allow rwx' osd 'allow rwx'

注意

监视器密钥环（即 **mon**）包含密钥但没有功能，不是集群 **auth** 数据库的一部分。

守护进程数据目录位置默认为表单的目录：

```
/var/lib/ceph/$type/$cluster-$id
```

例如，**osd.12** 是：

```
/var/lib/ceph/osd/ceph-12
```

您可以覆盖这些位置，但不推荐进行。

4.3.4. 签名

红帽建议 Ceph 使用为该初始身份验证设置的会话密钥验证实体之间的所有持续消息。

与 Ceph 身份验证的其他部分一样，Ceph 提供精细的控制，以便您可以启用或禁用客户端和 Ceph 之间的服务消息的签名，您可以为 Ceph 守护进程之间的消息启用或禁用签名。

cephx_require_signatures

描述

如果设置为 **true**，Ceph 需要在 Ceph 客户端和 Red Hat Ceph Storage 集群间的所有消息流量上签名，并且守护进程之间由 Red Hat Ceph Storage 集群组成。

类型

布尔值

必需

否

默认

false

cephx_cluster_require_signatures

描述

如果设置为 **true**，Ceph 需要 Ceph 守护进程间的所有消息流量签名，由 Red Hat Ceph Storage 集群组成。

类型

布尔值

必需

否

默认

false

cephx_service_require_signatures

描述

如果设置为 **true**，Ceph 需要 Ceph 客户端和 Red Hat Ceph Storage 集群间的所有消息流量签名。

类型

布尔值

必需

否

默认

false

cephx_sign_messages

描述

如果 Ceph 版本支持消息签名，Ceph 会对所有消息进行签名，使其无法欺骗。

类型

布尔值

默认

true

注意

Ceph 内核模块尚不支持签名。

4.3.5. 实时到实时

auth_service_ticket_ttl

描述

当 Red Hat Ceph Storage 集群向 Ceph 客户端发送一个 ticket 进行身份验证时，集群会为 ticket 分配一个生存时间。

类型

双

默认

60*60

第 5 章 池、PG 和 CRUSH 配置参考

当您创建池并为池设置放置组数量时，Ceph 在不特别覆盖默认值时会使用默认值。红帽建议覆盖一些默认值。特别是，设置池的副本大小并覆盖默认放置组数量。您可以在运行池命令时设置这些值。您还可以通过在 Ceph 配置文件的 **[global]** 部分添加新项来覆盖默认值。

```
[global]

# By default, Ceph makes 3 replicas of objects. If you want to set 4
# copies of an object as the default value--a primary copy and three replica
# copies--reset the default values as shown in 'osd pool default size'.
# If you want to allow Ceph to write a lesser number of copies in a degraded
# state, set 'osd pool default min size' to a number less than the
# 'osd pool default size' value.

osd_pool_default_size = 4  # Write an object 4 times.
osd_pool_default_min_size = 1 # Allow writing one copy in a degraded state.

# Ensure you have a realistic number of placement groups. We recommend
# approximately 100 per OSD. E.g., total number of OSDs multiplied by 100
# divided by the number of replicas (i.e., osd pool default size). So for
# 10 OSDs and osd pool default size = 4, we'd recommend approximately
# (100 * 10) / 4 = 250.

osd_pool_default_pg_num = 250
osd_pool_default_pgp_num = 250
```

5.1. 设置

mon_allow_pool_delete

描述

允许 monitor 删除池。在 RHCS 3 及更新的版本中，监控器无法默认删除池，以保护数据。

类型

布尔值

默认

false

mon_max_pool_pg_num

描述

每个池的最大放置组数量。

类型

整数

默认

65536

mon_pg_create_interval

描述

在同一 Ceph OSD 守护进程中创建 PG 间隔的秒数。

类型

浮点值

默认

30.0

mon_pg_stuck_threshold

描述

PG 被视为卡住的秒数。

类型

32 位整数

默认

300

mon_pg_min_inactive

描述

如果处于非活跃状态的时间超过 **mon_pg_stuck_threshold** 的 PG 数量超过这个设置，Ceph 会在集群日志记录中记录一个 **HEALTH_ERR** 状态。默认设置为一个 PG。非正数代表禁用此设置。

类型

整数

默认

1

mon_pg_warn_min_per_osd

描述

如果集群中每个 OSD 的平均 PG 数量小于此设置，Ceph 会在集群日志记录中发出 **HEALTH_WARN** 状态。非正数代表禁用此设置。

类型

整数

默认

30

mon_pg_warn_max_per_osd

描述

如果集群中每个 OSD 的平均 PG 数量大于此设置，Ceph 会在集群日志记录中发出 **HEALTH_WARN** 状态。非正数代表禁用此设置。

类型

整数

默认

300

mon_pg_warn_min_objects

描述

如果集群中的对象总数低于这个数字，则不发出警告。

类型

整数

默认

1000

mon_pg_warn_min_pool_objects

描述

不要在对象数低于这个值的池中警告。

类型

整数

默认

1000

mon_pg_check_down_all_threshold

描述

down OSD 的阈值（百分比），在超过这个值时 Ceph 检查所有 PG 以确保它们没有处于 stuck 或 stale 状态。

类型

浮点值

默认

0.5

mon_pg_warn_max_object_skew

描述

如果池中对象平均数量大于 **mon pg warn max object skew** 乘以所有池的平均数量，则 Ceph 在集群日志中发出 **HEALTH_WARN** 状态。非正数代表禁用此设置。

类型

浮点值

默认

10

mon_delta_reset_interval

描述

在 Ceph 将 PG 增量重置为零之前需要经过的不活跃的秒数。Ceph 追踪各个池的已用空间增量，以帮助管理员评估恢复和性能的进度。

类型

整数

默认

10

mon_osd_max_op_age

描述

在发出 **HEALTH_WARN** 状态前，要完成的操作的最大期限（以秒为单位）。

类型

浮点值

默认

32.0

osd_pg_bits

描述

每个 Ceph OSD 守护进程的放置组位。

类型

32 位整数

默认

6

osd_pgp_bits

描述

用于放置目的 (PGP) 的每个 Ceph OSD 守护进程用于放置组的位数。

类型

32 位整数

默认

6

osd_crush_chooseleaf_type

描述

在 CRUSH 规则中，用于 **chooseleaf** 的 bucket 类型。使用等级排名，而不是名称。

类型

32 位整数

默认

1.通常，含有一个或多个 Ceph OSD 守护进程的主机。

osd_pool_default_crush_replicated_ruleset

描述

创建复制池时要使用的默认 CRUSH 规则集。

类型

8 位整数

默认

0

osd_pool_erasure_code_stripe_unit

描述

为纠删代码池设置对象条带的块的默认大小，以字节为单位。每个大小为 S 的对象将存储为 N 个条带，每个数据块都会有 **stripe unit** 个字节。**N * stripe unit** 字节的每个条带都将单独编码/解码。此选项可以通过 profile 中的 **stripe_unit** 设置覆盖。

类型

Unsigned 32 位整数

默认

4096

osd_pool_default_size

描述

设置池中对象的副本数量。默认值与 **ceph osd pool set {pool-name} size {size}** 相同。

类型

32 位整数

默认

3

osd_pool_default_min_size

描述

为池中对象设置最少写入副本数，以确认对客户端的写操作。如果至少不满足，Ceph 不会确认对客户端的写入。此设置确保以 **degraded** 模式运行时有最小副本数。

类型

32 位整数

默认

0，表示没有特定最小值。如果为 **0**，最小为 **size - (size / 2)** 。

osd_pool_default_pg_num

描述

池的默认放置组数量。默认值为 **pg_num** 和 **mkpool**。

类型

32 位整数

默认

8

osd_pool_default_pgp_num

描述

池放置的默认放置组数量。默认值为 **pgp_num** 和 **mkpool**。PG 和 PGP 应该相等（目前）。

类型

32 位整数

默认

8

osd_pool_default_flags

描述

新池的默认标记。

类型

32 位整数

默认

0

osd_max_pgls

描述

要列出的最大放置组数量。请求大量客户端可以连接 Ceph OSD 守护进程。

类型

unsigned 64 位整数

默认

1024

备注

默认应该是正常的。

osd_min_pg_log_entries

描述

修剪日志文件时要维护的最小放置组日志数量。

类型

32 位整数（Unsigned）

默认

1000

osd_default_data_pool_replay_window

描述

OSD 等待客户端重播请求的时间（以秒为单位）。

类型

32 位整数

默认

45

第 6 章 OSD 配置参考

您可以在 Ceph 配置文件中配置 Ceph OSD，但 Ceph OSD 可以使用默认值和非常少的配置。最小的 Ceph OSD 配置会设置 **osd 日志大小** 和 **osd 主机** 选项，并将默认值用于几乎所有选项。

Ceph OSD 以增量方式标识，从以下惯例以 **0** 开始：

```
osd.0
osd.1
osd.2
```

在配置文件中，您可以通过将配置设置添加到配置文件的 **[osd]** 部分，指定集群中的所有 Ceph OSD 的设置。要直接将设置添加到特定的 Ceph OSD（如 **osd host**），请在仅针对该 OSD 在 Ceph 配置文件中输入它。例如：

```
[osd]
osd journal size = 1024

[osd.0]
osd host = osd-host-a

[osd.1]
osd host = osd-host-b
```

6.1. 常规设置

以下设置提供 Ceph OSD 的 ID，并决定到数据和日志的路径。Ceph 部署脚本通常自动生成 UUID。

重要

红帽不推荐更改数据或日志的默认路径，因为以后对 Ceph 进行故障排除会更有问题。

日志大小应至少两倍预期驱动器产品的速度乘以 **filestore max sync interval** 选项的值。但是，最常见的做法是对日志驱动器进行分区（通常是 SSD），然后挂载它，以便 Ceph 将整个分区用于日志。

osd_uuid

描述

Ceph OSD 的通用唯一识别符 (UUID)。

类型

UUID

默认

UUID。

备注

osd uuid 应用到单个 Ceph OSD。 **fsid** 应用到整个集群。

osd_data

描述

OSD 数据路径。在部署 Ceph 时，您必须创建该目录。在此挂载点上挂载 OSD 数据的驱动器。红帽不推荐修改默认设置。

类型

字符串

默认

/var/lib/ceph/osd/$cluster-$id

osd_max_write_size

描述

以 MB 为单位的最大写入大小。

类型

32 位整数

默认

90

osd_client_message_size_cap

描述

内存中允许的最大客户端数据消息。

类型

64 位 Unsigned 整数

默认

500MB 默认。**500*1024L*1024L**

osd_class_dir

描述

RADOS 类插件的类路径。

类型

字符串

默认

$libdir/rados-classes

6.2. 日志设置

默认情况下，Ceph 预期将存储 Ceph OSD 的日志并具有以下路径：

```
/var/lib/ceph/osd/$cluster-$id/journal
```

如果不进行性能优化，Ceph 会将日志存储在与 Ceph OSD 的数据相同的磁盘上。对性能进行优化的 Ceph OSD 可以使用单独的磁盘来存储日志数据，例如，使用固态硬盘提供高性能日志。

日志大小应该找到 **文件存储最大同步间隔** 的产品以及预期吞吐量，并将产品分为两 (2) ：

```
osd journal size = <2 * (expected throughput * filestore max sync interval)>
```

预期的吞吐量数应包含预期的磁盘吞吐量（即，可持续的数据传输率），以及网络吞吐量。例如，7200 RPM 磁盘可能大约有 100 MB/s。使用磁盘和网络吞吐量的 **min()** 应该提供合理的预期吞吐量。有些用户只是以 10GB 的日志大小启动。例如：

-

```
osd journal size = 10000
```

警告

为您的 OSD 正确调整日志大小非常重要。使用小日志会导致在 OSD 出现故障时进行较慢的恢复。恢复线程数量必须降低，以便进行稳定的恢复，方法是使日志在可接受的级别保持压力。另外，提交对文件存储的事务会较慢，如果排队的事务大小大于日志大小，则文件存储可能会挂起。

osd_journal

描述

OSD 日志的路径。这可以是到文件或块设备（比如 SSD 的分区）的路径。如果是一个文件，则必须创建包含该目录的目录。我们建议您使用独立于 **osd data** 驱动器的驱动器。

类型

字符串

默认

/var/lib/ceph/osd/$cluster-$id/journal

osd_journal_size

描述

日志的大小（以 MB 为单位）。如果是 0，则日志是一个块设备，则会使用整个块设备。如果日志是块设备，并且使用整个块设备，这将会被忽略。

类型

32 位整数

默认

5120

推荐的

从 1GB 开始。大小应至少为产品的速度乘以 **filestore max sync interval** 的值的两倍。

6.3. 清理

除了生成多个对象副本外，Ceph 还通过清理放置组来确保数据完整性。Ceph 清理与对象存储层上的 **fsck** 命令类似。

对于每个放置组，Ceph 都会生成所有对象的目录，并比较每个主对象及其副本，以确保缺少对象或不匹配。

轻度清理 (每日) 会检查对象大小和属性。深度刮除（每周）读取数据并使用 checksum 来确保数据完整性。

清理对于保持数据完整性非常重要，但可能会降低性能。调整以下设置以增加或减少清理操作。

osd_max_scrubs

描述

Ceph OSD 同步清理操作的最大数量。

类型

32 位整数

默认

1

osd_scrub_thread_timeout

描述

刮除线程超时前需要经过的最大时间（以秒为单位）。

类型

32 位整数

默认

60

osd_scrub_finalize_thread_timeout

描述

刮除完成线程超时前需要经过的最大时间（以秒为单位）。

类型

32 位整数

默认

60*10

osd_scrub_begin_hour

描述

轻量或深度刮除可以开始的最早的小时。它用于 **osd scrub end hour** 参数，用于定义清理时间窗，并允许将清理限制为非高峰小时。此设置使用一个整数来指定 24 小时周期内的小时，其中 **0** 代表从 12:01 a.m. 到 1:00 a.m.，13 代表从 1:01 p.m. 到 2:00 p.m。

类型

32 位整数

默认

0 代表 12:01 到 1:00。

osd_scrub_end_hour

描述

轻量或深度刮除可以开始的最完的小时。它用于 **osd scrub start hour** 参数来定义清理时间窗，并允许将清理限制为非高峰小时。此设置使用一个整数来指定 24 小时周期内的小时，其中 **0** 代表从 12:01 a.m. 到 1:00 a.m.，13 代表从 1:01 p.m. 到 2:00 p.m。**结束** 小时必须大于 **开始** 小时。

类型

32 位整数

默认

24 for 11:01 p.m. to 12:00 a.

osd_scrub_load_threshold

描述

最大负载。当系统负载（由 **getloadavg ()** 功能定义）超过这个数值时，Ceph 不会进行刮除。默认为 **0.5**。

类型

浮点值

默认

0.5

osd_scrub_min_interval

描述

当 Red Hat Ceph Storage 集群负载较低时，清理 Ceph OSD 的最小间隔（以秒为单位）。

类型

浮点值

默认

每天一次。**60*60*24**

osd_scrub_max_interval

描述

清理 Ceph OSD 所需负载时的最长时间（以秒为单位）。

类型

浮点值

默认

每周一次。**7*60*60*24**

osd_scrub_interval_randomize_ratio

描述

使用这个比率，在 **osd scrub min interval** 和 **osd scrub max interval** 间随机化调度的刮除。

类型

浮点值

默认

0.5.

mon_warn_not_scrubbed

描述

osd_scrub_interval 后的秒数，以警告任何未清理的 PG。

类型

整数

默认

0 (无警告) 。

osd_scrub_chunk_min

描述

对象存储被分区为以哈希界限结尾的块。对于块清理，Ceph 一次清理对象一个块，且对这个块的写入被阻止。**osd scrub chunk min** 设置表示要清理的最小块数量。

类型

32 位整数

默认

5

osd_scrub_chunk_max

描述

清理的最大块数量。

类型

32 位整数

默认

25

osd_scrub_sleep

描述

深度清理操作之间休眠的时间。

类型

浮点值

默认

0 (或关闭) 。

osd_scrub_during_recovery

描述

允许在恢复期间进行清理。

类型

Bool

默认

false

osd_scrub_invalid_stats

描述

强制执行额外的清理，以修复标记为无效统计数据。

类型

Bool

默认

true

osd_scrub_priority

描述

控制清理操作与客户端 I/O 的队列优先级。

类型

Unsigned 32 位整数

默认

5

osd_scrub_cost

描述

以 MB 为单位清理操作的成本，用于队列调度目的。

类型

Unsigned 32 位整数

默认

50 << 20

osd_deep_scrub_interval

描述

深度清理的时间间隔，即完全读取所有数据。**osd scrub load threshold** 参数不会影响此设置。

类型

浮点值

默认

每周一次。**60*60*24*7**

osd_deep_scrub_stride

描述

在进行深度清理时读取大小。

类型

32 位整数

默认

512 KB.**524288**

mon_warn_not_deep_scrubbed

描述

osd_deep_scrub_interval 后的秒数，以警告任何未清理的 PG。

类型

整数

默认

0 (无警告) 。

osd_deep_scrub_randomize_ratio

描述

清理的速率会随机变得深度清理（甚至在 **osd_deep_scrub_interval** 之前）。

类型

浮点值

默认

0.15 或 15%。

osd_deep_scrub_update_digest_min_age

描述

在清理更新整个对象摘要前，旧对象需要有多少秒。

类型

整数

默认

120（2 小时）。

6.4. 操作

操作设置允许您为服务请求配置线程数量。

默认情况下，Ceph 使用两个线程，超时时间为 30 秒，如果某个操作没有在这些时间参数指定的时间内完成，有 30 秒的 complaint 时间。在客户端操作和恢复操作之间设置操作优先级权重，以确保恢复过程中获得最佳性能。

osd_op_num_shards

描述

客户端操作的分片数量。

类型

32 位整数

默认

0

osd_op_num_threads_per_shard

描述

客户端操作的每个分片的线程数量。

类型

32 位整数

默认

0

osd_op_num_shards_hdd

描述

HDD 操作的分片数量。

类型

32 位整数

默认

5

osd_op_num_threads_per_shard_hdd

描述

每个分片用于 HDD 操作的线程数量。

类型

32 位整数

默认

1

osd_op_num_shards_ssd

描述

SSD 操作的分片数量。

类型

32 位整数

默认

8

osd_op_num_threads_per_shard_ssd

描述

用于 SSD 操作的每个分片的线程数量。

类型

32 位整数

默认

2

osd_client_op_priority

描述

为客户端操作设置的优先级。它相对于 **osd recovery op priority**。

类型

32 位整数

默认

63

有效范围

1-63

osd_recovery_op_priority

描述

恢复操作设置的优先级。它相对于 **osd client op priority**。

类型

32 位整数

默认

3

有效范围

1-63

osd_op_thread_timeout

描述

Ceph OSD 操作线程超时（以秒为单位）。

类型

32 位整数

默认

30

osd_op_complaint_time

描述

在经过指定秒数后，某个操作会变得令人满意。

类型

浮点值

默认

30

osd_disk_threads

描述

用于执行后台磁盘密集型 OSD 操作的磁盘线程数量，如清理和 snap 修剪。

类型

32 位整数

默认

1

osd_disk_thread_ioprio_class

描述

为磁盘线程设置 **ioprio_set (2)** I/O **调度**类。可接受值为：

- **idle**
- **be**
- **rt**

  idle 类表示磁盘线程的优先级比 OSD 中的其他线程的优先级更低。这对在处理客户端操作的 OSD 上减慢清理速度非常有用。

  be 类是默认值，与 OSD 中所有其他线程的优先级相同。

  rt 类表示磁盘线程的优先级高于 OSD 中所有其他线程。如果不再需要清理，并且必须以客户端操作为代价进行进度，这非常有用。

类型

字符串

默认

一个空字符串

osd_disk_thread_ioprio_priority

描述

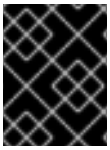
它设置磁盘线程的 **ioprio_set(2)** I/O 调度**优先级**，范围从 0（最高）到 7（最低）。如果给定主机上的所有 OSD 都**处于空闲状态**，且由于控制器拥塞导致 I/O 竞争，它可用于将一个 OSD 的磁盘线程优先级降低到 7，从而使另一个具有优先级 0 的 OSD 可能会更快地清理。

类型

0 到 7 范围内的整数，如果没有使用则为 -1。

默认

-1



重要

只有当两者都被设置为非默认值时，才会使用 **osd disk thread ioprio class** 和 **osd disk thread ioprio priority** 选项。另外，它只适用于 Linux 内核 CFQ 调度程序。

osd_op_history_size

描述

要跟踪的最大完成操作数。

类型

32-bit Unsigned 整数

默认

20

osd_op_history_duration

描述

要跟踪的最旧的已完成操作。

类型

32-bit Unsigned 整数

默认

600

osd_op_log_threshold

描述

一次显示多少个操作日志。

类型

32 位整数

默认

5

osd_op_timeout

描述

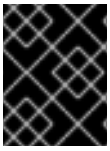
运行 OSD 操作超时的时间（以秒为单位）。

类型

整数

默认

0



重要

不要设置 **osd op timeout** 选项，除非您的客户端可以处理后果。例如，在虚拟机中运行的客户端设置此参数可能会导致数据崩溃，因为虚拟机将此超时解释为硬件故障。

6.5. 回填

将 Ceph OSD 添加到集群或从集群中删除时，CRUSH 算法会通过将放置组移到 Ceph OSD 或从中移出来重新平衡集群。迁移放置组和包含的对象可以大大降低集群操作性能。为保持操作性能，Ceph 会使用"回填"进程执行此迁移，这使 Ceph 将回填操作设置为比读取或写入数据的请求较低优先级。

osd_max_backfills

描述

允许从一个 OSD 或单个 OSD 允许的最大回填操作数。

类型

64-bit Unsigned 整数

默认

1

osd_backfill_scan_min

描述

每次回填扫描的最小对象数量。

类型

32 位整数

默认

64

osd_backfill_scan_max

描述

每次回填扫描的最大对象数量。

类型

32 位整数

默认

512

osd_backfillfull_ratio

描述

当 Ceph OSD 的全满比率超过这个值时，拒绝接受回填请求。

类型

浮点值

默认

0.85

osd_backfill_retry_interval

描述

在重试回填请求前等待的秒数。

类型

双

默认

10.0

6.6. OSD MAP

OSD 映射反映集群中运行的 OSD 守护进程。随着时间的推移，map epochs 的数量会增加。Ceph 提供以下设置，以确保 Ceph 性能和 OSD map 的增长更大。

osd_map_dedup

描述

启用删除 OSD map 中的重复项。

类型

布尔值

默认

true

osd_map_cache_size

描述

以 MB 为单位的 OSD map 缓存的大小。

类型

32 位整数

默认

50

osd_map_cache_bl_size

描述

OSD 守护进程中的内存中 OSD map 缓存的大小。

类型

32 位整数

默认

50

osd_map_cache_bl_inc_size

描述

内存中 OSD 映射缓存在 OSD 守护进程中递增的大小。

类型

32 位整数

默认

100

osd_map_message_max

描述

每个 MOSDMap 消息允许的最大映射条目。

类型

32 位整数

默认

40

6.7. 恢复

当集群启动或 Ceph OSD 意外终止并重启时，OSD 开始与其他 Ceph OSD 的对等点，然后才能执行写入操作。

如果 Ceph OSD 崩溃然后又恢复在线，通常它将与其他 Ceph OSD 同步，包含 PG 中最新版本的对象。发生这种情况时，Ceph OSD 进入恢复模式并寻求数据的最新副本，并使其映射重新变为最新。根据 Ceph OSD 停机的时长，OSD 对象和放置组可能会显著不同步。另外，如果故障域停止（例如，一个机架出现问题），则在恢复过程中可能出现多个 Ceph OSD 同时上线的问题。这样可使恢复过程消耗和大量资源。

为保持可操作的性能，Ceph 对允许 Ceph 在降级状态进行大规模恢复、线程和对象块大小的限制。

osd_recovery_delay_start

描述

在对等点完成后，Ceph 会在开始恢复对象前延迟指定的秒数。

类型

浮点值

默认

0

osd_recovery_max_active

描述

一次每个 OSD 活跃的恢复请求数。更多请求将加快恢复速度，但请求会给集群带来更大的负载。

类型

32 位整数

默认

3

osd_recovery_max_chunk

描述

要推送的数据恢复块的最大大小。

类型

64 位 Unsigned 整数

默认

8 << 20

osd_recovery_threads

描述

恢复数据的线程数量。

类型

32 位整数

默认

1

osd_recovery_thread_timeout

描述

超时恢复线程前的最大时间（以秒为单位）。

类型

32 位整数

默认

30

osd_recover_clone_overlap

描述

在恢复期间保留克隆重叠。应始终设为 **true**。

类型

布尔值

默认

true

6.8. 其它

osd_snap_trim_thread_timeout

描述

在超时 snap trim 线程前的最大时间（以秒为单位）。

类型

32 位整数

默认

60*60*1

osd_pg_max_concurrent_snap_trims

描述

并行 snap 修剪/PG 的最大数量。这将控制每个 PG 要一次修剪的对象数量。

类型

32 位整数

默认

2

osd_snap_trim_sleep

描述

在 PG 发布的每个修剪操作之间插入一个 sleep。

类型

32 位整数

默认

0

osd_max_trimming_pgs

描述

修剪 PG 的最大数量

类型

32 位整数

默认

2

osd_backlog_thread_timeout

描述

backlog 线程超时前需要经过的最大时间（以秒为单位）。

类型

32 位整数

默认

60*60*1

osd_default_notify_timeout

描述

OSD 默认通知超时（以秒为单位）。

类型

32 位整数（Unsigned）

默认

30

osd_check_for_log_corruption

描述

检查日志文件是否存在损坏。计算的代价可能会比较高。

类型

布尔值

默认

false

osd_remove_thread_timeout

描述

在超时删除 OSD 线程前的最大时间（以秒为单位）。

类型

32 位整数

默认

60*60

osd_command_thread_timeout

描述

命令线程超时前需要经过的最大时间（以秒为单位）。

类型

32 位整数

默认

10*60

osd_command_max_records

描述

限制丢失对象的数量。

类型

32 位整数

默认

256

osd_auto_upgrade_tmap

描述

在旧对象为 **omap** 使用 **tmap**。

类型

布尔值

默认

true

osd_tmapput_sets_users_tmap

描述

仅使用 **tmap** 进行调试。

类型

布尔值

默认

false

osd_preserve_trimmed_log

描述

保留会修剪的日志文件，但会占用更多磁盘空间。

类型

布尔值

默认

false

rados_osd_op_timeout

描述

RADOS 在从 RADOS 操作返回错误之前等待来自 OSD 的响应的秒数。值为 0 表示没有限制。

类型

双

默认

0

第 7 章 配置 MONITOR 和 OSD 互动

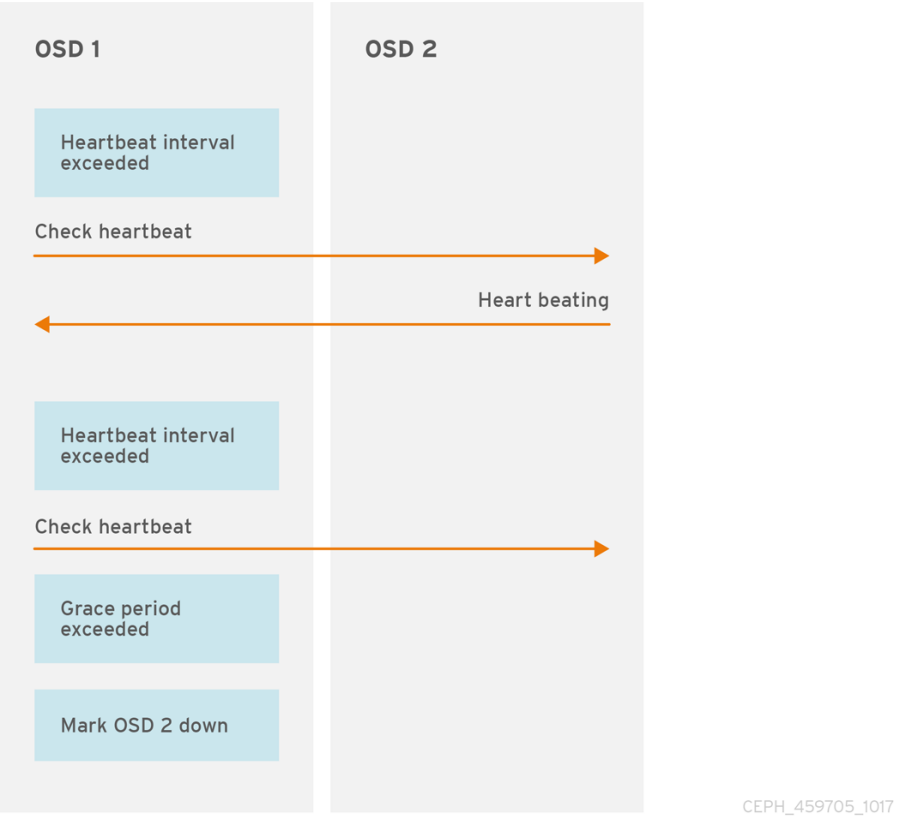
完成初始 Ceph 配置后，您可以部署并运行 Ceph。当您执行 **ceph health** 或 **ceph -s** 等命令时，Ceph Monitor 会报告 Ceph Storage 集群的当前状态。Ceph Monitor 通过需要来自每个 Ceph OSD 守护进程的报告，以及从 Ceph OSD 守护进程接收报告来了解 Ceph Storage 集群。如果 Ceph Monitor 没有接收报告，或者它收到 Ceph Storage 集群中的更改报告，Ceph 监控会更新 Ceph Cluster Map 的状态。

Ceph 为 Ceph Monitor 和 Ceph OSD 守护进程交互提供合理的默认设置。但是，您可以覆盖默认值。以下小节论述了 Ceph 监控器和 Ceph OSD 守护进程如何进行交互，以满足监控 Ceph Storage 集群的目的。

7.1. OSD 检查 HEARTBEATS

每个 Ceph OSD 守护进程会每 6 秒检查其他 Ceph OSD 守护进程的心跳。要更改心跳间隔，请在 Ceph 配置文件的 **[osd]** 部分下添加 osd **heartbeat 间隔** 设置，或者在运行时更改其值。

如果邻居 Ceph OSD 守护进程没有在 20 秒宽限期中发送 heartbeat 数据包，Ceph OSD 守护进程可能会认为邻居 Ceph OSD 守护进程 **停机**，并将它报告回 Ceph monitor，这将更新 Ceph 集群映射。若要更改此宽限期，请在 Ceph 配置文件的 **[osd]** 部分下添加 **osd heartbeat grace** 设置，或者在运行时设置它的值。



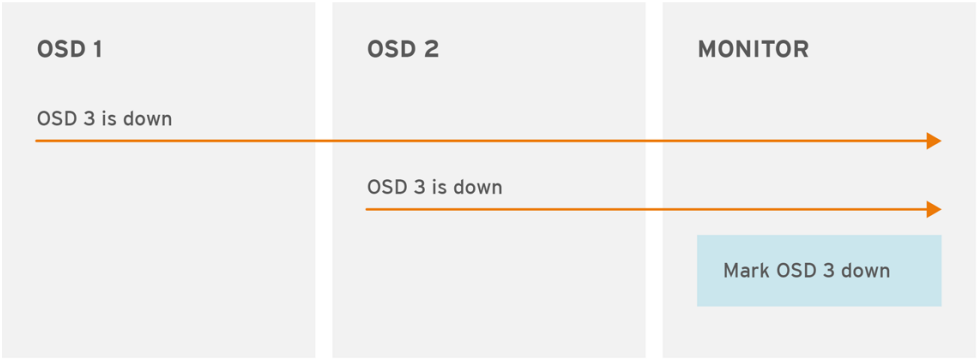
7.2. OSD 报告故障 OSD

默认情况下，**来自不同主机**的两个 Ceph OSD 守护进程必须报告给另一个 Ceph OSD 守护进程处于 **down** 状态的 Ceph 监控器，然后确认报告的 Ceph OSD 守护进程为 **down**。

但是，所有 OSD 报告失败的可能性都位于具有错误交换机的机架中，导致 OSD 之间的连接问题。

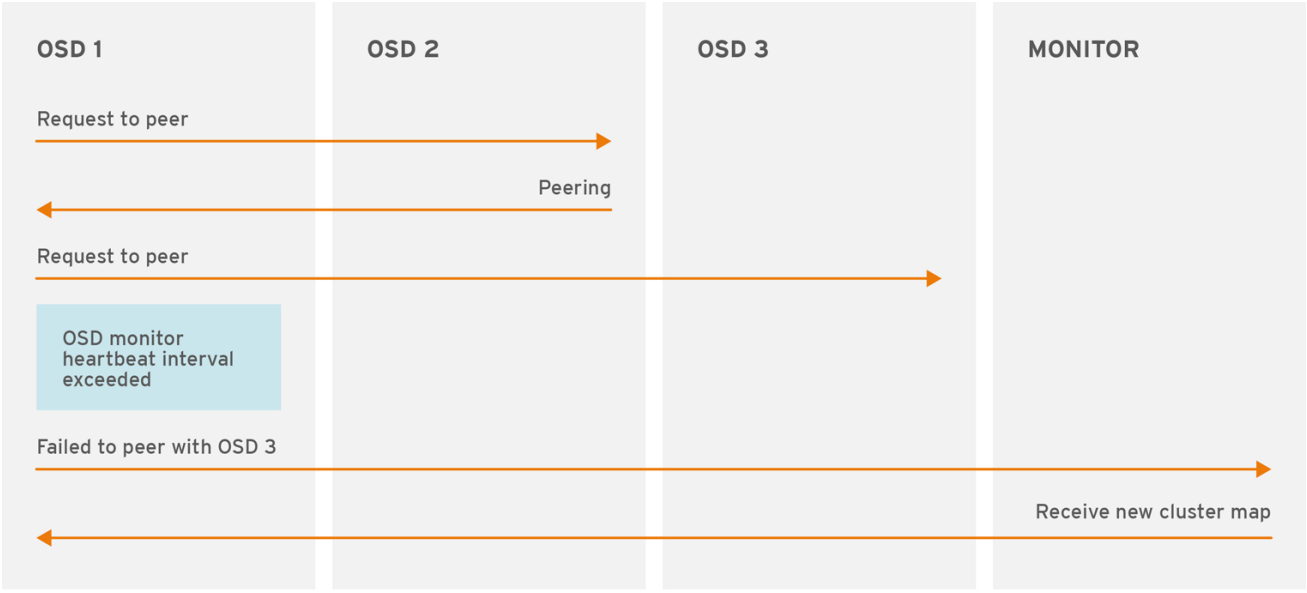
为避免"错误警报"，Ceph 会将故障报告为类似 lagy 的"subcluster"的代理。虽然情况并非总是如此，但可能帮助管理员对性能不良的系统子集进行本地化处理。

Ceph 使用 **mon_osd_reporter_subtree_level** 设置，将 peer 分到"cluster"的常用级别类型。默认情况下，仅需要两个**来自不同子树**的报告，才能报告另一个 Ceph OSD 守护进程为 **down**。管理员可以通过在 Ceph 的配置文件的 **[mon]** 部分下添加 **mon_osd_min_down_reporters** 和 **mon_osd_reporter_subtree_level** 设置，或在运行时指定值，修改报告一个 Ceph OSD Daemon 为 **down** 所需的来自唯一子树的报告者的数量以及通用祖先类型。



7.3. OSD 报告同线故障

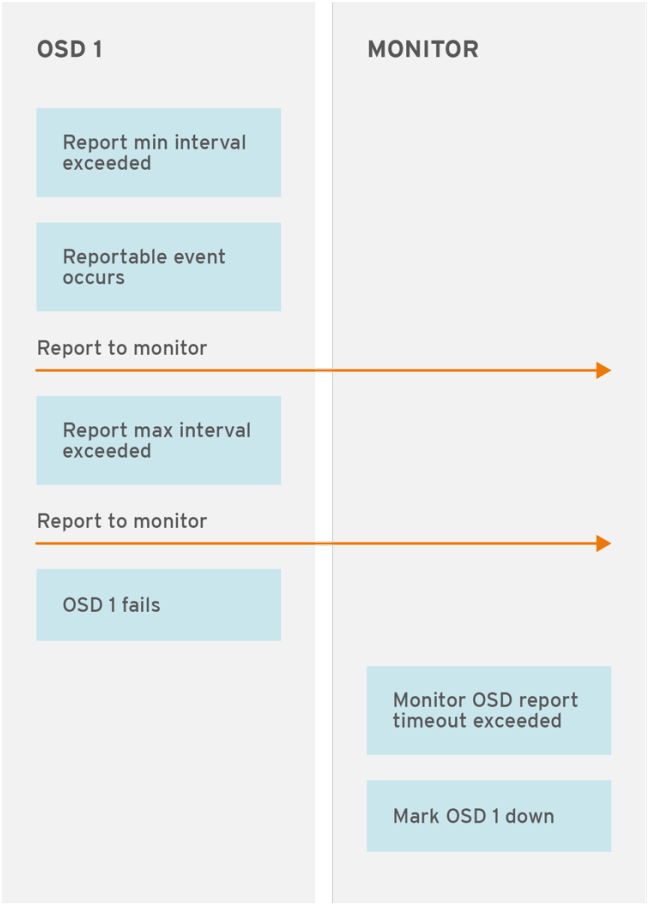
如果 Ceph OSD 守护进程无法与其 Ceph 配置文件或 cluster map 中定义的任何 Ceph OSD 守护进程的对等点，它会每 30 秒对集群 map 的最新副本发出 Ceph Monitor 命令。您可以通过在 Ceph 配置文件的 **[osd] 部分下添加 osd mon heartbeat 间隔**，或者在运行时设置值来更改 Ceph 监控心跳间隔。



7.4. OSD 报告状态

如果一个 Ceph OSD 守护进程没有报告到 Ceph 监控器，Ceph Monitor 会在 **mon osd report timeout** 指定的时间超过是认为 Ceph OSD 守护进程处于 **down** 状态。当可报告事件（如故障）时，Ceph OSD 守护进程会向 Ceph 监控器发送报告，这是放置组统计的变化、**up_thru** 或在 5 秒内引导时发生的变化。您可以通过在 Ceph 配置文件的 **[osd]** 部分下添加 **osd mon report interval min** 设置，或者通过设置运行时设置值来更改 Ceph OSD 守护进程最小报告间隔。

Ceph OSD 守护进程每 120 秒向 Ceph 监控器发送报告，无论发生任何显著变化。您可以通过在 Ceph 配置文件的 **[osd]** 部分下添加 **osd mon report interval max** 设置，或者在运行时设置值，以更改 Ceph Monitor 报告间隔。



7.5. 配置设置

修改 heartbeat 设置时，请将它们包含在 Ceph 配置文件的 **[global]** 部分中。

7.5.1. 监控设置

mon_osd_min_up_ratio

描述

Ceph 在标记 Ceph OSD 守护进程为 **down** 前的最小 **up** Ceph OSD 守护进程比率。

类型

双

默认

.3

mon_osd_min_in_ratio

描述

在 Ceph 将 Ceph OSD 守护进程标记为 **out** 之前， **in** Ceph OSD 守护进程的最小比率。

类型

双

默认

.3

mon_osd_laggy_halflife

描述

laggy 估算将会衰变的秒数。

类型

整数

默认

60*60

mon_osd_laggy_weight

描述

laggy 估算衰变中新样本的权重。

类型

双

默认

0.3

mon_osd_laggy_max_interval

描述

设置 **lagggy_interval** 的值（以秒为单位）。monitor 使用调整性方法来评估特定 OSD 的 **laggy_interval**。这个值将用于计算该 OSD 的宽限期。

类型

整数

默认

300

mon_osd_adjust_heartbeat_grace

描述

如果设置为 **true**，Ceph 将根据 **laggy** 估算进行扩展。

类型

布尔值

默认

true

mon_osd_adjust_down_out_interval

描述

如果设置为 **true**，Ceph 将基于 **laggy** 估算进行扩展。

类型

布尔值

默认

true

mon_osd_auto_mark_in

描述

Ceph 会将任何启动 Ceph OSD 守护进程标记为 **in** Ceph Storage 集群。

类型

布尔值

默认

false

mon_osd_auto_mark_auto_out_in

描述

Ceph 将自动标记为 **out** Ceph Storage Cluster 的引导 Ceph OSD 守护进程标记为 **in** 集群。

类型

布尔值

默认

true

mon_osd_auto_mark_new_in

描述

Ceph 会将引导新的 Ceph OSD 守护进程标记为 **in** Ceph Storage 集群。

类型

布尔值

默认

true

mon_osd_down_out_interval

描述

当一个 Ceph OSD 守护进程没有响应时，Ceph 在将其标记为 **down** 和 **out** 前等待的时间。

类型

32 位整数

默认

600

mon_osd_downout_subtree_limit

描述

Ceph 将自动标记为 **out** 的最大 CRUSH 单元类型。

类型

字符串

默认

rack

mon_osd_reporter_subtree_level

描述

此设置为报告 OSD 定义父 CRUSH 单元类型。如果 OSD 找到不响应的对等点，OSD 会向监控器发送失败报告。monitor 标记报告的 OSD **down**，并在宽限期后设置为 **out**。

类型

字符串

默认

主机

mon_osd_report_timeout

描述

在将没有响应的 Ceph OSD Daemons 声明为 **down** 前经过的宽限期（以秒为单位）。

类型

32 位整数

默认

900

mon_osd_min_down_reporters

描述

报告 **down** Ceph OSD 守护进程需要的最小 Ceph OSD 守护进程的数量。

类型

32 位整数

默认

2

7.5.2. OSD 设置

osd_heartbeat_address

描述

用于心跳的 Ceph OSD 守护进程的网络地址。

类型

地址

默认

主机地址。

osd_heartbeat_interval

描述

Ceph OSD 守护进程如何 ping 对等点（以秒为单位）。

类型

32 位整数

默认

6

osd_heartbeat_grace

描述

当 Ceph OSD 守护进程未显示 Ceph Storage 集群考虑它时的心跳时间。

类型

32 位整数

默认

20

osd_mon_heartbeat_interval

描述

如果没有 Ceph OSD 守护进程同级服务器，Ceph OSD 守护进程会如何 ping Ceph 监控器。

类型

32 位整数

默认

30

osd_mon_report_interval_max

描述

Ceph OSD 守护进程在报告到 Ceph 监控器之前可以等待的时间（以秒为单位）。

类型

32 位整数

默认

120

osd_mon_report_interval_min

描述

在向 Ceph monitor 报告前，Ceph OSD 守护进程可以从启动或其他可报告事件等待的最少秒数。

类型

32 位整数

默认

5

有效范围

应小于 **osd mon report interval max**

osd_mon_ack_timeout

描述

等待 Ceph Monitor 的秒数，以确认对统计数据的请求。

类型

32 位整数

默认

30

第 8 章 文件存储配置参考

8.1. 扩展属性

扩展属性 (XATTR) 是 CephFS 配置中的一个重要方面。有些文件系统对存储在扩展属性中的字节数有限制。另外，在某些情况下，该文件系统可能无法作为存储扩展属性的替代方法。以下设置通过使用一种方法存储底层文件系统中的扩展属性来提高 CephFS 性能。

Ceph 扩展属性作为 **inline xattr** 存储，使用底层文件系统提供的扩展属性（若不强制实施大小限制）。如果大小限制（例如，ext4 上总共 4KB），则当 **文件存储最大内联 xattrs** 阈值时，一些 Ceph 扩展属性将存储在名为 **omap** 的 key-value 数据库中。

filestore_xattr_use_omap

描述

对 XATTRS 使用对象映射。对于 ext4 文件系统，设置为 **true**。

类型

布尔值

必需

否

默认

false

filestore_omap_header_cache_size

描述

决定用于缓存对象 **omap** 标头的 LRU 的大小。较大的值会使用更多内存，但可以减少 **omap** 中的查找。（仅限专家）。

类型

整数

默认

1024

filestore_omap_backend

描述

用于确定哪个后端用于 **omap**。可以设置为 **leveldb** 或 **rocksdb**。（仅限专家。**rocksdb** 是实验性的。）

类型

字符串

默认

leveldb

filestore_debug_omap_check

描述

对同步进行调试检查。昂贵。仅用于调试。

类型

布尔值

必需

否

默认

0

filestore_max_inline_xattr_size

描述

每个对象存储在文件系统中的最大扩展属性大小（即 XFS、btrfs、ext4 等等）。不应大于文件系统可以处理的量。

类型

Unsigned 32 位整数

必需

否

默认

512

filestore_max_inline_xattrs

描述

每个对象存储在文件系统中的最大扩展属性数量。

类型

32 位整数

必填

否

默认

2

filestore_max_inline_xattr_size_xfs

描述

每个对象的 XFS 文件系统中存储的扩展属性的最大大小。不应大于文件系统可以处理的量。

类型

Unsigned 32 位整数

默认

65536

filestore_max_inline_xattr_size_btrfs

描述

每个对象存储在文件系统中的扩展属性的最大大小。不应大于文件系统可以处理的量。

类型

Unsigned 32 位整数

默认

2048

filestore_max_inline_xattr_size_other

描述

存储在文件系统中为 btrfs 或 XFS 以外的文件系统的最大扩展属性大小。不应大于文件系统可以处理的量。

类型

Unsigned 32 位整数

默认

512

filestore_max_inline_xattrs

描述

每个对象存储在文件系统中的最大扩展属性数量。覆盖细粒度设置。

类型

Unsigned 32 位整数

默认

0

filestore_max_inline_xattrs_xfs

描述

每个对象的 XFS 文件系统中存储的最大扩展属性数。

类型

Unsigned 32 位整数

默认

10

filestore_max_inline_xattrs_btrfs

描述

每个对象存储在 btrfs 文件系统中的扩展属性的最大数量。

类型

Unsigned 32 位整数

默认

10

filestore_max_inline_xattrs_other

描述

每个对象存储在 btrfs 或 XFS 的文件系统中的扩展属性的最大数量。

类型

Unsigned 32 位整数

默认

2

8.2. 同步间隔

文件存储需要定期静止写操作并同步文件系统，这将创建一致的提交点。然后，它可以释放与提交点的日

志条目。同步会更频繁地减少执行同步所需的时间，并减少日志中需要保留的数据量。不太频繁的同步允许后备文件系统合并小的写入和元数据更新，从而获得更优化的编写和元数据更新，从而带来更高效的同步。

filestore_max_sync_interval

描述

同步文件存储的最大间隔时间（以秒为单位）。

类型

双

必填

否

默认

5

filestore_min_sync_interval

描述

同步文件存储的最小间隔（以秒为单位）。

类型

双

必填

否

默认

.01

8.3. FLUSHER

文件存储冲刷器会强制使用 **sync file range** 选项写入大型写操作中的数据，以便降低事件同步的成本。在实践中，禁用文件存储冲刷器似乎提高了性能。

filestore_flusher

描述

启用文件存储冲刷器。

类型

布尔值

必需

否

默认

false

filestore_flusher_max_fds

描述

设置 flusher 的最大文件描述符数。

类型

整数

必需

否

默认

512

filestore_sync_flush

描述

启用同步清除器。

类型

布尔值

必需

否

默认

false

filestore_fsync_flushes_journal_data

描述

在文件系统同步期间清除日志数据。

类型

布尔值

必需

否

默认

false

8.4. 队列

以下设置对文件存储队列的大小提供了限制。

filestore_queue_max_ops

描述

定义文件存储在排队新操作前接受的最大操作数量。

类型

整数

必需

不。对性能的影响最小。

默认

500

filestore_queue_max_bytes

描述

操作的最大字节数。

类型

整数

必需

否

默认

100 << 20

filestore_queue_committing_max_ops

描述

文件存储可以提交的最多操作数。

类型

整数

必需

否

默认

500

filestore_queue_committing_max_bytes

描述

文件存储可以提交的最多字节数。

类型

整数

必需

否

默认

100 << 20

8.5. WRITEBACK THROTTLE

Ceph 在内核中复制一些回写行为，因为页面缓存往往保持脏数据回路太长。

filestore_wbthrottle_enable

描述

启用文件存储 write-back throttle。文件存储节流被用来防止每个文件存储同步前进行大量未提交的数据。（仅限专家）。

类型

布尔值

默认

true

filestore_wbthrottle_btrfs_bytes_start_flusher

描述

Ceph 开始刷新 btrfs 文件系统的脏字节数。

类型

64-bit Unsigned 整数

默认

41943040

filestore_wbthrottle_btrfs_bytes_hard_limit

描述

Ceph 开始节流 I/O 的脏字节数阈值，直到 btrfs 的 flusher 捕获为止。

类型

64-bit Unsigned 整数

默认

419430400

filestore_wbthrottle_btrfs_ios_start_flusher

描述

Ceph 开始刷新 btrfs 的脏 I/Os 阈值。

类型

64-bit Unsigned 整数

默认

500

filestore_wbthrottle_btrfs_ios_hard_limit

描述

Ceph 开始节流 IO 的脏 I/Os 阈值，直到 btrfs 达到清理者为止。

类型

64-bit Unsigned 整数

默认

5000

filestore_wbthrottle_btrfs_inodes_start_flusher

描述

Ceph 开始刷新 btrfs 的脏内节点阈值。

类型

64-bit Unsigned 整数

默认

500

filestore_wbthrottle_btrfs_inodes_hard_limit

描述

Ceph 开始节流 IO 的脏内节点阈值，直到 btrfs 的 flusher 捕获为止。必须小于 **fd** 限值。

类型

64-bit Unsigned 整数

默认

5000

filestore_wbthrottle_xfs_bytes_start_flusher

描述

Ceph 开始刷新 XFS 文件系统的脏字节数阈值。

类型

64-bit Unsigned 整数

默认

41943040

filestore_wbthrottle_xfs_bytes_hard_limit

描述

Ceph 开始节流 IO 的脏字节数阈值直至 XFS 的清理器捕获为止。

类型

64-bit Unsigned 整数

默认

419430400

filestore_wbthrottle_xfs_ios_start_flusher

描述

Ceph 开始刷新 XFS 的脏 I/Os 阈值。

类型

64-bit Unsigned 整数

默认

500

filestore_wbthrottle_xfs_ios_hard_limit

描述

Ceph 开始节流 IO 的脏 I/Os 阈值，直到 XFS 的清理器捕获为止。

类型

64-bit Unsigned 整数

默认

5000

filestore_wbthrottle_xfs_inodes_start_flusher

描述

Ceph 开始对 XFS 后台刷新的脏内节点阈值。

类型

64-bit Unsigned 整数

默认

500

filestore_wbthrottle_xfs_inodes_hard_limit

描述

Ceph 开始节流 IO 的脏内节点阈值，直到 XFS 的清理器捕获为止。必须小于 **fd** 限值。

类型

64-bit Unsigned 整数

默认

5000

8.6. 超时

filestore_op_threads

描述

并行执行的文件系统操作线程的数量。

类型

整数

必需

否

默认

2

filestore_op_thread_timeout

描述

文件系统操作线程的超时（以秒为单位）。

类型

整数

必需

否

默认

60

filestore_op_thread_suicide_timeout

描述

提交操作的超时时间，然后取消提交（以秒为单位）。

类型

整数

必需

否

默认

180

8.7. B-TREE 文件系统

filestore_btrfs_snap

描述

启用 btrfs 文件存储的快照。

类型

布尔值

必需

不。仅用于 btrfs。

默认

true

filestore_btrfs_clone_range

描述

为 btrfs 文件存储启用克隆范围。

类型

布尔值

必需

不。仅用于 btrfs。

默认

true

8.8. JOURNAL

filestore_journal_parallel

描述

为 btrfs 启用并行日志。

类型

布尔值

必需

否

默认

false

filestore_journal_writeahead

描述

启用 write-ahead 日志，默认为 XFS。

类型

布尔值

必需

否

默认

false

filestore_journal_trailing

描述

弃用，不要使用。

类型

布尔值

必需

否

默认

false

8.9. 其它

filestore_merge_threshold

描述

在合并到父进程前，子目录中的最小文件数量：负数值意味着禁用子目录合并。

类型

整数

必需

否

默认

10

filestore_split_multiple

描述

filestore_split_multiple * abs (filestore_merge_threshold) * 16 是子目录中的最大文件数量，然后再分割为子目录。

类型

整数

必需

否

默认

2

filestore_update_to

描述

限制文件存储自动升级到指定版本。

类型

整数

必需

否

默认

1000

filestore_blackhole

描述

丢弃所有新事务。

类型

布尔值

必需

否

默认

false

filestore_dump_file

描述

存储事务转储的文件。

类型

布尔值

必需

否

默认

false

filestore_kill_at

描述

在第 n 机会中注入失败。

类型

字符串

必需

否

默认

false

filestore_fail_eio

描述

EIO 上意外失败或终止。

类型

布尔值

必需

否

默认

true

第 9 章 日志配置参考

Ceph OSD 出于以下原因使用日志：

速度

日志可让 Ceph OSD 守护进程快速提交小的写操作。Ceph 会按顺序将小的随机 I/O 写入日志，这可以通过使后备文件系统有更多时间来合并写入操作，从而加快激增的工作负载。但是，Ceph OSD 守护进程的日志可能会导致性能激增，缩短高速写入的短暂增加，并在文件系统捕获到日志时没有写入进度。

一致性

Ceph OSD 守护进程需要一个文件系统接口来保证原子复合操作。Ceph OSD 守护进程向日志写入操作描述，并将操作应用到文件系统。这允许对对象进行原子更新（如放置组元数据）。每隔几秒 - 在 **filestore max sync interval** 和 **filestore min sync interval** 设置之间，Ceph OSD 会停止写操作并与文件系统同步日志，从而使 Ceph OSD 能够修剪日志中的操作并重复利用空间。在失败时，Ceph OSD 在最后一次同步操作后重新显示日志。

9.1. 设置

Ceph OSD 守护进程支持以下日志设置：

journal_dio

描述

启用直接 I/O 到日志。要求 **journal 块对齐** 选项设为 **true**。

类型

布尔值

必需

使用 A **io 时是的**。

默认

true

journal_aio

描述

启用 **libaio** 对日志进行异步写入。要求 **journal dio** 选项设为 **true**。

类型

布尔值

必需

No.

默认

True.

journal_block_align

描述

块对齐写入操作。**dio** 和 **aio** 是必需的。

类型

布尔值

必需

使用 **dio** 和 **aio** 时是的。

默认

true

journal_max_write_bytes

描述

日志一次写入的最大字节数。

类型

整数

必需

否

默认

10 << 20

journal_max_write_entries

描述

日志每次将写入的最大条目数。

类型

整数

必需

否

默认

100

journal_queue_max_ops

描述

队列中允许的最大操作数量。

类型

整数

必需

否

默认

500

journal_queue_max_bytes

描述

队列中允许的最大字节数。

类型

整数

必需

否

默认

10 << 20

journal_align_min_size

描述

使数据有效负载对齐大于指定最小值。

类型

整数

必需

否

默认

64 << 10

journal_zero_on_create

描述

导致文件存储在" **mkfs 期间使用 0"** 覆盖整个日志。

类型

布尔值

必需

否

默认

false

第 10 章 日志配置参考

Ceph 配置文件中不需要记录和调试设置，但您可以根据需要覆盖默认设置。

选项采用单个项目，无论频道如何，假定为所有守护进程的默认值。例如，指定 "info" 被解释为 "default=info"。但是，选项也可以使用键/值对。例如： "default=daemon audit=local0" 解释为 "default all to 'daemon'，使用 'local0' 覆盖 'audit'。

Ceph 支持以下设置：

log_file

描述

集群日志记录文件的位置。

类型

字符串

必需

否

默认

/var/log/ceph/$cluster-$name.log

mon_cluster_log_file

描述

监控集群日志文件的位置。

类型

字符串

必需

否

默认

/var/log/ceph/$cluster.log

log_max_new

描述

新日志文件的最大数量。

类型

整数

必需

否

默认

1000

log_max_recent

描述

日志文件中包括的最新事件的最大数量。

类型

整数

必需

否

默认

1000000

log_flush_on_exit

描述

决定 Ceph 在退出后是否清除日志文件。

类型

布尔值

必需

否

默认

true

mon_cluster_log_file_level

描述

监控集群的文件日志级别。有效设置包括 "debug"、"info"、"sec"、"warn"和"error"。

类型

字符串

默认

"info"

log_to_stderr

描述

确定日志记录消息是否出现在 **stderr** 中。

类型

布尔值

必需

否

默认

true

err_to_stderr

描述

确定 **stderr** 中显示的错误消息。

类型

布尔值

必需

否

默认

true

log_to_syslog

描述

确定日志记录信息是否出现在 **syslog** 中。

类型

布尔值

必需

否

默认

false

err_to_syslog

描述

确定 **syslog** 中显示的错误消息。

类型

布尔值

必需

否

默认

false

clog_to_syslog

描述

确定 **clog** 消息是否发送到 **syslog**。

类型

布尔值

必需

否

默认

false

mon_cluster_log_to_syslog

描述

确定集群日志是否输出到 **syslog**。

类型

布尔值

必需

否

默认

false

mon_cluster_log_to_syslog_level

描述

监控集群的 syslog 日志记录级别。有效设置包括 "debug"、"info"、"sec"、"warn"和"error"。

类型

字符串

默认

"info"

mon_cluster_log_to_syslog_facility

描述

生成 syslog 输出的工具。这通常设置为 Ceph 守护进程的"后台程序"。

类型

字符串

默认

"daemon"

clog_to_monitors

描述

决定是否将 **clog** 消息发送到 monitor。

类型

布尔值

必需

否

默认

true

mon_cluster_log_to_graylog

描述

确定集群是否输出日志消息到 graylog。

类型

字符串

默认

"false"

mon_cluster_log_to_graylog_host

描述

graylog 主机的 IP 地址。如果 graylog 主机与监控器主机不同，请使用适当的 IP 地址覆盖此设置。

类型

字符串

默认

"127.0.0.1"

mon_cluster_log_to_graylog_port

描述

Graylog 日志将发送到此端口。确保端口已打开以接收数据。

类型

字符串

默认

"12201"

10.1. OSD

osd_preserve_trimmed_log

描述

在修剪后保留修剪日志。

类型

布尔值

必需

否

默认

false

osd_tmapput_sets_uses_tmap

描述

使用 **tmap**。仅限 debug。

类型

布尔值

必需

否

默认

false

osd_min_pg_log_entries

描述

放置组的最小日志条目数量。

类型

32-bit Unsigned 整数

必需

否

默认

1000

osd_op_log_threshold

描述

在一次传递中显示多少个 op 日志消息。

类型

整数

必需

否

默认

5

10.2. 文件存储

filestore_debug_omap_check

描述

对同步进行调试检查。这是代价昂贵的操作。

类型

布尔值

必需

否

默认

0

10.3. CEPH 对象网关

rgw_log_nonexistent_bucket

描述

记录不存在的存储桶。

类型

布尔值

必需

否

默认

false

rgw_log_object_name

描述

记录对象的名称。

类型

字符串

必需

否

默认

%Y-%m-%d-%H-%i-%n

rgw_log_object_name_utc

描述

对象日志名称包含 UTC。

类型

布尔值

必需

否

默认

false

rgw_enable_ops_log

描述

启用每个 RGW 操作的日志记录。

类型

布尔值

必需

否

默认

true

rgw_enable_usage_log

描述

启用 RGW 的带宽使用量的日志记录。

类型

布尔值

必需

否

默认

true

rgw_usage_log_flush_threshold

描述

清除待处理的日志数据的阈值。

类型

整数

必需

否

默认

1024

rgw_usage_log_tick_interval

描述

每隔 **s** 秒刷新待处理的日志数据。

类型

整数

必需

否

默认

30

rgw_intent_log_object_name

描述, 类型

字符串

必需

否

默认

%Y-%m-%d-%i-%n

rgw_intent_log_object_name utc

描述

在意图日志对象名称中包含 UTC 时间戳。

类型

布尔值

必需

否

默认

false