Red Hat Ceph Storage 3

Red Hat Enterprise Linux 安装指南

在 Red Hat Enterprise Linux 上安装 Red Hat Ceph Storage

Last Updated: 2024-01-21

Red Hat Ceph Storage 3 Red Hat Enterprise Linux 安装指南

在 Red Hat Enterprise Linux 上安装 Red Hat Ceph Storage

法律通告

摘要

本文档提供有关在 AMD64 和 Intel 64 架构上运行的 Red Hat Enterprise Linux 7 上安装 Red Hat Ceph Storage 的说明。

目录

第 1 章 什么是 RED HAT CEPH STORAGE？

Red Hat Ceph Storage 是一个可扩展、开放、软件定义的存储平台，它将最稳定版本的 Ceph 存储系统与 Ceph 管理平台、部署实用程序和支持服务相结合。

Red Hat Ceph Storage 专为云基础架构和 Web 规模对象存储而设计。Red Hat Ceph Storage 集群由以下类型的节点组成：

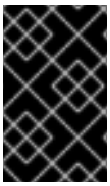
Red Hat Ceph Storage Ansible 管理节点

此类节点充当之前版本的 Red Hat Ceph Storage 的传统 Ceph 管理节点。这种类型的节点提供以下功能：

- 集中存储集群管理
- Ceph 配置文件和密钥
- （可选）用于在因安全原因无法访问互联网的节点上安装 Ceph 的本地存储库。

监控节点

每个监控节点运行 monitor 守护进程(**ceph-mon**)，后者维护 cluster map 的主副本。集群映射包含集群拓扑。连接 Ceph 集群的客户端从 monitor 中检索 cluster map 的当前副本，使客户端能够从集群读取和写入数据。



重要

Ceph 可以使用一个监控器运行；但是，为了保证生产环境集群中的高可用性，红帽将仅支持具有至少三个 monitor 节点的部署。红帽建议为超过 750 OSD 的存储集群部署总计 5 个 Ceph Monitor。

OSD 节点

每个对象存储设备(OSD)节点运行 Ceph OSD 守护进程(**ceph-osd**)，它与附加到节点的逻辑卷交互。Ceph 在这些 OSD 节点上存储数据。
Ceph 可在只有很少 OSD 节点的环境中运行，默认为三个。但对于生产环境，自中等范围环境开始（例如，在一个存储集群中包括 50 个 OSD）才可能看到其在性能方面的优势。理想情况下，Ceph 集群具有多个 OSD 节点，通过创建 CRUSH map 来允许隔离的故障域。

MDS 节点

每个元数据服务器(MDS)节点运行 MDS 守护进程(**ceph-mds**)，后者管理与 Ceph 文件系统(CephFS)中存储的文件相关的元数据。MDS 守护进程也协调对共享集群的访问。

对象网关节点

Ceph 对象网关节点运行 Ceph RADOS 网关守护进程(**ceph-radosgw**)，它是构建于 **librados** 上的对象存储接口，为应用提供 Ceph 存储群集的 RESTful 网关。Ceph 对象网关支持两个接口：

S3

通过与 Amazon S3 RESTful API 的大子集兼容的接口提供对象存储功能。

Swift

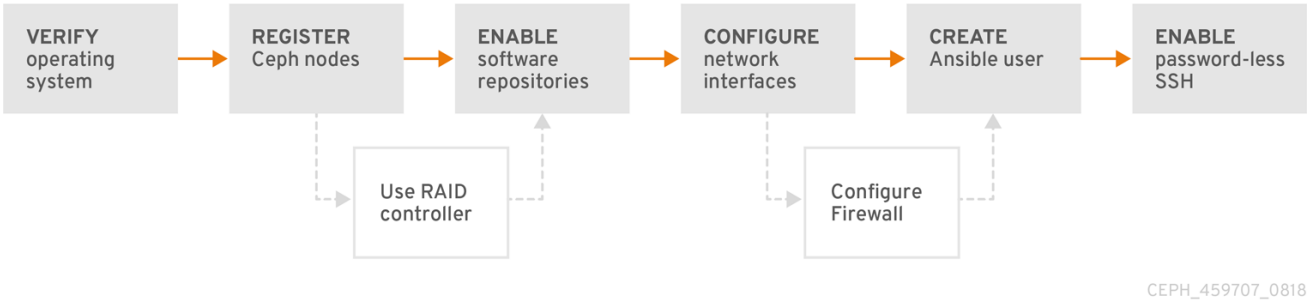
通过与 OpenStack Swift API 的大集兼容的接口提供对象存储功能。

有关 Ceph 架构的详情，请查看 Red Hat Ceph Storage 3 的 *架构指南*。

有关最低推荐硬件，请参阅 Red Hat Ceph Storage Hardware Selection Guide 3。

第 2 章 安装 RED HAT CEPH STORAGE 的要求

图 2.1. 先决条件工作流



在安装 Red Hat Ceph Storage 之前，请先查看以下要求，并相应地准备各个 monitor、OSD、元数据服务器和客户端节点。

2.1. 先决条件

- 验证硬件是否满足最低要求。详情请参阅 Red Hat Ceph Storage 3 的硬件指南。

2.2. 安装 RED HAT CEPH STORAGE 的要求检查列表

任务	必需	节	建议
验证操作系统版本	是	第 2.3 节 “Red Hat Ceph Storage 的操作系统要求”	
注册 Ceph 节点	是	第 2.4 节 “将 Red Hat Ceph Storage 节点注册到 CDN 和附加订阅”	
启用 Ceph 软件存储库	是	第 2.5 节 “启用 Red Hat Ceph Storage Repositories”	
使用带有 OSD 节点的 RAID 控制器	否	第 2.6 节 “将 RAID 控制器用于 OSD 节点的注意事项（可选）”	在 RAID 控制器中启用回写缓存可能会导致 OSD 节点增加较小的 I/O 写入吞吐量。
配置网络	是	第 2.8 节 “验证 Red Hat Ceph Storage 的网络配置”	至少需要一个公共网络。但是，推荐使用一个专用的网络以用于集群通信。
配置防火墙	否	第 2.9 节 “为 Red Hat Ceph Storage 配置防火墙”	防火墙可以提高网络的信任级别。

任务	必需	节	建议
创建 Ansible 用户	是	第 2.10 节 “创建具有 **sudo** 访问权限的 Ansible 用户”	所有 Ceph 节点上都需要创建 Ansible 用户。
启用无密码 SSH	是	第 2.11 节 “为 Ansible 启用无密码 SSH”	Ansible 需要。

注意

默认情况下，**ceph-ansible** 会根据需要安装 NTP。如果自定义 NTP，请参考手动安装 Red Hat Ceph Storage 中的*为 Red Hat Ceph Storage 配置网络时间协议*，了解如何配置 NTP 才能正常工作。

2.3. RED HAT CEPH STORAGE 的操作系统要求

Red Hat Ceph Storage 3 需要 Red Hat Enterprise Linux 7 更新 5 或更高版本。在集群中的所有节点上使用相同的版本和架构。

重要

Red Hat Enterprise Linux 8 不支持 Red Hat Ceph Storage 3。

重要

红帽不支持带有异构操作系统或版本的集群。

其它资源

- Red Hat Enterprise Linux 7 的 安装指南。
- Red Hat Enterprise Linux 7 系统管理员指南。

返回要求清单

2.4. 将 RED HAT CEPH STORAGE 节点注册到 CDN 和附加订阅

将每个 Red Hat Ceph Storage (RHCS) 节点注册到 Content Delivery Network (CDN)，再附加适当的订阅，以便节点可以访问软件存储库。每个 RHCS 节点都必须能够访问完整的 Red Hat Enterprise Linux 7 基本内容和额外 (extras) 存储库的内容。

注意

对于在安装过程中无法访问互联网的 RHCS 节点，请使用 Red Hat Satellite 服务器提供软件内容。或者，挂载本地 Red Hat Enterprise Linux 7 Server ISO 镜像，并将 RHCS 节点指向 ISO 镜像。如需更多详细信息，请联系红帽支持。

有关将 Ceph 节点注册到 Red Hat Satellite 服务器的更多信息，请参阅 如何将 Ceph 注册到 Satellite 6，以及如何在红帽客户门户上将 Ceph 注册到 Satellite 5 文章。

先决条件

- 有效的红帽订阅
- RHCS 节点必须能够连接到互联网。

流程

以 **root** 用户身份在存储集群中的所有节点上执行以下步骤。

1. 注册节点。提示时，请输入您的红帽客户门户网站凭证：

```
# subscription-manager register
```

2. 从 CDN 拉取最新的订阅数据：

```
# subscription-manager refresh
```

3. 列出 Red Hat Ceph Storage 的所有可用订阅：

```
# subscription-manager list --available --all --matches="*Ceph*"
```

确定适当的订阅并检索其池 ID。

4. 附加订阅：

```
# subscription-manager attach --pool=$POOL_ID
```

替换

- **$POOL_ID**，带有上一步中标识的池 ID。

5. 禁用默认软件存储库。然后，启用 Red Hat Enterprise Linux 7 Server、Red Hat Enterprise Linux 7 Server Extras 和 RHCS 软件仓库：

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-extras-rpms
# subscription-manager repos --enable=rhel-7-server-rhceph-3-mon-els-rpms
# subscription-manager repos --enable=rhel-7-server-rhceph-3-osd-els-rpms
# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

6. 更新系统以接收最新的软件包：

```
# yum update
```

其它资源

- 请参阅 Red Hat Enterprise Linux 7 系统管理员指南中的注册系统和管理订阅章节。
- 第 2.5 节 “启用 Red Hat Ceph Storage Repositories”

返回要求清单

2.5. 启用 RED HAT CEPH STORAGE REPOSITORIES

您必须先选择安装方法，然后才能安装 Red Hat Ceph Storage。Red Hat Ceph Storage 支持两种安装方法：

- 内容交付网络 (CDN)
  对于带有可以直接连接到互联网的 Ceph 节点的 Ceph 存储集群，请使用红帽订阅管理器来启用所需的 Ceph 存储库。
- 本地存储库
  对于安全措施使节点无法访问互联网的 Ceph 存储集群，请从作为 ISO 镜像提供的单个软件构建中安装 Red Hat Ceph Storage 3.3，这将允许您安装本地存储库。

先决条件

- 有效的客户订阅。
- 对于 CDN 安装，RHCS 节点必须能够连接到互联网。
- 对于 CDN 安装，使用 CDN 注册集群节点。
- 禁用 EPEL 软件存储库：

```
[root@monitor ~]# yum install yum-utils vim -y
[root@monitor ~]# yum-config-manager --disable epel
```

流程

对于 CDN 安装：

在 **Ansible 管理节点上**，启用 Red Hat Ceph Storage 3 Tools 存储库和 Ansible 存储库：

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms --
enable=rhel-7-server-ansible-2.6-rpms
```

对于 ISO 安装：

1. 登录红帽客户门户。
2. 点 **Downloads** 访问 **Software & Download**Center。
3. 在 Red Hat Ceph Storage 区域中，单击 **Download Software** 以下载最新版本的软件。

其它资源

- Red Hat Enterprise Linux 系统管理员指南中的注册和管理订阅章节。

返回要求清单

2.6. 将 RAID 控制器用于 OSD 节点的注意事项（可选）

如果 OSD 节点安装了 1-2GB 缓存，启用回写缓存可能会导致小 I/O 写入吞吐量增加。但是，缓存必须具有非易失性。

大多数现代 RAID 控制器都具有超大容量，在出现电源不足时有足够的能力为非易失性 NAND 内存排空易失性内存。务必要了解特定控制器及其固件在恢复电源后的行为。

有些 RAID 控制器需要手动干预。硬盘驱动器通常会向操作系统播发其磁盘缓存，无论是默认应启用或禁用其磁盘缓存。但是，某些 RAID 控制器和某些固件不提供此类信息。验证磁盘级别的缓存是否已禁用，以避免文件系统损坏。

为启用了回写缓存的每个 Ceph OSD 数据驱动器创建一个 RAID 0 卷。

如果 RAID 控制器中也存在 Serial Attached SCSI(SAS)或 SATA 连接的 Solid-state Drive(SSD)磁盘，然后调查控制器和固件是否支持*透传（pass-through）*模式。启用*透传*模式有助于避免缓存逻辑，通常会降低快速介质的延迟。

返回要求清单

2.7. 在对象网关中使用 NVME 的注意事项（可选）

如果您计划使用 Red Hat Ceph Storage 的 Object Gateway 功能，且您的 OSD 节点基于 NVMe SSD 或 SATA SSD，请考虑用于生产环境的 Ceph Object Gateway 中的以最佳方式使用带有 LVM 的 NVMe。这些步骤解释了如何使用专门设计的 Ansible playbook 将日志和 bucket 索引放在 SSD 上，这可以提高性能，与将所有日志放在一个设备上相比。应当结合本安装指南引用有关在 LVM 中使用 NVMe 的信息。

返回要求清单

2.8. 验证 RED HAT CEPH STORAGE 的网络配置

所有红帽 Ceph 存储(RHCS)节点都需要一个公共网络。您必须具有一个网络接口卡，配置为一个公共网络，Ceph 客户端可以访问 Ceph 监视器和 Ceph OSD 节点。

您可能有一个用于集群网络的网络接口卡，以便 Ceph 可以在独立于公共网络的网络上执行心跳、对等、复制和恢复。

配置网络接口设置，并确保这些更改永久保留。

重要

红帽不推荐为公共和专用网络使用单个网络接口卡。

先决条件

- 连接到网络的网络接口卡。

流程

以 **root** 用户身份，在存储集群中的所有 RHCS 节点上执行以下步骤。

1. 验证对应于面向公共的网络接口卡的 **/etc/sysconfig/network-scripts/ifcfg-*** 文件中是否有以下设置：

    a. 对于静态 IP 地址，把 **BOOTPROTO** 参数设置为 **none**。

b. **ONBOOT** 参数必须设置为 **yes**。
如果设为 **no**，Ceph 存储集群在重启后可能无法成为 peer。

c. 如果要使用 IPv6，您必须将 IPv6 参数（如 **IPV6INIT** ）设置为 **yes**，**IPV6_FAILURE_FATAL** 参数除外。
此外，编辑 Ceph 配置文件 **/etc/ceph/ceph.conf**，以指示 Ceph 使用 IPv6，否则 Ceph 会使用 IPv4。

其它资源

- 有关为 Red Hat Enterprise Linux 7 配置网络接口脚本的详情，请参考 Red Hat Enterprise Linux 7 *网络指南*中的*使用 ifcfg 文件配置网络接口*一章。
- 有关网络配置的更多信息，请参见 Red Hat Ceph Storage 3 *配置指南*中的*网络配置参考*一章。

返回要求清单

2.9. 为 RED HAT CEPH STORAGE 配置防火墙

Red Hat Ceph Storage (RHCS) 使用 **firewalld** 服务。

Monitor 守护进程使用端口 **6789** 用于在 Ceph 存储群集内进行通信。

在每个 Ceph OSD 节点上，OSD 守护进程使用 **6800-7300** 范围内的多个端口：

- 一个用于通过公共网络与客户端通信和监控器
- 一个用于通过集群网络发送数据到其他 OSD（如果可用）；否则，通过公共网络发送数据
- 一个用于通过集群网络（如果有）交换心跳数据包；否则，通过公共网络交换。

Ceph 管理器 (**ceph-mgr**) 守护进程使用范围为 **6800-7300** 的端口。考虑将 **ceph-mgr** 守护进程与 Ceph monitor 在同一节点上并置。

Ceph 元数据服务器节点(**ceph-mds**)使用范围为 **6800-7300** 的端口。

Ceph 对象网关节点由 Ansible 配置为使用默认端口 **8080**。但是，您可以更改默认端口，例如端口 **80**。

要使用 SSL/TLS 服务，请打开端口 **443**。

前提条件

- 网络硬件已连接。

流程

以 **root** 用户身份运行以下命令：

1. 在所有 RHCS 节点上，启动 **firewalld** 服务。启用它在引导时运行，并确保它正在运行：

```
# systemctl enable firewalld
# systemctl start firewalld
# systemctl status firewalld
```

2. 在所有 monitor 节点上，打开公共网络中的端口 **6789** ：

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6789/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6789/tcp --permanent
```

根据源地址限制访问：

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="6789" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="6789" accept" --permanent
```

替换

- **ip_address**，带有 monitor 节点的网络地址。
- **netmask_prefix**，使用 CIDR 表示法的子网掩码 。

示例

```
[root@monitor ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.11/24" port protocol="tcp" \
port="6789" accept"
```

```
[root@monitor ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.11/24" port protocol="tcp" \
port="6789" accept" --permanent
```

3. 在所有 OSD 节点上，打开公共网络上的端口 **6800-7300** ：

```
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

如果您有单独的集群网络，请对适当的区重复这些命令。

4. 在所有 Ceph Manager(**ceph-mgr**)节点上（通常与 monitor 节点相同），在公共网络上打开端口 **6800-7300** ：

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

如果您有单独的集群网络，请对适当的区重复这些命令。

5. 在所有 Ceph 元数据服务器(**ceph-mds**)节点上，在公共网络上打开端口 **6800** ：

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800/tcp --permanent
```

如果您有单独的集群网络，请对适当的区重复这些命令。

6. 在所有 Ceph 对象网关节点上，打开公共网络上的相关端口或端口。

a. 打开默认 Ansible 配置的端口 **8080**:

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

根据源地址限制访问：

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="8080" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="8080" accept" --permanent
```

替换

- **ip_address**，带有对象网关节点的网络地址。
- **netmask_prefix**，使用 CIDR 表示法的子网掩码 。

示例

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept" --permanent
```

b. 可选。如果您使用 Ansible 安装 Ceph 对象网关，并将 Ansible 用于配置 Ceph 对象网关的默认端口从 **8080** 改为其他端口（例如 **80**），打开此端口：

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
```

要根据源地址限制访问，请运行以下命令：

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="80" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="80" accept" --permanent
```

替换

- **ip_address**，带有对象网关节点的网络地址。
- **netmask_prefix**，使用 CIDR 表示法的子网掩码 。

示例

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept" --permanent
```

c. 可选。要使用 SSL/TLS，请打开端口 **443** ：

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp --permanent
```

要根据源地址限制访问，请运行以下命令：

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="443" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="443" accept" --permanent
```

替换

- **ip_address**，带有对象网关节点的网络地址。
- **netmask_prefix**，使用 CIDR 表示法的子网掩码 。

示例

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept"
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept" --permanent
```

其它资源

- 如需有关公共和集群网络的更多信息，请参阅验证 Red Hat Ceph Storage 的网络配置 。
- 有关 **firewalld** 的详情，请查看 Red Hat Enterprise Linux 7 安全指南中的 使用防火墙章节。

返回要求清单

2.10. 创建具有 SUDO 访问权限的 ANSIBLE 用户

Ansible 必须能够以具有 **root** 权限的用户身份登录所有 Red Hat Ceph Storage 节点，以便在不提示输入密码的情况下安装软件和创建配置文件。在使用 Ansible 部署和配置 Red Hat Ceph Storage 集群时，您必须在存储集群的所有节点上创建一个没有密码的 **root** 用户。

前提条件

- 对存储集群中的所有节点具有 **root** 或 **sudo** 访问权限。

流程

1. 以 **root** 用户身份登录 Ceph 节点：

```
ssh root@$HOST_NAME
```

替换

- **$HOST_NAME**，其主机名为 Ceph 节点。

示例

```
# ssh root@mon01
```

出现提示时，输入 **root** 密码。

2. 创建一个新的 Ansible 用户：

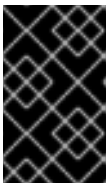
```
adduser $USER_NAME
```

替换

- **$USER_NAME**，它具有 Ansible 用户的新用户名。

示例

```
# adduser admin
```



重要

不要使用 **ceph** 作为用户名。**ceph** 用户名保留用于 Ceph 守护进程。整个集群中的统一用户名可以提高易用性，但避免使用明显的用户名，因为入侵者通常使用它们进行暴力攻击。

3. 为这个用户设置一个新密码：

```
# passwd $USER_NAME
```

替换

- **$USER_NAME**，它具有 Ansible 用户的新用户名。

示例

```
# passwd admin
```

出现提示时，输入新密码两次。

4. 为新创建的用户配置 **sudo** 访问权限：

```
cat << EOF >/etc/sudoers.d/$USER_NAME
$USER_NAME ALL = (root) NOPASSWD:ALL
EOF
```

替换

- **$USER_NAME**，它具有 Ansible 用户的新用户名。

示例

```
# cat << EOF >/etc/sudoers.d/admin
admin ALL = (root) NOPASSWD:ALL
EOF
```

5. 为新文件分配正确的文件权限：

```
chmod 0440 /etc/sudoers.d/$USER_NAME
```

替换

- **$USER_NAME**，它具有 Ansible 用户的新用户名。

示例

```
# chmod 0440 /etc/sudoers.d/admin
```

其它资源

- Red Hat Enterprise Linux 7 *系统管理员指南* 中的 Adding a New User 部分。

返回要求清单

2.11. 为 ANSIBLE 启用无密码 SSH

在 Ansible 管理节点上生成 SSH 密钥对，并将公钥分发到存储集群中的每个节点，以便 Ansible 可以在不提示输入密码的情况下访问节点。

先决条件

- 创建具有 **sudo** 访问权限的 Ansible 用户。

流程

从 Ansible 管理节点，并以 Ansible 用户身份执行下列步骤。

1. 生成 SSH 密钥对，接受默认文件名并将密语留空：

```
[user@admin ~]$ ssh-keygen
```

2. 将公钥复制到存储集群中的所有节点：

```
ssh-copy-id $USER_NAME@$HOST_NAME
```

替换

- **$USER_NAME**，它具有 Ansible 用户的新用户名。
- **$HOST_NAME**，其主机名为 Ceph 节点。

示例

```
[user@admin ~]$ ssh-copy-id admin@ceph-mon01
```

3. 创建并编辑 **~/.ssh/config** 文件。

重要

通过创建并编辑 **~/.ssh/config** 文件，您不必在每次执行 **ansible-playbook** 命令时指定 **-u $USER_NAME** 选项。

a. 创建 SSH **配置文件** ：

```
[user@admin ~]$ touch ~/.ssh/config
```

b. 打开 **配置文件** 进行编辑。为存储集群中的每个节点设置 **Hostname** 和 **User** 选项：

```
Host node1
  Hostname $HOST_NAME
  User $USER_NAME
Host node2
  Hostname $HOST_NAME
  User $USER_NAME
...
```

替换

- **$HOST_NAME**，其主机名为 Ceph 节点。
- **$USER_NAME**，它具有 Ansible 用户的新用户名。

示例

```
Host node1
  Hostname monitor
  User admin
Host node2
  Hostname osd
  User admin
```

```
Host node3
  Hostname gateway
  User admin
```

4. 为 **~/.ssh/config** 文件设置正确的文件权限：

```
[admin@admin ~]$ chmod 600 ~/.ssh/config
```

其它资源

- **ssh_config(5)** 手册页
- Red Hat Enterprise Linux 7 *系统管理员指南*中的 OpenSSH 章节

返回要求清单

第 3 章 部署 RED HAT CEPH STORAGE

本章介绍如何使用 Ansible 应用来部署 Red Hat Ceph Storage 集群和其他组件，如元数据服务器或 Ceph 对象网关。

- 要安装 Red Hat Ceph Storage 集群，请参阅 第 3.2 节 “安装 Red Hat Ceph Storage 集群” 。
- 要安装元数据服务器，请参阅 第 3.4 节 “安装元数据服务器”。
- 要安装 **ceph-client** 角色，请参阅 第 3.5 节 “安装 Ceph 客户端角色”。
- 要安装 Ceph 对象网关，请参阅 第 3.6 节 “安装 Ceph 对象网关”。
- 若要配置多站点 Ceph 对象网关，请参阅 第 3.6.1 节 “配置多站点 Ceph 对象网关”。
- 要了解 Ansible 的 **--limit** 选项，请参阅 第 3.8 节 “了解 **limit** 选项”。

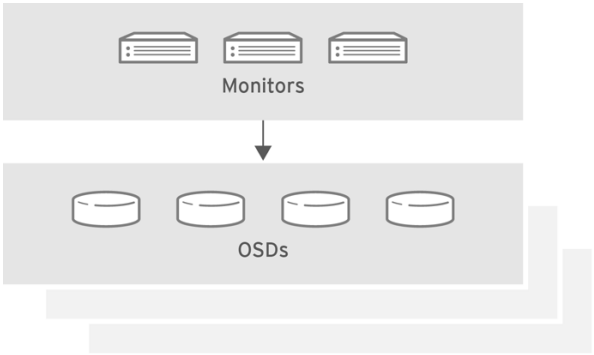
3.1. 先决条件

- 获取有效的客户订阅.
- 准备集群节点。在每个节点中：
  - 将节点注册到 Content Delivery Network (CDN) 并附加订阅 。
  - 启用适当的软件存储库。
  - 创建 Ansible 用户。
  - 启用免密码 SSH 访问。
  - 可选。配置防火墙.

3.2. 安装 RED HAT CEPH STORAGE 集群

将 Ansible 应用与 **ceph-ansible** playbook 搭配使用，以安装 Red Hat Ceph Storage 3。

生产用 Ceph 存储集群从至少三个 monitor 主机开始，以及包含多个 OSD 守护进程的三个 OSD 节点。



先决条件

- 在 Ansible 管理节点上使用 root 帐户安装 **ceph-ansible** 软件包：

```
[root@admin ~]# yum install ceph-ansible
```

流程

从 Ansible 管理节点运行以下命令，除非另有指示。

1. 以 Ansible 用户身份，创建 **ceph-ansible-keys** 目录，其中 Ansible 存储 **ceph-ansible** playbook 生成的临时值。

```
[user@admin ~]$ mkdir ~/ceph-ansible-keys
```

2. 以 root 用户身份，创建一个指向 **/etc/ansible/** 目录中的 **/usr/share/ceph-ansible/group_vars** 目录的符号链接：

```
[root@admin ~]# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

3. 进入 **/usr/share/ceph-ansible/** 目录：

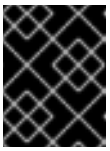
```
[root@admin ~]$ cd /usr/share/ceph-ansible
```

4. 为 **yml.sample** 文件创建新副本：

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp site.yml.sample site.yml
```

5. 编辑复制的文件。

   a. 编辑 **group_vars/all.yml** 文件。下表中列出了要取消注释的最常见必要参数和可选参数。请注意，该表不包含所有参数。



重要

不要将 **cluster: ceph** 参数设置为 **ceph** 以外的任何值，因为不支持使用自定义集群名称。

表 3.1. 常规 Ansible 设置

选项	值	必需	备注
ceph_origin	**repository** 或 **distro** 或 **local**	是	**repository** 代表 Ceph 将通过一个新的仓库安装。**distro** 值意味着不会添加单独的存储库文件，您将获得 Linux 发行版本中包含的任何 Ceph 版本。**local** 值表示将从本地计算机复制 Ceph 二进制文件。

选项	值	必需	备注
ceph_repository_type	**cdn** 或 **iso**	是	
ceph_rhcs_version	**3**	是	
ceph_rhcs_iso_path	ISO 镜像的路径	如果使用 ISO 镜像，为 Yes	
monitor_interface	monitor 节点侦听的接口	**monitor_interface**、**monitoring_address** 或 **monitor_address_block** 是必需的	
monitor_address	monitor 节点侦听的地址		
monitor_address_block	Ceph 公共网络的子网		当节点的 IP 地址未知时使用，但已知子网
ip_version	**ipv6**	如果使用 IPv6 地址，则为	
public_network	Ceph 公共网络的 IP 地址和子网掩码，或者对应的 IPv6 地址（若使用 IPv6）	是	第 2.8 节 “验证 Red Hat Ceph Storage 的网络配置”
cluster_network	Ceph 集群网络的 IP 地址和子网掩码	否，默认为 **public_network**	
configure_firewall	Ansible 将尝试配置适当的防火墙规则	否，将值设为 **true** 或 **false**。	

all.yml 文件的示例如下：

```
ceph_origin: distro
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 3
monitor_interface: eth0
public_network: 192.168.0.0/24
```

注意

务必将 **ceph_origin** 设置为 **all.yml** 文件中的 **distro**。这样可确保安装过程使用正确的下载存储库。

注意

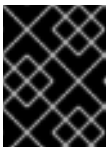
将 **ceph_rhcs_version** 选项设置为 **3** 将引入最新版本的 Red Hat Ceph Storage 3。

警告

默认情况下，Ansible 会尝试重启已安装但屏蔽的 **firewalld** 服务，这可能会导致 Red Hat Ceph Storage 部署失败。要临时解决这个问题，请在 **all.yml** 文件中将 **configure_firewall** 选项设置为 **false**。如果您正在运行 **firewalld** 服务，则不需要在 **all.yml** 文件中使用 **configure_firewall** 选项。

如需了解更多详细信息，请参阅 **all.yml** 文件。

b. 编辑 **group_vars/osds.yml** 文件。下表中列出了要取消注释的最常见必要参数和可选参数。请注意，该表不包含所有参数。



重要

使用不同的物理设备来安装与安装操作系统的设备不同的 OSD。在操作系统和 OSD 之间共享相同的设备会导致性能问题。

表 3.2. OSD Ansible 设置

选项	值	必需	备注
osd_scenario	**collocated** 使用相同的设备进行写入日志记录和键/值数据(BlueStore)或日志(FileStore)和 OSD 数据 **non-collocated** 为使用专用设备，如 SSD 或 NVMe 介质，以存储 write-ahead 日志和键/值数据(BlueStore)或日志数据(FileStore) **LVM** 使用逻辑卷管理器存储 OSD 数据	是	使用 **osd_scenario: non-collocated** 时，**ceph-ansible** 期望 **devices** 和 **dedicated_devices** 中的变量数量相匹配。例如，如果您在 **devices** 中指定了 10 个磁盘，则必须在 **dedicated_devices** 中指定 10 个条目。
osd_auto_discovery	**true** 来自动发现 OSD	如果使用 **osd_scenario: collocated** 为 Yes	使用 **devices** 设置时无法使用

选项	值	必需	备注
devices	存储 **Ceph 数据** 的设备列表	Yes 用来指定设备列表	使用 **osd_auto_discovery** 设置时无法使用。当使用 **lvm** 作为 **osd_scenario** 并设置 **devices** 选项时，**ceph-volume lvm batch** 模式将创建优化的 OSD 配置。
dedicated_devices	存储 **ceph 日志** 的非并置 OSD 的专用设备列表	如果 **osd_scenario: non-collocated**，则为 yes	应该是非分区的设备
dmcrypt	**true** 来加密 OSD	否	默认值为 **false**
lvm_volumes	FileStore 或 BlueStore 字典列表	如果使用 **osd_scenario: lvm**且存储设备没有使用 **devices**定义 时为 Yes	每一字典必须包含 **data**、**journal** 和 **data_vg** 键。任何逻辑卷或卷组都必须是名称，而不是完整路径。**data** 和 **journal** 键可以是逻辑卷 (LV) 或分区，但不能将一个日志用于多个 **data** LV。**data_vg** 键必须是包含 **data** LV 的卷组。（可选）**journal_vg** 键可用于指定包含 journal LV 的卷组（如果适用）。有关各种支持的配置，请参见以下示例。
osds_per_device	每个设备要创建的 OSD 数量。	否	默认为 **1**
osd_objectstore	OSD 的 Ceph 对象存储类型。	否	默认为 **bluestore**。另一个选项是 **filestore**。升级需要。

以下是使用三种 OSD 方案 (**collocated**, **non-collocated**, 和 **lvm**) 的**osds.yml** 文件的示例：如果没有指定，默认的 OSD 对象存储格式为 BlueStore。

Collocated

-

```
osd_objectstore: filestore
osd_scenario: collocated
devices:
  - /dev/sda
  - /dev/sdb
```

Non-collocated - BlueStore

```
osd_objectstore: bluestore
osd_scenario: non-collocated
devices:
 - /dev/sda
 - /dev/sdb
 - /dev/sdc
 - /dev/sdd
dedicated_devices:
 - /dev/nvme0n1
 - /dev/nvme0n1
 - /dev/nvme1n1
 - /dev/nvme1n1
```

此 non-collocated 示例将创建四个 BlueStore OSD，每个设备一个。在本例中，传统的硬盘驱动器（**sda**, **sdb**, **sdc**, **sdd**）用于对象数据，以及固态驱动器(SSD)（**/dev/nvme0n1**、**/dev/nvme1n1**）用于 BlueStore 数据库和 write-ahead 日志。此配置将 **/dev/sda** 和 **/dev/sdb** 设备与 **/dev/nvme0n1** 设备配对，并将 **/dev/sdc** 和 **/dev/sdd** 设备与 **/dev/nvme1n1** 设备配对。

non-collocated - FileStore

```
osd_objectstore: filestore
osd_scenario: non-collocated
devices:
  - /dev/sda
  - /dev/sdb
  - /dev/sdc
  - /dev/sdd
dedicated_devices:
   - /dev/nvme0n1
   - /dev/nvme0n1
   - /dev/nvme1n1
   - /dev/nvme1n1
```

LVM 简单

```
osd_objectstore: bluestore
osd_scenario: lvm
devices:
  - /dev/sda
  - /dev/sdb
```

或者

```
osd_objectstore: bluestore
osd_scenario: lvm
```

```
devices:
  - /dev/sda
  - /dev/sdb
  - /dev/nvme0n1
```

使用这些简单的配置，**ceph-ansible** 使用批处理模式(**ceph-volume lvm batch**)来创建 OSD。

在第一个场景中，如果 **devices** 是传统的硬盘驱动器或 SSD，则每个设备会创建一个 OSD。

在第二种场景中，当结合了传统的硬盘驱动器和 SSD 时，数据将放置在传统的硬盘驱动器（**sda**、**sdb**）上，并且将最大型的 BlueStore 数据库(**block.db**)在 SSD(**nvme0n1**)上创建。

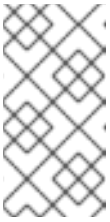
LVM 高级设置

```
osd_objectstore: filestore
osd_scenario: lvm
lvm_volumes:
  - data: data-lv1
    data_vg: vg1
    journal: journal-lv1
    journal_vg: vg2
  - data: data-lv2
    journal: /dev/sda
    data_vg: vg1
```

或者

```
osd_objectstore: bluestore
osd_scenario: lvm
lvm_volumes:
  - data: data-lv1
    data_vg: data-vg1
    db: db-lv1
    db_vg: db-vg1
    wal: wal-lv1
    wal_vg: wal-vg1
  - data: data-lv2
    data_vg: data-vg2
    db: db-lv2
    db_vg: db-vg2
    wal: wal-lv2
    wal_vg: wal-vg2
```

使用这些高级场景示例时，必须事先创建卷组和逻辑卷。它们不会由 **ceph-ansible** 创建。



注意

如果使用所有 NVMe SSD，请设置 **osd_scenario: lvm** 和 **osds_per_device: 4** 选项。有关更多信息，请参阅 *安装指南*中的 *Configuring OSD Ansible settings for all NVMe Storage*（Red Hat Enterprise Linux）或 *Configuring OSD Ansible settings for all NVMe Storage*（Ubuntu）。

如需了解更多详细信息，请参阅 **osds.yml** 文件中的注释。

6. 编辑位于 **/etc/ansible/hosts** 的 Ansible 清单文件。记住注释掉示例主机。

   a. 在 **[mons]** 部分下添加 monitor 节点：

   ```
   [mons]
   MONITOR_NODE_NAME1
   MONITOR_NODE_NAME2
   MONITOR_NODE_NAME3
   ```

   b. 在 **[osds]** 部分下添加 OSD 节点。如果节点有顺序命名，请考虑使用范围：

   ```
   [osds]
   OSD_NODE_NAME1[1:10]
   ```

   注意

   对于新安装的 OSD，默认的对象存储格式为 BlueStore。

   i. （可选）使用 **devices** 和 **dedicated_devices** 选项指定 OSD 节点使用的设备。使用逗号分隔的列表列出多个设备。

   语法

   ```
   [osds]
   CEPH_NODE_NAME devices="['DEVICE_1', 'DEVICE_2']" dedicated_devices="
   ['DEVICE_3', 'DEVICE_4']"
   ```

   示例

   ```
   [osds]
   ceph-osd-01 devices="['/dev/sdc', '/dev/sdd']" dedicated_devices="['/dev/sda',
   '/dev/sdb']"
   ceph-osd-02 devices="['/dev/sdc', '/dev/sdd', '/dev/sde']" dedicated_devices="
   ['/dev/sdf', '/dev/sdg']"
   ```

   在没有指定设备时，在 **osds.yml** 文件中将 **osd_auto_discovery** 选项设置为 **true**。

   注意

   当 OSD 与不同名称使用设备或者其中一个 OSD 上失败时，使用 **devices** 和 **dedicated_devices** 参数很有用。

7. 另外，如果您想要将主机特定参数用于所有部署(**裸机或在 容器** 中)，请在 **host_vars** 目录中创建主机文件，使其包含特定于主机的参数。

   a. 在 **/etc/ansible/host_vars/** 目录下，为每个添加到存储集群的每个新 Ceph OSD 节点创建一个新文件：

   语法

   ```
   touch /etc/ansible/host_vars/OSD_NODE_NAME
   ```

示例

```
[root@admin ~]# touch /etc/ansible/host_vars/osd07
```

b. 使用特定于主机的参数更新 文件。**在裸机部署中**，您可以在 文件中添加 **devices:** 和 **dedicated_devices:** 部分。

示例

```
devices:
  - /dev/sdc
  - /dev/sdd
  - /dev/sde
  - /dev/sdf

dedicated_devices:
  - /dev/sda
  - /dev/sdb
```

8. 另外，对于所有部署（**裸机** 或 **容器** ），您可以使用 **ansible-playbook** 创建自定义 CRUSH 层次结构：

a. 设置 Ansible 清单文件。使用 **osd_crush_location** 参数，指定 OSD 主机处于 CRUSH map 的层次结构中的位置。您必须指定至少两种 CRUSH bucket 类型来指定 OSD 的位置，一种 bucket 类型 必须是 host。默认情况下，包括 **root**, **datacenter**, **room**, **row**, **pod**, **pdu**, **rack**, **chassis** 和 **host**。

语法

```
[osds]
CEPH_OSD_NAME osd_crush_location="{ 'root': ROOT_BUCKET', 'rack':
'RACK_BUCKET', 'pod': 'POD_BUCKET', 'host': 'CEPH_HOST_NAME' }"
```

示例

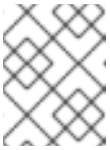
```
[osds]
ceph-osd-01 osd_crush_location="{ 'root': 'default', 'rack': 'rack1', 'pod': 'monpod', 'host':
'ceph-osd-01' }"
```

b. 将 **crush_rule_config** 和 **create_crush_tree** 参数设置为 **True**，如果您不想使用默认的 CRUSH 规则，至少创建一个 CRUSH 规则。例如，如果您使用 **HDD** 设备，请按如下所示编辑参数：

```
crush_rule_config: True
crush_rule_hdd:
    name: replicated_hdd_rule
    root: root-hdd
    type: host
    class: hdd
    default: True
crush_rules:
  - "{{ crush_rule_hdd }}"
create_crush_tree: True
```

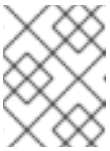
如果使用 **SSD** 设备，请按如下所示编辑参数：

```
crush_rule_config: True
crush_rule_ssd:
   name: replicated_ssd_rule
   root: root-ssd
   type: host
   class: ssd
   default: True
crush_rules:
  - "{{ crush_rule_ssd }}"
create_crush_tree: True
```



注意

如果没有部署 **ssd** 和 **hdd** OSD，则默认 CRUSH 规则会失败，因为默认规则现在包含 类**参数** （必须定义）。



注意

此外，将自定义 CRUSH 层次结构添加到 **host_vars** 目录中的 OSD 文件，如上面的步骤中所述，使此配置正常工作。

c. 使用在 **group_vars/clients.yml** 文件中创建的 **crush_rules** 来创建 **pools**。

示例

```
copy_admin_key: True
user_config: True
pool1:
  name: "pool1"
  pg_num: 128
  pgp_num: 128
  rule_name: "HDD"
  type: "replicated"
  device_class: "hdd"
pools:
  - "{{ pool1 }}"
```

d. 查看树。

```
[root@mon ~]# ceph osd tree
```

e. 验证池。

```
# for i in $(rados lspools);do echo "pool: $i"; ceph osd pool get $i crush_rule;done

pool: pool1
crush_rule: HDD
```

9. 对于 **裸机** 或 **容器** 的所有部署，打开并编辑 Ansible 清单文件（默认为 **/etc/ansible/hosts** 文件）。注释掉示例主机。

a. 在 **[mgrs]** 部分下，添加 Ceph Manager（**ceph-mgr**）节点。将 Ceph 管理器守护进程与 monitor 节点并置。

```
[mgrs]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

10. 以 Ansible 用户身份，确保 Ansible 可以访问 Ceph 主机：

```
[user@admin ~]$ ansible all -m ping
```

11. 将以下行添加到 **/etc/ansible/ansible.cfg** 文件中：

```
retry_files_save_path = ~/
```

12. 以 **root** 用户身份，创建 **/var/log/ansible/** 目录，并为 **ansible** 用户分配适当的权限：

```
[root@admin ~]# mkdir /var/log/ansible
[root@admin ~]# chown ansible:ansible  /var/log/ansible
[root@admin ~]# chmod 755 /var/log/ansible
```

a. 编辑 **/usr/share/ceph-ansible/ansible.cfg** 文件，更新 **log_path** 值，如下所示：

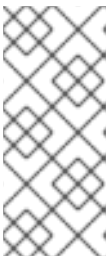
```
log_path = /var/log/ansible/ansible.log
```

13. 以 Ansible 用户身份，切换到 **/usr/share/ceph-ansible/** 目录：

```
[user@admin ~]$ cd /usr/share/ceph-ansible/
```

14. 运行 **ceph-ansible** playbook：

```
[user@admin ceph-ansible]$ ansible-playbook site.yml
```



注意

要提高部署速度，请在 **ansible-playbook** 中使用 **--forks** 选项。默认情况下，**ceph-ansible** 将 fork 设置为 **20**。在这个版本中，最多 20 个节点将同时安装。要一次安装最多 30 个节点，请运行 **ansible-playbook --forks 30 *PLAYBOOK 文件***。必须监控管理节点上的资源，以确保它们不会被过度使用。如果是，则减少传递给 **--forks** 的数字。

15. 使用 monitor 节点上的 root 帐户，验证 Ceph 集群的状态：

```
[root@monitor ~]# ceph health
HEALTH_OK
```

16. 使用 **rados** 验证群集是否正常运行。

a. 在监控节点上，创建一个包含八个 PG 的测试池：
语法

```
[root@monitor ~]# ceph osd pool create <pool-name> <pg-number>
```

示例

```
[root@monitor ~]# ceph osd pool create test 8
```

b. 创建名为 **hello-world.txt** 的文件：
语法

```
[root@monitor ~]# vim <file-name>
```

示例

```
[root@monitor ~]# vim hello-world.txt
```

c. 使用对象名称 **hello-world** 将 **hello-world.txt** 上传到测试池中：
语法

```
[root@monitor ~]# rados --pool <pool-name> put <object-name> <object-file>
```

示例

```
[root@monitor ~]# rados --pool test put hello-world hello-world.txt
```

d. 从 test 池下载 **hello-world**，保存为 **fetch.txt** ：
语法

```
[root@monitor ~]# rados --pool <pool-name> get <object-name> <object-file>
```

示例

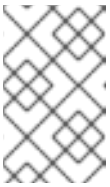
```
[root@monitor ~]# rados --pool test get hello-world fetch.txt
```

e. 检查 **fetch.txt** 的内容：

```
[root@monitor ~]# cat fetch.txt
```

输出应该是：

```
"Hello World!"
```



注意

除了验证集群状态外，您还可以使用 **ceph-medic** 实用程序来全面诊断 Ceph 存储群集。请参阅 Red Hat Ceph Storage 3 *管理指南*中的*使用 **ceph-medic** 诊断 Ceph Storage 集群Cluster*。

3.3. 为所有 NVME 存储配置 OSD ANSIBLE 设置

若要在仅使用非易失性内存表达(NVMe)设备进行存储时优化性能，可在每个 NVMe 设备上配置四个 OSD。通常，每个设备仅配置一个 OSD，这将充分利用 NVMe 设备的吞吐量。

注意

如果混合了 SSD 和 HDD，则 SSD 将用于日志或 **block.db**，而非 OSD。

注意

在测试过程中，发现每个 NVMe 设备上配置四个 OSD 可提供最佳性能。建议设置 **osds_per_device: 4**，但这不是强制要求。其他值可以在您的环境中提供更好的性能。

先决条件

- 满足 Ceph 群集的所有软件和硬件要求.

流程

1. 在 **group_vars/osds.yml** 中设置 **osd_scenario: lvm** 和 **osds_per_device: 4**：

```
osd_scenario: lvm
osds_per_device: 4
```

2. 列出 **devices** 中的 NVMe 设备：

```
devices:
  - /dev/nvme0n1
  - /dev/nvme1n1
  - /dev/nvme2n1
  - /dev/nvme3n1
```

3. **group_vars/osds.yml** 中的设置类似以下示例：

```
osd_scenario: lvm
osds_per_device: 4
devices:
  - /dev/nvme0n1
  - /dev/nvme1n1
  - /dev/nvme2n1
  - /dev/nvme3n1
```

注意

您必须将 **devices** 用于此配置，而不是使用 **lvm_volumes**。这是因为 **lvm_volumes** 通常与预先创建的逻辑卷一起使用，而 **osds_per_device** 则表示 Ceph 自动创建逻辑卷。

其它资源

- 在 Red Hat Enterprise Linux 上安装 Red Hat Ceph Storage 集群
- 在 Ubuntu 上安装 Red Hat Ceph Storage 集群

3.4. 安装元数据服务器

使用 Ansible 自动化应用安装 Ceph 元数据服务器 (MDS)。元数据服务器守护进程是部署 Ceph 文件系统所必需的。

先决条件

- 一个正常工作的 Red Hat Ceph Storage 集群。

流程

在 Ansible 管理节点上执行下列步骤。

1. 在 **/etc/ansible/hosts** 文件中添加新部分 **[mdss]** ：

```
[mdss]
hostname
hostname
hostname
```

   使用您要安装 Ceph 元数据服务器的节点的主机名替换 *hostname*。

2. 进入 **/usr/share/ceph-ansible** 目录：

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. 可选。更改默认变量。

   a. 创建名为 **mdss.yml** 的 **group_vars/mdss.yml** 的副本：

```
[root@admin ceph-ansible]# cp group_vars/mdss.yml.sample group_vars/mdss.yml
```

   b. （可选）编辑 **mdss.yml** 中的参数。详情请查看 **mdss.yml**。

4. 以 Ansible 用户身份，运行 Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit mdss
```

5. 安装元数据服务器后，对它们进行配置。详情请参阅 Red Hat Ceph Storage 3 的 Ceph 文件系统指南中的*配置元数据服务器守护进程* 一章。

其它资源

- Red Hat Ceph Storage 3 的 *Ceph 文件系统指南*
- *了解 **限制** 选项*

3.5. 安装 CEPH 客户端角色

ceph-ansible 实用程序提供 **ceph-client** 角色，将 Ceph 配置文件和管理密钥环复制到节点。此外，您还可以使用此角色创建自定义池和客户端。

先决条件

- 正在运行的 Ceph 存储集群，最好处于 **active + clean** 状态。

- 执行 第 2 章 *安装 Red Hat Ceph Storage 的要求* 中列出的任务。

流程

在 Ansible 管理节点上执行下列任务：

1. 在 **/etc/ansible/hosts** 文件中添加新部分 **[clients]**：

```
[clients]
<client-hostname>
```

将 **<client-hostname>** 替换为您要安装 **ceph-client** 角色的节点的主机名。

2. 进入 **/usr/share/ceph-ansible** 目录：

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. 为 **clients.yml.sample** 文件创建一个新副本，取名为 **client.yml**:

```
[root@admin ceph-ansible ~]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4. 打开 **group_vars/clients.yml** 文件，取消注释以下行：

```
keys:
  - { name: client.test, caps: { mon: "allow r", osd: "allow class-read object_prefix
rbd_children, allow rwx pool=test" },  mode: "{{ ceph_keyring_permissions }}" }
```

a. 使用实际客户端名称替换 **client.test**，并在客户端定义行中添加客户端密钥，例如：

```
key: "ADD-KEYRING-HERE=="
```

现在，整行示例类似如下：

```
- { name: client.test, key: "AQAin8tUMICVFBAALRHNrV0Z4MXupRw4v9JQ6Q==", caps:
{ mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" },
mode: "{{ ceph_keyring_permissions }}" }
```

注意

ceph-authtool --gen-print-key 命令可以生成新的客户端密钥。

5. （可选）指示 **ceph-client** 创建池和客户端。

a. 更新 **clients.yml**。

- 取消注释 **user_config** 设置并将其设置为 **true**。
- 取消注释 **pools** 和 **keys** 部分，并根据需要进行更新。您可以使用 **cephx** 功能定义自定义池和客户端名称。

b. 将 **osd_pool_default_pg_num** 设置添加到 **all.yml** 文件的 **ceph_conf_overrides** 部分：

```
ceph_conf_overrides:
  global:
    osd_pool_default_pg_num: <number>
```

-

  将 **<number>** 替换为 PG 的默认数量。

6. 运行 Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit clients
```

其它资源

- 第 3.8 节 “了解 **limit** 选项”

3.6. 安装 CEPH 对象网关

Ceph 对象网关（也称为 RADOS 网关）是在 **librados** API 基础上构建的对象存储接口，为应用提供 Ceph 存储集群的 RESTful 网关。

先决条件

- 正在运行一个 Red Hat Ceph Storage 集群，最好处于 **active + clean** 状态。
- 在 Ceph 对象网关节点上，执行 第 2 章 *安装 Red Hat Ceph Storage 的要求* 中列出的任务。

流程

在 Ansible 管理节点上执行下列任务：

1. 将网关主机添加到 **[rgws]** 部分下的 **/etc/ansible/hosts** 文件中，以将其角色标识到 Ansible。如果主机有顺序命名，请使用范围，例如：

```
[rgws]
<rgw_host_name_1>
<rgw_host_name_2>
<rgw_host_name[3..10]>
```

2. 进入 Ansible 配置目录：

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

3. 从示例文件创建 **rgws.yml** 文件：

```
[root@ansible ~]# cp group_vars/rgws.yml.sample group_vars/rgws.yml
```

4. 打开并编辑 **group_vars/rgws.yml** 文件。要将管理员密钥复制到 Ceph 对象网关节点，取消注释 **copy_admin_key** 选项：

```
copy_admin_key: true
```

5. **rgws.yml** 文件可以指定与默认端口 **7480** 不同的默认端口。例如：

```
ceph_rgw_civetweb_port: 80
```

6. **all.yml** 文件必须 指定一个 **radosgw_interface**。例如：

```
radosgw_interface: eth0
```

指定该接口可防止 Civetweb 在同一主机上运行多个实例时绑定到与另一个 Civetweb 实例相同的 IP 地址。

7. 通常，要更改默认设置，请取消注释 **rgw.yml** 文件中的设置，并相应地进行更改。要对没有在 **rgw.yml** 文件中的设置进行其他更改，请在 **all.yml** 文件中使用 **ceph_conf_overrides:**。例如，将 **rgw_dns_name:** 设置为 DNS 服务器的主机，并确保集群的 DNS 服务器配置为启用 S3 子域。

```
ceph_conf_overrides:
  client.rgw.rgw1:
    rgw_dns_name: <host_name>
    rgw_override_bucket_index_max_shards: 16
    rgw_bucket_default_quota_max_objects: 1638400
```

有关高级配置详情，请参阅 Red Hat Ceph Storage 3 *Ceph Object Gateway for Production* 指南。高级议题包括：

- *配置 Ansible 组*
- *开发存储策略*.如需有关如何创建和配置池的更多详细信息，请参阅*创建根池*、*创建系统池*和*创建数据放置策略*部分。
  有关存储桶分片的配置详情，请参阅 Bucket Sharding。

8. 取消注释 **group_vars/all.yml** 文件中的 **radosgw_interface** 参数。

```
radosgw_interface: <interface>
```

替换：

- 使用 Ceph 对象网关节点侦听的接口替换 **<interface>**

如需了解更多详细信息，请参阅 **all.yml** 文件。

9. 运行 Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit rgws
```

注意

Ansible 确保每个 Ceph 对象网关正在运行。

对于单一站点配置，请将 Ceph 对象网关添加到 Ansible 配置。

对于多站点部署，您应该为每个区域都有一个 Ansible 配置。也就是说，Ansible 将为该区创建一个 Ceph 存储集群和网关实例。

在多站点集群安装完成后，请查阅 *Red Hat Enterprise Linux 的对象网关指南* 中的 多站点一章，以了解有关为多站点配置集群的详细信息。

其它资源

- 第 3.8 节 “了解 **limit** 选项”
- *Red Hat Enterprise Linux 对象网关指南*

3.6.1. 配置多站点 Ceph 对象网关

Ansible 将配置 realm、zonegroup，以及用于多站点环境中的 Ceph 对象网关的主和次要区域。

先决条件

- 两个正在运行的 Red Hat Ceph Storage 集群。
- 在 Ceph 对象网关节点上，执行 *Red Hat Ceph Storage 安装指南* 中的 *安装 Red Hat Ceph Storage 要求* 一节中列出的任务。
- 安装和配置每个存储群集一个 Ceph 对象网关。

流程

1. 在 Ansible 节点上为主存储集群执行以下步骤：

   a. 生成系统密钥并将其输出捕获至 **multi-site-keys.txt** 文件中：

   ```
   [root@ansible ~]# echo system_access_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 20 | head -n 1) > multi-site-keys.txt
   [root@ansible ~]# echo system_secret_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 40 | head -n 1) >> multi-site-keys.txt
   ```

   b. 进入 Ansible 配置目录 **/usr/share/ceph-ansible** ：

   ```
   [root@ansible ~]# cd /usr/share/ceph-ansible
   ```

   c. 打开并编辑 **group_vars/all.yml** 文件。通过添加下列选项启用多站点支持，并相应地更新 **$ZONE_NAME**、**$ZONE_GROUP_NAME**、**$REALM_NAME**、**$ACCESS_KEY** 和 **$SECRET_KEY** 值。
   当多个 Ceph 对象网关位于 master 区域中时，需要设置 **rgw_multisite_endpoints** 选项。**rgw_multisite_endpoints** 选项的值是一个逗号分隔的列表，没有空格。

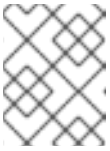
   示例

   ```
   rgw_multisite: true
   rgw_zone: $ZONE_NAME
   rgw_zonemaster: true
   rgw_zonesecondary: false
   rgw_multisite_endpoint_addr: "{{ ansible_fqdn }}"
   rgw_multisite_endpoints:
   http://foo.example.com:8080,http://bar.example.com:8080,http://baz.example.com:8080
   rgw_zonegroup: $ZONE_GROUP_NAME
   rgw_zone_user: zone.user
   rgw_realm: $REALM_NAME
   system_access_key: $ACCESS_KEY
   system_secret_key: $SECRET_KEY
   ```

   注意

   ansible_fqdn 域名必须从辅助存储集群解析。



注意

添加新对象网关时，请先使用新对象网关的端点 URL 将它附加到 **rgw_multisite_endpoints** 列表的末尾，然后再运行 Ansible playbook。

d. 运行 Ansible playbook:

```
[user@ansible ceph-ansible]$ ansible-playbook site.yml --limit rgws
```

e. 重启 Ceph 对象网关守护进程：

```
[root@rgw ~]# systemctl restart ceph-radosgw@rgw.`hostname -s`
```

2. 在 Ansible 节点上为次要存储集群执行以下步骤：

a. 进入 Ansible 配置目录 **/usr/share/ceph-ansible** ：

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

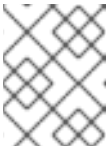
b. 打开并编辑 **group_vars/all.yml** 文件。通过添加下列选项并同时更新 **$ZONE_NAME**、**$ZONE_GROUP_NAME**、**$REALM_NAME**、**$ACCESS_KEY** 和 **$SECRET_KEY** 的值来启用多站点支持：**rgw_zone_user**, **system_access_key**, 和 **system_secret_key** 的值必须与 master zone 配置中使用的值相同。 **rgw_pullhost** 选项必须是 master 区域的 Ceph 对象网关。
当多个 Ceph 对象网关位于 second 区域中时，需要设置 **rgw_multisite_endpoints** 选项。**rgw_multisite_endpoints** 选项的值是一个逗号分隔的列表，没有空格。

示例

```
rgw_multisite: true
rgw_zone: $ZONE_NAME
rgw_zonemaster: false
rgw_zonesecondary: true
rgw_multisite_endpoint_addr: "{{ ansible_fqdn }}"
rgw_multisite_endpoints:
http://foo.example.com:8080,http://bar.example.com:8080,http://baz.example.com:8080
rgw_zonegroup: $ZONE_GROUP_NAME
rgw_zone_user: zone.user
rgw_realm: $REALM_NAME
system_access_key: $ACCESS_KEY
system_secret_key: $SECRET_KEY
rgw_pull_proto: http
rgw_pull_port: 8080
rgw_pullhost: $MASTER_RGW_NODE_NAME
```

注意

ansible_fqdn 域名必须可从主存储集群解析。



注意

添加新对象网关时，请先使用新对象网关的端点 URL 将它附加到 **rgw_multisite_endpoints** 列表的末尾，然后再运行 Ansible playbook。

c. 运行 Ansible playbook:

```
[user@ansible ceph-ansible]$ ansible-playbook site.yml --limit rgws
```

d. 重启 Ceph 对象网关守护进程：

```
[root@rgw ~]# systemctl restart ceph-radosgw@rgw.`hostname -s`
```

3. 在主控机和次要存储集群上运行 Ansible playbook 后，您将拥有运行中的主动 Ceph 对象网关配置。

4. 验证多站点 Ceph 对象网关配置：

a. 在每个站点（主要和次要）的 Ceph 监控和对象网关节点中，必须能够对另一站点进行 **curl**。

b. 对两个站点运行 **radosgw-admin sync status** 命令。

3.7. 安装 NFS-GANESHA 网关

Ceph NFS Ganesha 网关是在 Ceph 对象网关基础上构建的 NFS 接口，为应用提供 POSIX 文件系统接口到 Ceph 对象网关，以便在文件系统内将文件迁移到 Ceph 对象存储。

先决条件

- 正在运行的 Ceph 存储集群，最好处于 **active + clean** 状态。
- 至少一个运行 Ceph 对象网关的节点.
- 执行开始前中的步骤。

流程

在 Ansible 管理节点上执行下列任务：

1. 从示例文件创建 **nfss** 文件：

```
[root@ansible ~]# cd /usr/share/ceph-ansible/group_vars
[root@ansible ~]# cp nfss.yml.sample nfss.yml
```

2. 将网关主机添加到 **[nfss]** 组下的 **/etc/ansible/hosts** 文件中，以识别其组成员资格。如果主机具有连续命名，则使用范围。例如：

```
[nfss]
<nfs_host_name_1>
<nfs_host_name_2>
<nfs_host_name[3..10]>
```

3. 进入 Ansible 配置目录 **/etc/ansible/**:

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

4. 要将管理员密钥复制到 Ceph 对象网关节点，请取消注释 **/usr/share/ceph-ansible/group_vars/nfss.yml** 文件中的 **copy_admin_key** 设置：

```
copy_admin_key: true
```

5. 配置 **/usr/share/ceph-ansible/group_vars/nfss.yml** 文件的 FSAL (File System Abstraction Layer) 部分。提供 ID、S3 用户 ID、S3 访问密钥和机密。对于 NFSv4，它应类似如下：

```
###################
# FSAL RGW Config #
###################
#ceph_nfs_rgw_export_id: <replace-w-numeric-export-id>
#ceph_nfs_rgw_pseudo_path: "/"
#ceph_nfs_rgw_protocols: "3,4"
#ceph_nfs_rgw_access_type: "RW"
#ceph_nfs_rgw_user: "cephnfs"
# Note: keys are optional and can be generated, but not on containerized, where
# they must be configered.
#ceph_nfs_rgw_access_key: "<replace-w-access-key>"
#ceph_nfs_rgw_secret_key: "<replace-w-secret-key>"
```

警告

访问和密钥是可选的，可以生成。

6. 运行 Ansible playbook:

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit nfss
```

其它资源

- 第 3.8 节 “了解 **limit** 选项”
- *Red Hat Enterprise Linux 对象网关指南*

3.8. 了解 LIMIT 选项

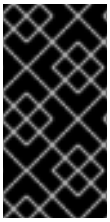
本节包含有关 Ansible **--limit** 选项的信息。

Ansible 支持 **--limit** 选项，允许您将 **site**、**site-docker** 和 **rolling_upgrade** Ansible playbook 用于清单文件的特定部分。

```
$ ansible-playbook site.yml|rolling_upgrade.yml|site-docker.yml --limit
osds|rgws|clients|mdss|nfss|iscsigws
```

例如，若要仅重新部署裸机上的 OSD，请以 Ansible 用户身份运行以下命令：

```
$ ansible-playbook /usr/share/ceph-ansible/site.yml --limit osds
```



重要

如果您在一个节点上并置 Ceph 组件，Ansible 会将 playbook 应用到节点上的所有组件，尽管通过 **limit** 选项仅指定了一种组件类型。例如，如果您在包含 OSD 和元数据服务器 (MDS)的节点上使用 **--limit osds** 选项运行 **rolling_update** playbook，Ansible 将升级组件、OSD 和 MDS。

3.9. 其它资源

- Ansible 文档

第 4 章 升级 RED HAT CEPH STORAGE 集群

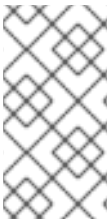
本节论述了如何升级到 Red Hat Ceph Storage 的新主版本或次版本。

- 要升级存储集群，请参阅 第 4.1 节 “升级存储集群”。
- 要升级 Red Hat Ceph Storage Dashboard，请参阅 第 4.2 节 “升级 Red Hat Ceph Storage Dashboard”。

使用管理节点 **/usr/share/ceph-ansible/infrastructure-playbooks/** 目录中的 Ansible **rolling_update.yml** playbook，在 Red Hat Ceph Storage 的两个主要或次要版本间升级，或者应用异步更新。

Ansible 按照以下顺序升级 Ceph 节点：

- 监控节点
- MGR 节点
- OSD 节点
- MDS 节点
- Ceph 对象网关节点
- 所有其他 Ceph 客户端节点

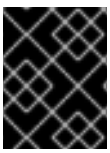


注意

Red Hat Ceph Storage 3 对位于 **/usr/share/ceph-ansible/group_vars/** 目录的 Ansible 配置文件引入了一些更改；某些参数被重命名或删除。因此，在升级到版本 3 后，在从 **all.yml.sample** 和 **osds.yml.sample** 文件创建新副本前，备份 **all.yml** 和 **osds.yml** 文件。有关更改的详情，请查看 附录 H, *版本 2 和版本 3 之间的 Ansible 变量更改*。

注意

Red Hat Ceph Storage 3.1 及更高版本引入了新的 Ansible playbook，以便在使用对象网关和基于 NVMe 的 SSD（及 SATA SSD）时优化存储的性能。Playbook 通过将日志和 bucket 索引放在 SSD 上来实现此目的，与将所有日志放在一个设备上相比，这可以提高性能。这些 playbook 设计为在安装 Ceph 时使用。现有的 OSD 继续工作，升级期间不需要额外的步骤。无法升级 Ceph 集群，同时重新配置 OSD 以优化存储。若要将不同的设备用于日志或 bucket 索引，需要重新调配 OSD。如需更多信息，请参阅生产环境指南中的 Ceph 对象网关中的最佳使用 NVMe。



重要

rolling_update.yml playbook 包含 **serial** 变量，用于调整要同时更新的节点数量。红帽强烈建议使用默认值 (**1**)，以确保 Ansible 逐一升级集群节点。

重要

如果升级在任何点上失败，请使用 **ceph status** 命令检查集群状态以了解升级失败的原因。如果您不确定故障原因及如何解决，请联系 *红帽支持*以获得帮助。

重要

在使用 **rolling_update.yml** playbook 升级到任何 Red Hat Ceph Storage 3.x 版本时，使用 Ceph 文件系统(CephFS)的用户必须手动更新元数据服务器(MDS)集群。这是因为一个已知问题。

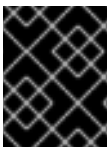
在使用 **ceph- ansible rolling-upgrade.yml 升级整个集群前，注释掉 /etc/ansible /hosts** 中的 MDS 主机，然后手动升级 MDS。在 **/etc/ansible/hosts** 文件中：

```
#[mdss]
#host-abc
```

有关此已知问题的更多详细信息，包括如何更新 MDS 集群，请参阅 Red Hat Ceph Storage 3.0 发行注记。

重要

当将 Red Hat Ceph Storage 从以前的版本升级到版本 3.2 时，Ceph Ansible 配置会将对象存储类型默认设置为 BlueStore。如果您仍然希望将 FileStore 用作 OSD 对象存储，则明确将 Ceph Ansible 配置设置为 FileStore。这可确保新部署和替换的 OSD 使用 FileStore。



重要

在使用 **rolling_update.yml** playbook 升级到任何 Red Hat Ceph Storage 3.x 版本时，如果您使用多站点 Ceph 对象网关配置，则不必手动更新 **all.yml** 文件来指定多站点配置。

先决条件

- 以 **root** 用户身份登录存储集群中的所有节点。
- 在存储集群中的所有节点上，启用 **rhel-7-server-extras-rpms** 存储库。

```
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

- 如果 Ceph 节点没有连接到 Red Hat Content Delivery Network(CDN)，并且您使用 ISO 镜像安装 Red Hat Ceph Storage，请使用最新版本的红帽 Ceph 存储更新本地存储库。详情请查看 第 2.5 节 “启用 Red Hat Ceph Storage Repositories” 。
- 如果从 Red Hat Ceph Storage 2.x 升级到 3.x，在 Ansible 管理节点上和 RBD 镜像节点上，启用 Red Hat Ceph Storage 3 Tools 存储库：

```
# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

- 在 Ansible 管理节点上，启用 Ansible 存储库：

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-ansible-2.6-rpms
```

- 在 Ansible 管理节点上，确保已安装了 **ansible** 和 **ceph-ansible** 软件包的最新版本。

```
[root@admin ~]# yum update ansible ceph-ansible
```

- 在 **rolling_update.yml** playbook 中，将 **health_osd_check_retries** 和 **health_osd_check_delay** 值分别改为 **50** 和 **30**。

```
health_osd_check_retries: 50
health_osd_check_delay: 30
```

设置这些值后，Ansible 将等待每个 OSD 节点最多等待 25 分钟，并且每隔 30 秒检查存储集群运行状况，等待继续升级过程。

注意

根据存储集群的已用存储容量，调整 **health_osd_check_retries** 选项的值。例如，如果您在 436 TB 中使用 218 TB，基本上使用 50% 的存储容量，然后将 **health_osd_check_retries** 选项设置为 **50**。

- 如果要升级的集群包含使用 **exclusive-lock** 功能的 Ceph 块设备镜像，请确保所有 Ceph 块设备用户都有将客户端列入黑名单的权限：

```
ceph auth caps client.<ID> mon 'allow r, allow command "osd blacklist"' osd '<existing-OSD-user-capabilities>'
```

4.1. 升级存储集群

流程

从 Ansible 管理节点使用以下命令：

1. 以 **root** 用户身份，导航到 **/usr/share/ceph-ansible/** 目录：

```
[root@admin ~]# cd /usr/share/ceph-ansible/
```

2. 从 Red Hat Ceph Storage 版本 3.x 升级到最新版本时跳过此步骤。备份 **group_vars/all.yml** 和 **group_vars/osds.yml** 文件。

```
[root@admin ceph-ansible]# cp group_vars/all.yml group_vars/all_old.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml group_vars/osds_old.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml group_vars/clients_old.yml
```

3. 从 Red Hat Ceph Storage 版本 3.x 升级到最新版本时跳过此步骤。从 Red Hat Ceph Storage 2.x 升级到 3.x 时，创建 **group_vars/all.yml.sample**、**group_vars/osds.yml.sample** 和 **group_vars/clients.yml.sample** 文件的新副本，并将它们分别重命名为 **group_vars/all.yml**、**group_vars/osds.yml** 和 **group_vars/clients.yml**。打开并相应地编辑它们。详情请查看 *附录 H, 版本 2 和版本 3 之间的 Ansible 变量更改* 和 第 3.2 节 “安装 Red Hat Ceph Storage 集群”。

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4. 从 Red Hat Ceph Storage 版本 3.x 升级到最新版本时跳过此步骤。当从 Red Hat Ceph Storage 2.x 升级到 3.x 时，打开 **group_vars/clients.yml** 文件并取消注释以下行：

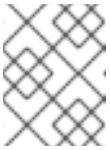
```
keys:
  - { name: client.test, caps: { mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" },  mode: "{{ ceph_keyring_permissions }}" }
```

a. 使用实际客户端名称替换 **client.test**，并在客户端定义行中添加客户端密钥，例如：

```
key: "ADD-KEYRING-HERE=="
```

现在，整行示例类似如下：

```
- { name: client.test, key: "AQAin8tUMICVFBAALRHNrV0Z4MXupRw4v9JQ6Q==", caps: { mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" }, mode: "{{ ceph_keyring_permissions }}" }
```



注意

若要获取客户端密钥，可运行 **ceph auth get-or-create** 命令，以查看指定客户端的密钥。

5. 在 **group_vars/all.yml** 文件中，取消注释 **upgrade_ceph_packages** 选项，并将它设为 **True**。

```
upgrade_ceph_packages: True
```

6. 在 **group_vars/all.yml** 文件中，将 **ceph_rhcs_version** 设置为 **3**。

```
ceph_rhcs_version: 3
```

注意

将 **ceph_rhcs_version** 选项设置为 **3** 将引入最新版本的 Red Hat Ceph Storage 3。

7. 在 **group_vars/all.yml** 文件中，将 **ceph_origin** 参数设置为 **distro** ：

```
ceph_origin: distro
```

8. 将 **fetch_directory** 参数添加到 **group_vars/all.yml** 文件。

```
fetch_directory: <full_directory_path>
```

替换：

- **<full_directory_path>**，位置一个可写的位置，如 Ansible 用户的主目录。提供用于初始存储集群安装的现有路径。

如果现有路径丢失或缺失，请首先执行以下操作：

a. 在现有 **group_vars/all.yml** 文件中添加以下选项：

```
fsid: <add_the_fsid>
generate_fsid: false
```

b. 运行 **take-over-existing-cluster.yml** Ansible playbook:

```
[user@admin ceph-ansible]$ cp infrastructure-playbooks/take-over-existing-cluster.yml .
[user@admin ceph-ansible]$ ansible-playbook take-over-existing-cluster.yml
```

9. 如果要升级的集群包含任何 Ceph 对象网关节点，请将 **radosgw_interface** 参数添加到 **group_vars/all.yml** 文件中。

```
radosgw_interface: <interface>
```

替换：

- **<interface>** 以及 Ceph 对象网关节点侦听的接口。

10. 从 Red Hat Ceph Storage 3.2 开始，默认的 OSD 对象存储是 BlueStore。若要保留传统的 OSD 对象存储，您必须将 **osd_objectstore** 选项明确设置为 **group_vars/all.yml** 文件中的 **filestore**。

```
osd_objectstore: filestore
```

注意

将 **osd_objectstore** 选项设置为 **filestore** 时，替换 OSD 将使用 FileStore，而不是 BlueStore。

11. 在位于 **/etc/ansible/hosts** 的 Ansible 清单文件中，将 Ceph Manager(**ceph-mgr**)节点添加到 **[mgrs]** 部分下。将 Ceph 管理器守护进程与 monitor 节点并置。从 3.x 升级到最新版本时跳过此步骤。

```
[mgrs]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

12. 将 **rolling_update.yml** 从 **infrastructure-playbooks** 目录中复制到当前目录中。

```
[root@admin ceph-ansible]# cp infrastructure-playbooks/rolling_update.yml .
```

重要

不要将 **limit** ansible 选项与 **rolling_update.yml** 搭配使用。

13. 创建 **/var/log/ansible/** 目录，并为 **ansible** 用户分配适当的权限：

```
[root@admin ceph-ansible]# mkdir /var/log/ansible
[root@admin ceph-ansible]# chown ansible:ansible  /var/log/ansible
[root@admin ceph-ansible]# chmod 755 /var/log/ansible
```

a. 编辑 **/usr/share/ceph-ansible/ansible.cfg** 文件，更新 **log_path** 值，如下所示：

```
log_path = /var/log/ansible/ansible.log
```

14. 以 Ansible 用户身份，运行 playbook:

```
[user@admin ceph-ansible]$ ansible-playbook rolling_update.yml
```

15. 在以 **root** 用户身份登录 RBD 镜像守护进程节点时，请手动升级 **rbd-mirror** ：

```
# yum upgrade rbd-mirror
```

重启守护进程：

```
# systemctl restart  ceph-rbd-mirror@<client-id>
```

16. 验证集群运行状况是否为 OK。以 **root** 用户身份登陆到监控节点，再运行 ceph status 命令。

```
[root@monitor ~]# ceph -s
```

1. 如果在 OpenStack 环境中工作，请更新所有 **cephx** 用户，以将 RBD 配置文件用于池。以下命令必须以 **root 用户身份运行** ：

- Glance 用户

```
ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd pool=<glance-pool-name>'
```

示例

```
[root@monitor ~]# ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd
pool=images'
```

- Cinder 用户

```
ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd pool=<cinder-volume-pool-
name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=<glance-pool-
name>'
```

示例

```
[root@monitor ~]# ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd
pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```

- OpenStack 常规用户

```
ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-only pool=<cinder-
volume-pool-name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=
<glance-pool-name>'
```

示例

```
[root@monitor ~]# ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-
only pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```

重要

在执行任何实时客户端迁移前，进行这些 CAPS 更新。这使得客户端能够使用内存中运行的新库，从而导致旧 CAPS 设置从缓存中丢弃并应用新的 RBD 配置集设置。

4.2. 升级 RED HAT CEPH STORAGE DASHBOARD

以下流程概述了将 Red Hat Ceph Storage Dashboard 从版本 3.1 升级到 3.2 的步骤。

在升级前，确保 Red Hat Ceph Storage 从版本 3.1 升级到 3.2。请参阅 4.1.升级存储集群。

警告

升级步骤会删除历史存储仪表板数据。

流程

1. 以 **root** 用户身份，从 Ansible 管理节点更新 **cephmetrics-ansible** 软件包：

```
[root@admin ~]# yum update cephmetrics-ansible
```

2. 进入 **/usr/share/cephmetrics-ansible** 目录：

```
[root@admin ~]# cd /usr/share/cephmetrics-ansible
```

3. 安装更新的 Red Hat Ceph Storage 仪表板：

```
[root@admin cephmetrics-ansible]# ansible-playbook -v playbook.yml
```

第 5 章 下一步做什么？

这仅仅是 Red Hat Ceph Storage 为帮助您满足现代数据中心富有挑战性的存储需求所它可以起到的作用的开始。以下是有关各种主题的更多信息的链接：

- 基准测试性能和访问性能计数器，请参见 Red Hat Ceph Storage 3 管理指南中的基准测试性能章节。
- 创建和管理快照，请参阅 Red Hat Ceph Storage 3 块设备指南中的 快照 章节。
- 扩展 Red Hat Ceph Storage 集群，请参阅 Red Hat Ceph Storage 3 管理指南中的管理集群大小章节。
- 镜像 Ceph 块设备，请参见 Red Hat Ceph Storage 3 块设备指南中的块设备镜像一章。
- 流程管理，请参见 Red Hat Ceph Storage 3 管理指南中的进程管理一章。
- 可调整参数，请参阅 Red Hat Ceph Storage 3 的 配置指南。
- 将 Ceph 用作 OpenStack 的后端存储，请参见 Red Hat OpenStack Platform 存储指南中的后端章节。

附录 A. 故障排除

A.1. ANSIBLE 停止安装，因为它检测了更少的设备超过预期

Ansible 自动化应用程序停止安装过程并返回以下错误：

```
- name: fix partitions gpt header or labels of the osd disks (autodiscover disks)
  shell: "sgdisk --zap-all --clear --mbrtogpt -- '/dev/{{ item.0.item.key }}' || sgdisk --zap-all --clear --
mbrtogpt -- '/dev/{{ item.0.item.key }}'"
  with_together:
    - "{{ osd_partition_status_results.results }}"
    - "{{ ansible_devices }}"
  changed_when: false
  when:
    - ansible_devices is defined
    - item.0.item.value.removable == "0"
    - item.0.item.value.partitions|count == 0
    - item.0.rc != 0
```

这意味着：

当 **/usr/share/ceph-ansible/group_vars/osds.yml** 文件中的 **osd_auto_discovery** 参数设置为 **true** 时，Ansible 会自动检测并配置所有可用的设备。在这一过程中，Ansible 期望所有 OSD 都使用相同的设备。设备按照 Ansible 检测到的名称的顺序获得它们的名称。如果其中一个设备在其中一个 OSD 上失败，Ansible 无法检测到失败的设备并停止整个安装过程。

示例情况：

1. 三个 OSD 节点（**host1**、**host2**、**host3**）使用 **/dev/sdb**、**/dev/sdc** 和 **dev/sdd** 磁盘。
2. 在 **host2** 上，**/dev/sdc** 磁盘失败并被删除。
3. 下一次重启后，Ansible 无法检测已移除的 **/dev/sdc** 磁盘，并且希望只有两个磁盘将用于 **host2**，即**/dev/sdb** 和 **/dev/sdc** （以前为 **/dev/sdd**）。
4. Ansible 将停止安装过程并返回上述错误消息。

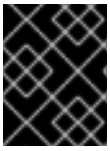
解决此问题：

在 **/etc/ansible/hosts** 文件中，指定带有故障磁盘的 OSD 节点使用的设备（上面的示例中为 **host2** ）：

```
[osds]
host1
host2 devices="[ '/dev/sdb', '/dev/sdc' ]"
host3
```

详情请查看 第 3 章 *部署 Red Hat Ceph Storage* 。

附录 B. 手动安装 RED HAT CEPH STORAGE



重要

红帽不支持或测试手动部署的集群的升级。因此，红帽建议使用 Ansible 来使用 Red Hat Ceph Storage 3 部署新集群。详情请查看 第 3 章 *部署 Red Hat Ceph Storage* 。

您可以使用命令行实用程序（如 Yum）来安装手动部署的集群。

所有 Ceph 集群需要至少一个 monitor，并且至少与集群中存储的对象副本数量相同。红帽建议在生产环境中使用三个监视器，至少三个对象存储设备 (OSD)。

使用命令行界面安装 Ceph 存储集群涉及以下步骤：

- 引导初始监控器节点.
- 安装 Ceph 管理器守护进程.
- 添加对象存储设备(OSD)节点.

B.1. 先决条件

为 Red Hat Ceph Storage 配置网络时间协议

所有 Ceph 监控器和 OSD 节点都需要配置网络时间协议(NTP)。确保 Ceph 节点是 NTP 对等节点。NTP 有助于抢占时钟偏移所带来的问题。

注意

在使用 Ansible 部署 Red Hat Ceph Storage 集群时，Ansible 会自动安装、配置和启用 NTP。

先决条件

- 网络访问有效时间源.

步骤：为 RHCS 配置网络时间协议

以 **root** 用户身份，在存储集群中的所有 RHCS 节点上执行以下步骤。

1. 安装 **ntp** 软件包：

```
# yum install ntp
```

2. 启动并确定 NTP 服务在重启后可以保持启动状态：

```
# systemctl start ntpd
# systemctl enable ntpd
```

3. 确保 NTP 正确同步时钟：

```
$ ntpq -p
```

其它资源

- Red Hat Enterprise Linux 7 系统管理员指南中的使用 ntpd 配置 NTP 章节

- Red Hat Enterprise Linux 7 系统管理员指南中的使用 ntpd 配置 NTP 章节。

监控 Bootstrapping

引导 monitor 和扩展 Ceph 存储集群需要以下数据：

唯一标识符

文件系统标识符(**fsid**)是集群的唯一标识符。**fsid** 最初在 Ceph 存储集群主要用于 Ceph 文件系统时使用。Ceph 现在也支持原生接口、块设备和对象存储网关接口，因此 **fsid** 可能会有一些问题。

集群名称

Ceph 集群具有集群名称，这是不含空格的简单字符串。默认集群名称为 **ceph**，但您可以指定不同的集群名称。当您使用多个集群时，覆盖默认集群名称特别有用。
当您在多站点架构中运行多个集群时，集群名称（如 **us-west,us-east**）标识当前命令行会话的集群。

注意

若要识别命令行界面上的集群名称，请使用集群名称指定 Ceph 配置文件，如 **ceph.conf**、**us-west.conf**、**us-east.conf** 等。

例如：

```
# ceph --cluster us-west.conf ...
```

Monitor 名称

集群中的每一个 Monitor 实例都有唯一的名称。在常见做法中，Ceph monitor 名称是节点名称。红帽建议每个节点一个 Ceph 监控器，而不与 Ceph 监控守护进程共同定位 Ceph OSD 守护进程。要获得较短的节点名称，请使用 **hostname -s** 命令。

Monitor Map

启动初始 Monitor 要求您生成 Monitor Map。Monitor map 需要：

- 文件系统识别符(**fsid**)
- 使用集群名称或 **ceph** 的默认集群名称
- 至少一个主机名及其 IP 地址。

监控密钥环

Monitor 使用 secret 密钥相互通信。您必须使用 Monitor secret 密钥生成密钥环，并在引导初始 Monitor 时提供密钥环。

管理员密钥环

要使用 **ceph** 命令行界面实用程序，请创建 **client.admin** 用户并生成其密钥环。此外，您必须将 **client.admin** 用户添加到 monitor 密钥环中。

强制要求不表示创建 Ceph 配置文件。但是，作为一种最佳实践，红帽建议创建一个 Ceph 配置文件并使用 **fsid** 填充它的数据，**mon initial members** 和 **mon host** 是最小设置。

您还可以在运行时获取和设置所有 Monitor 设置。但是，Ceph 配置文件可能仅包含覆盖默认值的设置。当您向 Ceph 配置文件添加设置时，这些设置将覆盖默认设置。在 Ceph 配置文件中维护这些设置可以更加轻松地维护集群。

要引导初始 Monitor，请执行以下步骤：

1. 启用 Red Hat Ceph Storage 3 monitor 存储库：

```
[root@monitor ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-mon-els-rpms
```

2. 在初始监控节点上，以 **root** 用户身份安装 **ceph-mon** 软件包：

```
# yum install ceph-mon
```

3. 以 **root** 用户身份，在 **/etc/ceph/** 目录中创建 Ceph 配置文件。默认情况下，Ceph 使用 **ceph.conf**，其中 **ceph** 反映了集群名称：

语法

```
# touch /etc/ceph/<cluster_name>.conf
```

示例

```
# touch /etc/ceph/ceph.conf
```

4. 以 **root** 用户身份，为集群生成唯一标识符，并将唯一标识符添加到 Ceph 配置文件的 **[global]** 部分：

语法

```
# echo "[global]" > /etc/ceph/<cluster_name>.conf
# echo "fsid = `uuidgen`" >> /etc/ceph/<cluster_name>.conf
```

示例

```
# echo "[global]" > /etc/ceph/ceph.conf
# echo "fsid = `uuidgen`" >> /etc/ceph/ceph.conf
```

5. 查看当前的 Ceph 配置文件：

```
$ cat /etc/ceph/ceph.conf
[global]
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
```

6. 以 **root** 用户身份，将初始 monitor 添加到 Ceph 配置文件：

语法

```
# echo "mon initial members = <monitor_host_name>[,<monitor_host_name>]" >> /etc/ceph/<cluster_name>.conf
```

示例

```
# echo "mon initial members = node1" >> /etc/ceph/ceph.conf
```

7. 以 **root** 用户身份，将初始 monitor 的 IP 地址添加到 Ceph 配置文件：

语法

```
# echo "mon host = <ip-address>[,<ip-address>]" >> /etc/ceph/<cluster_name>.conf
```

示例

```
# echo "mon host = 192.168.0.120" >> /etc/ceph/ceph.conf
```

注意

要使用 IPv6 地址，请将 **ms bind ipv6** 选项设置为 **true**。详情请参阅 Red Hat Ceph Storage 3 配置指南中的 Bind 部分。

8. 以 **root** 用户身份，为集群创建密钥环并生成 monitor secret 密钥：

语法

```
# ceph-authtool --create-keyring /tmp/<cluster_name>.mon.keyring --gen-key -n mon. --cap mon '<capabilites>'
```

示例

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

9. 以 **root** 用户身份生成管理员密钥环，生成 **<cluster_name>.client.admin.keyring** 用户，并将该用户添加到密钥环中：

语法

```
# ceph-authtool --create-keyring /etc/ceph/<cluster_name>.client.admin.keyring --gen-key -n client.admin --set-uid=0 --cap mon '<capabilites>' --cap osd '<capabilites>' --cap mds '<capabilites>'
```

示例

```
# ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n client.admin --set-uid=0 --cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow'
creating /etc/ceph/ceph.client.admin.keyring
```

10. 以 **root** 用户身份，将 **<cluster_name>.client.admin.keyring** 密钥添加到 **<cluster_name>.mon.keyring**:

语法

```
# ceph-authtool /tmp/<cluster_name>.mon.keyring --import-keyring /etc/ceph/<cluster_name>.client.admin.keyring
```

示例

```
# ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring
importing contents of /etc/ceph/ceph.client.admin.keyring into /tmp/ceph.mon.keyring
```

11. 生成 Monitor map。使用初始 monitor 的节点名称、IP 地址和 **fsid** 指定，并将其保存为 **/tmp/monmap**：

    语法

    ```
    $ monmaptool --create --add <monitor_host_name> <ip-address> --fsid <uuid>
    /tmp/monmap
    ```

    示例

    ```
    $ monmaptool --create --add node1 192.168.0.120 --fsid a7f64266-0894-4f1e-a635-
    d0aeaca0e993 /tmp/monmap
    monmaptool: monmap file /tmp/monmap
    monmaptool: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993
    monmaptool: writing epoch 0 to /tmp/monmap (1 monitors)
    ```

12. 作为初始监控节点上的 **root 用户**，创建一个默认数据目录：

    语法

    ```
    # mkdir /var/lib/ceph/mon/<cluster_name>-<monitor_host_name>
    ```

    示例

    ```
    # mkdir /var/lib/ceph/mon/ceph-node1
    ```

13. 以 **root** 用户身份，使用 monitor 映射和密钥环填充初始 monitor 守护进程：

    语法

    ```
    # ceph-mon [--cluster <cluster_name>] --mkfs -i <monitor_host_name> --monmap
    /tmp/monmap --keyring /tmp/<cluster_name>.mon.keyring
    ```

    示例

    ```
    # ceph-mon --mkfs -i node1 --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring
    ceph-mon: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993
    ceph-mon: created monfs at /var/lib/ceph/mon/ceph-node1 for mon.node1
    ```

14. 查看当前的 Ceph 配置文件：

    ```
    # cat /etc/ceph/ceph.conf
    [global]
    fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
    mon_initial_members = node1
    mon_host = 192.168.0.120
    ```

    有关各种 Ceph 配置设置的更多详细信息，请参见 Red Hat Ceph Storage 3 配置指南 。以下 Ceph 配置文件示例列出了一些最常见的配置设置：

    示例

```
[global]
fsid = <cluster-id>
mon initial members = <monitor_host_name>[, <monitor_host_name>]
mon host = <ip-address>[, <ip-address>]
public network = <network>[, <network>]
cluster network = <network>[, <network>]
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
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>
osd crush chooseleaf type = <n>
```

15. 以 **root** 用户身份，创建 **done** 文件：

语法

```
# touch /var/lib/ceph/mon/<cluster_name>-<monitor_host_name>/done
```

示例

```
# touch /var/lib/ceph/mon/ceph-node1/done
```

16. 以 **root** 用户身份，更新新创建的目录和文件的所有者和组权限：

语法

```
# chown -R <owner>:<group> <path_to_directory>
```

示例

```
# chown -R ceph:ceph /var/lib/ceph/mon
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown ceph:ceph /etc/ceph/ceph.client.admin.keyring
# chown ceph:ceph /etc/ceph/ceph.conf
# chown ceph:ceph /etc/ceph/rbdmap
```

注意

如果 Ceph 监控节点与 OpenStack 控制器节点在一起，则 Glance 和 Cinder 密钥环文件必须分别归 **glance** 和 **cinder** 所有。例如：

```
# ls -l /etc/ceph/
...
-rw-------.  1 glance glance      64 <date> ceph.client.glance.keyring
-rw-------.  1 cinder cinder      64 <date> ceph.client.cinder.keyring
...
```

17. 对于带有自定义名称的存储集群，以 **root** 用户身份添加以下行：

语法

```
# echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
```

示例

```
# echo "CLUSTER=test123" >> /etc/sysconfig/ceph
```

18. 以 **root** 用户身份，在初始监控节点上启动并启用 **ceph-mon** 进程：

语法

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@<monitor_host_name>
# systemctl start ceph-mon@<monitor_host_name>
```

示例

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@node1
# systemctl start ceph-mon@node1
```

19. 以 **root** 用户身份，验证 monitor 守护进程是否正在运行：

语法

```
# systemctl status ceph-mon@<monitor_host_name>
```

示例

```
# systemctl status ceph-mon@node1
● ceph-mon@node1.service - Ceph cluster monitor daemon
   Loaded: loaded (/usr/lib/systemd/system/ceph-mon@.service; enabled; vendor preset:
disabled)
   Active: active (running) since Wed 2018-06-27 11:31:30 PDT; 5min ago
 Main PID: 1017 (ceph-mon)
   CGroup: /system.slice/system-ceph\x2dmon.slice/ceph-mon@node1.service
           └─1017 /usr/bin/ceph-mon -f --cluster ceph --id node1 --setuser ceph --setgroup ceph

Jun 27 11:31:30 node1 systemd[1]: Started Ceph cluster monitor daemon.
Jun 27 11:31:30 node1 systemd[1]: Starting Ceph cluster monitor daemon...
```

要将更多 Red Hat Ceph Storage Monitor 添加到存储集群中，请参阅 Red Hat Ceph Storage 3 管理指南中的添加 Monitor 部分。

B.2. 手动安装 CEPH MANAGER

通常，在部署 Red Hat Ceph Storage 集群时，Ansible 自动化实用程序会安装 Ceph Manager 守护进程 (**ceph-mgr**)。但是，如果您不使用 Ansible 管理红帽 Ceph 存储，您可以手动安装 Ceph Manager。红帽建议在同一节点上并置 Ceph 管理器和 Ceph 监控守护进程。

先决条件

- 正常工作的 Red Hat Ceph Storage 集群
- **root** 或 **sudo** 访问权限
- **rhel-7-server-rhceph-3-mon-els-rpms** 存储库已启用
- 如果使用防火墙，需要在公共网络上打开端口 **6800-7300**

流程

在要部署 **ceph-mgr** 的节点上，以 **root** 用户身份或通过 **sudo** 实用程序，使用以下命令。

1. 安装 **ceph-mgr** 软件包：

```
[root@node1 ~]# yum install ceph-mgr
```

2. 创建 **/var/lib/ceph/mgr/ceph-*hostname*/** 目录：

```
mkdir /var/lib/ceph/mgr/ceph-hostname
```

使用部署 **ceph-mgr** 守护进程的节点的主机名替换 *hostname*，例如：

```
[root@node1 ~]# mkdir /var/lib/ceph/mgr/ceph-node1
```

3. 在新创建的目录中，为 **ceph-mgr** 守护进程创建一个身份验证密钥：

```
[root@node1 ~]# ceph auth get-or-create mgr.`hostname -s` mon 'allow profile mgr' osd
'allow *' mds 'allow *' -o /var/lib/ceph/mgr/ceph-node1/keyring
```

4. 将 **/var/lib/ceph/mgr/** 目录的所有者和组更改为 **ceph:ceph** ：

```
[root@node1 ~]# chown -R ceph:ceph /var/lib/ceph/mgr
```

5. 启用 **ceph-mgr** 目标：

```
[root@node1 ~]# systemctl enable ceph-mgr.target
```

6. 启用并启动 **ceph-mgr** 实例：

```
systemctl enable ceph-mgr@hostname
systemctl start ceph-mgr@hostname
```

使用部署 **ceph-mgr** 的节点的主机名替换 *hostname*，例如：

```
[root@node1 ~]# systemctl enable ceph-mgr@node1
[root@node1 ~]# systemctl start ceph-mgr@node1
```

7. 验证 **ceph-mgr** 守护进程是否已成功启动：

```
ceph -s
```

输出将在 **services** 部分下包括类似如下的行：

```
    mgr: node1(active)
```

8. 安装更多 **ceph-mgr** 守护进程以作为备用守护进程（如果当前活跃守护进程失败）处于活跃状态。

其他资源

- *安装 Red Hat Ceph Storage 的要求*

OSD Bootstrapping

运行初始监控器后，您可以开始添加对象存储设备 (OSD)。直到有足够的 OSD 来处理对象的副本数时，您的集群才会达到 **active + clean** 状态。

对象的默认副本数为三个。至少需要三个 OSD 节点：但是，如果您只需要一个对象的两个副本，因此仅添加两个 OSD 节点，然后更新 Ceph 配置文件中的 **osd pool default size** 和 **osd pool default min size** 设置。

如需了解更多详细信息，请参阅 Red Hat Ceph Storage 3 *配置指南*中的*OSD 配置参考*一节。

在引导初始监控器后，集群具有默认的 CRUSH map。但是，CRUSH map 没有任何 Ceph OSD 守护进程映射到 Ceph 节点。

要添加 OSD 到集群并更新默认的 CRUSH map，请在每个 OSD 节点上执行以下内容：

1. 启用 Red Hat Ceph Storage 3 OSD 存储库：

```
[root@osd ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-osd-els-rpms
```

2. 以 **root** 用户身份，在 Ceph OSD 节点上安装 **ceph-osd** 软件包：

```
# yum install ceph-osd
```

3. 将 Ceph 配置文件和管理密钥环文件从初始 Monitor 节点复制到 OSD 节点：

   语法

```
# scp <user_name>@<monitor_host_name>:<path_on_remote_system>
<path_to_local_file>
```

   示例

```
# scp root@node1:/etc/ceph/ceph.conf /etc/ceph
# scp root@node1:/etc/ceph/ceph.client.admin.keyring /etc/ceph
```

4. 为 OSD 生成通用唯一标识符 (UUID)：

```
$ uuidgen
b367c360-b364-4b1d-8fc6-09408a9cda7a
```

5. 以 **root** 用户身份，创建 OSD 实例：

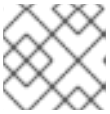
   语法

语法

```
# ceph osd create <uuid> [<osd_id>]
```

示例

```
# ceph osd create b367c360-b364-4b1d-8fc6-09408a9cda7a
0
```



注意

此命令输出后续步骤所需的 OSD 编号标识符。

6. 以 **root** 用户身份，为新 OSD 创建默认目录：

   语法

   ```
   # mkdir /var/lib/ceph/osd/<cluster_name>-<osd_id>
   ```

   示例

   ```
   # mkdir /var/lib/ceph/osd/ceph-0
   ```

7. 以 **root** 用户身份，准备好将驱动器用作 OSD，并将它挂载到您刚才创建的目录中。为 Ceph 数据和日志创建一个分区。日志和数据分区可以位于同一磁盘上。这个示例使用 15 GB 磁盘：

   语法

   ```
   # parted <path_to_disk> mklabel gpt
   # parted <path_to_disk> mkpart primary 1 10000
   # mkfs -t <fstype> <path_to_partition>
   # mount -o noatime <path_to_partition> /var/lib/ceph/osd/<cluster_name>-<osd_id>
   # echo "<path_to_partition>  /var/lib/ceph/osd/<cluster_name>-<osd_id>   xfs
   defaults,noatime 1 2" >> /etc/fstab
   ```

   示例

   ```
   # parted /dev/sdb mklabel gpt
   # parted /dev/sdb mkpart primary 1 10000
   # parted /dev/sdb mkpart primary 10001 15000
   # mkfs -t xfs /dev/sdb1
   # mount -o noatime /dev/sdb1 /var/lib/ceph/osd/ceph-0
   # echo "/dev/sdb1 /var/lib/ceph/osd/ceph-0  xfs defaults,noatime 1 2" >> /etc/fstab
   ```

8. 以 **root** 用户身份，初始化 OSD 数据目录：

   语法

   ```
   # ceph-osd -i <osd_id> --mkfs --mkkey --osd-uuid <uuid>
   ```

   示例

```
# ceph-osd -i 0 --mkfs --mkkey --osd-uuid b367c360-b364-4b1d-8fc6-09408a9cda7a
... auth: error reading file: /var/lib/ceph/osd/ceph-0/keyring: can't open /var/lib/ceph/osd/ceph-0/keyring: (2) No such file or directory
... created new key in keyring /var/lib/ceph/osd/ceph-0/keyring
```

注意

在使用 **--mkkey** 选项运行 **ceph-osd** 之前，目录必须为空。如果您有自定义集群名称，**ceph-osd** 实用程序需要 **--cluster** 选项。

9. 以 **root** 身份，注册 OSD 身份验证密钥。如果集群名称与 **ceph** 不同，请插入集群名称：

   语法

   ```
   # ceph auth add osd.<osd_id> osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/<cluster_name>-<osd_id>/keyring
   ```

   示例

   ```
   # ceph auth add osd.0 osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-0/keyring
   added key for osd.0
   ```

10. 以 **root** 用户身份，将 OSD 节点添加到 CRUSH map：

    语法

    ```
    # ceph [--cluster <cluster_name>] osd crush add-bucket <host_name> host
    ```

    示例

    ```
    # ceph osd crush add-bucket node2 host
    ```

11. 以 **root** 用户身份，将 OSD 节点放在 **default** CRUSH 树下：

    语法

    ```
    # ceph [--cluster <cluster_name>] osd crush move <host_name> root=default
    ```

    示例

    ```
    # ceph osd crush move node2 root=default
    ```

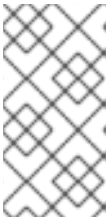
12. 以 **root** 用户身份，将 OSD 磁盘添加到 CRUSH map

    语法

    ```
    # ceph [--cluster <cluster_name>] osd crush add osd.<osd_id> <weight> [<bucket_type>=<bucket-name> ...]
    ```

    示例

```
# ceph osd crush add osd.0 1.0 host=node2
add item id 0 name 'osd.0' weight 1 at location {host=node2} to crush map
```



注意

您也可以解译 CRUSH map，并将 OSD 添加到设备列表中。将 OSD 节点添加为 bucket，然后将设备添加为 OSD 节点中的项目，为 OSD 分配一个权重，重新编译 CRUSH map，并且设置 CRUSH map。如需了解更多详细信息，请参阅 Red Hat Ceph Storage 3 的*存储策略指南*中的编辑 CRUSH map 部分。

13. 以 **root** 用户身份，更新新创建的目录和文件的所有者和组权限：

语法

```
# chown -R <owner>:<group> <path_to_directory>
```

示例

```
# chown -R ceph:ceph /var/lib/ceph/osd
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```

14. 对于带有自定义名称的存储集群，以 **root** 用户身份在 **/etc/sysconfig/ceph** 文件中添加以下行：

语法

```
# echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
```

示例

```
# echo "CLUSTER=test123" >> /etc/sysconfig/ceph
```

15. OSD 节点位于 Ceph 存储集群配置中。不过，OSD 守护进程为 **down** 和 **in**。新 OSD 的状态必须为 **up** 后才能开始接收数据。以 **root** 用户身份，启用并启动 OSD 过程：

语法

```
# systemctl enable ceph-osd.target
# systemctl enable ceph-osd@<osd_id>
# systemctl start ceph-osd@<osd_id>
```

示例

```
# systemctl enable ceph-osd.target
# systemctl enable ceph-osd@0
# systemctl start ceph-osd@0
```

启动 OSD 守护进程后，它就为 **up** 和 **in**。

现在，您已启动并运行 Monitor 和一些 OSD。您可以执行以下命令来观察放置组对等点：

```
$ ceph -w
```

要查看 OSD 树，请执行以下命令：

```
$ ceph osd tree
```

示例

```
ID  WEIGHT   TYPE NAME       UP/DOWN  REWEIGHT  PRIMARY-AFFINITY
-1       2   root default
-2       2       host node2
 0       1           osd.0        up       1              1
-3       1       host node3
 1       1           osd.1        up       1              1
```

若要通过添加新 OSD 到存储集群来扩展存储容量，请参阅 Red Hat Ceph Storage 3 *管理指南*中的添加 OSD 部分。

附录 C. 安装 CEPH 命令行界面

Ceph 命令行界面 (CLI) 让管理员可以执行 Ceph 管理命令。CLI 由 **ceph-common** 软件包提供，包括以下实用程序：

- **ceph**
- **ceph-authtool**
- **ceph-dencoder**
- **rados**

先决条件

- 正在运行的 Ceph 存储集群，最好处于 **active + clean** 状态。

流程

1. 在客户端节点上，启用 Red Hat Ceph Storage 3 Tools 存储库：

```
[root@gateway ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

2. 在客户端节点上安装 **ceph-common** 软件包：

```
# yum install ceph-common
```

3. 从初始监控节点，复制 Ceph 配置文件，本例中为 **ceph.conf**，以及管理密钥环到客户端节点：

   语法

```
# scp /etc/ceph/<cluster_name>.conf <user_name>@<client_host_name>:/etc/ceph/
# scp /etc/ceph/<cluster_name>.client.admin.keyring
<user_name>@<client_host_name:/etc/ceph/
```

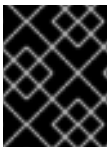
   示例

```
# scp /etc/ceph/ceph.conf root@node1:/etc/ceph/
# scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

   将 **<client_host_name>** 替换为客户端节点的主机名。

附录 D. 手动安装 CEPH 块设备

以下步骤演示了如何安装和挂载精简调配、可调整的 Ceph 块设备。



重要

Ceph 块设备必须部署到与 Ceph 监控器和 OSD 节点上独立的节点上。在同一节点上运行内核客户端和内核服务器守护进程可能会导致内核死锁。

先决条件

- 确保执行 附录 C, *安装 Ceph 命令行界面* 部分中列出的任务。
- 如果您使用 Ceph 块设备作为使用 QEMU 的虚拟机 (VM) 的后端，请增加默认的文件描述符。详情请参阅 Ceph - 虚拟机在将大量数据传输到 RBD 磁盘挂起 知识库文章。

流程

1. 创建名为 **client.rbd** 的 Ceph 块设备用户，该用户对 OSD 节点上的文件具有完整权限 (**osd 'allow rwx'**) 并将结果输出到密钥环文件：

```
ceph auth get-or-create client.rbd mon 'profile rbd' osd 'profile rbd pool=<pool_name>' \
-o /etc/ceph/rbd.keyring
```

将 **<pool_name>** 替换为您要允许 **client.rbd** 访问的池的名称，如 **rbd** ：

```
# ceph auth get-or-create \
client.rbd mon 'allow r' osd 'allow rwx pool=rbd' \
-o /etc/ceph/rbd.keyring
```

有关创建用户的更多信息，请参见 Red Hat Ceph Storage 3 *管理指南*中的*用户管理*一节。

2. 创建块设备镜像：

```
rbd create <image_name> --size <image_size> --pool <pool_name> \
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

指定 **<image_name>**、**<image_size>** 和 **<pool_name>**，例如：

```
$ rbd create image1 --size 4096 --pool rbd \
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

警告

默认 Ceph 配置包括以下 Ceph 块设备功能：

- **layering**
- **exclusive-lock**
- **object-map**
- **deep-flatten**
- **fast-diff**

如果使用内核 RBD(**krbd**)客户端，您将无法映射块设备镜像，因为 Red Hat Enterprise Linux 7.3 中包含的当前内核版本不支持 **object-map**、**bigh-flatten** 和 **fast-diff**。

要临时解决这个问题，请禁用不支持的功能。使用以下选项之一完成此操作：

- 动态禁用不支持的功能：

```
rbd feature disable <image_name> <feature_name>
```

例如：

```
# rbd feature disable image1 object-map deep-flatten fast-diff
```

- 在 **rbd create** 命令中使用 **--image-feature layering** 选项在新创建的块设备镜像仅启用 **layering**。
- 在 Ceph 配置文件中禁用默认功能：

```
rbd_default_features = 1
```

这是一个已知问题，请参阅 Red Hat Ceph Storage 3 *发行说明中*的*已知问题*章节。

所有这些功能适用于使用用户空间 RBD 客户端访问块设备镜像的用户。

3. 将新创建的镜像映射到块设备：

```
rbd map <image_name> --pool <pool_name>\
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

例如：

```
# rbd map image1 --pool rbd --name client.rbd \
--keyring /etc/ceph/rbd.keyring
```

重要

内核块设备目前仅支持 CRUSH 映射中的传统的 straw bucket 算法。如果将 CRUSH 可调项设置为最佳效果，您必须将它们设置为旧或较早的主版本，否则您将无法映射镜像。

或者，将 **straw2** 替换为 CRUSH 映射中的 **straw**。详情请参阅 Red Hat Ceph Storage 3 *存储策略*指南中的*编辑 CRUSH map* 章节。

4. 通过创建文件系统来使用块设备：

```
mkfs.ext4 -m5 /dev/rbd/<pool_name>/<image_name>
```

指定池名称和镜像名称，例如：

```
# mkfs.ext4 -m5 /dev/rbd/rbd/image1
```

这可能需要一些时间。

5. 挂载新创建的文件系统：

```
mkdir <mount_directory>
mount /dev/rbd/<pool_name>/<image_name> <mount_directory>
```

例如：

```
# mkdir /mnt/ceph-block-device
# mount /dev/rbd/rbd/image1 /mnt/ceph-block-device
```

如需了解更多详细信息，请参阅 Red Hat Ceph Storage 3 的块设备指南。

附录 E. 手动安装 CEPH 对象网关

Ceph 对象网关（也称为 RADOS 网关）是在 **librados** API 基础上构建的对象存储接口，为应用提供 Ceph 存储集群的 RESTful 网关。

先决条件

- 正在运行的 Ceph 存储集群，最好处于 **active + clean** 状态。
- 执行 *第 2 章 安装 Red Hat Ceph Storage 的要求* 中列出的任务。

流程

1. 启用 Red Hat Ceph Storage 3 Tools 存储库：

```
[root@gateway ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

2. 在 Object Gateway 节点上安装 **ceph-radosgw** 软件包：

```
# yum install ceph-radosgw
```

3. 在初始 monitor 节点上，执行以下步骤：

   a. 更新 Ceph 配置文件，如下所示：

```
[client.rgw.<obj_gw_hostname>]
host = <obj_gw_hostname>
rgw frontends = "civetweb port=80"
rgw dns name = <obj_gw_hostname>.example.com
```

   其中 **<obj_gw_hostname>** 是网关节点的短主机名。要查看短主机名，请使用 **hostname -s** 命令。

   b. 将更新的配置文件复制到新的对象网关节点和 Ceph 存储集群中的所有其他节点：

   语法

```
# scp /etc/ceph/<cluster_name>.conf <user_name>@<target_host_name>:/etc/ceph
```

   示例

```
# scp /etc/ceph/ceph.conf root@node1:/etc/ceph/
```

   c. 将 **<cluster_name>.client.admin.keyring** 文件复制到新的对象网关节点：

   语法

```
# scp /etc/ceph/<cluster_name>.client.admin.keyring
<user_name>@<target_host_name>:/etc/ceph/
```

   示例

```
# scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

4. 在对象网关节点上，创建数据目录：

语法

```
# mkdir -p /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`
```

示例

```
# mkdir -p /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`
```

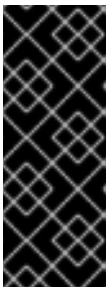
5. 在对象网关节点上，添加一个用户和密钥环来 bootstrap 对象网关：

语法

```
# ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o
/var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`/keyring
```

示例

```
# ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o
/var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/keyring
```



重要

为网关密钥提供功能时，您必须提供读取功能。但是，提供 monitor 写入功能是可选的；如果您提供此功能，Ceph 对象网关将能够自动创建池。

在这种情况下，请确保在池中指定合理的 PG 数量。否则，网关使用默认编号，该编号可能不适合您的需要。有关详细信息，请参阅每个池计算器的 Ceph Placement Group (PG)。

6. 在对象网关节点上，创建 **done** 文件：

语法

```
# touch /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`/done
```

示例

```
# touch /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/done
```

7. 在对象网关节点上，更改所有者和组权限：

```
# chown -R ceph:ceph /var/lib/ceph/radosgw
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```

8. 对于带有自定义名称的存储集群，以 **root** 用户身份添加以下行：

语法

```
# echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
```

示例

```
# echo "CLUSTER=test123" >> /etc/sysconfig/ceph
```

9. 在 Object Gateway 节点上打开 TCP 端口 80：

```
# firewall-cmd --zone=public --add-port=80/tcp
# firewall-cmd --zone=public --add-port=80/tcp --permanent
```

10. 在对象网关节点上，启动并启用 **ceph-radosgw** 进程：

语法

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.<rgw_hostname>
# systemctl start ceph-radosgw@rgw.<rgw_hostname>
```

示例

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.node1
# systemctl start ceph-radosgw@rgw.node1
```

安装后，如果在 monitor 上设置了写入功能，Ceph 对象网关会自动创建池。有关手动创建 池 的信息，请参阅存储策略指南中的池章节。

额外详情

- Red Hat Ceph Storage 3 Red Hat Enterprise Linux 对象网关指南

附录 F. 覆盖 CEPH 默认设置

除非在 Ansible 配置文件中另有指定，否则 Ceph 将使用其默认设置。

由于 Ansible 管理 Ceph 配置文件，请编辑 **/usr/share/ceph-ansible/group_vars/all.yml** 文件，以更改 Ceph 配置。使用 **ceph_conf_overrides** 设置覆盖默认的 Ceph 配置。

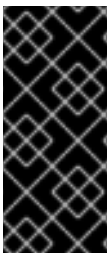
Ansible 支持与 Ceph 配置文件相同的部分；**[global]**、**[mon]**、**[osd]**、**[mds]**、**[rgw]** 等。您还可以覆盖特定的实例，如特定的 Ceph 对象网关实例。例如：

```
##################
# CONFIG OVERRIDE #
##################

ceph_conf_overrides:
  client.rgw.rgw1:
    log_file: /var/log/ceph/ceph-rgw-rgw1.log
```

注意

当引用 Ceph 配置文件的特定部分时，Ansible 不包含大括号。部分和设置名称以冒号结尾。



重要

不要使用 CONFIG OVERRIDE 部分中的 **cluster_network** 参数设置集群网络，因为这可能导致 Ceph 配置文件中设置两个相互冲突的集群网络。

要设置集群网络，请使用 CEPH CONFIGURATION 部分中的 **cluster_network** 参数。详情请查看 第 3.2 节 “安装 Red Hat Ceph Storage 集群” 。

附录 G. 手动从 RED HAT CEPH STORAGE 2 升级到 3

您可以滚动方式将 Ceph 存储集群从版本 2 升级到 3，并在集群运行时升级到 3。按顺序升级集群中的每个节点，仅在完成上一个节点后继续下一个节点。

红帽建议按照以下顺序升级 Ceph 组件：

- 监控节点
- OSD 节点
- Ceph 对象网关节点
- 所有其他 Ceph 客户端节点

Red Hat Ceph Storage 3 引入了一个新的守护进程 Ceph 管理器(**ceph-mgr**)。在升级 monitor 节点后安装 **ceph-mgr**。

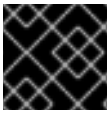
有两种方法可用来将 Red Hat Ceph Storage 2 升级到 3：

- 使用红帽的内容交付网络(CDN)
- 使用红帽提供的 ISO 镜像文件

在升级存储集群后，您可以使用传统的可调项，显示 CRUSH map 的运行状况警告。详情请参阅 Red Hat Ceph Storage 3 的存储策略指南中的 CRUSH Tunables 部分。

示例

```
$ ceph -s
    cluster 848135d7-cdb9-4084-8df2-fb5e41ae60bd
     health HEALTH_WARN
            crush map has legacy tunables (require bobtail, min is firefly)
     monmap e1: 1 mons at {ceph1=192.168.0.121:6789/0}
            election epoch 2, quorum 0 ceph1
     osdmap e83: 2 osds: 2 up, 2 in
      pgmap v1864: 64 pgs, 1 pools, 38192 kB data, 17 objects
            10376 MB used, 10083 MB / 20460 MB avail
                  64 active+clean
```



重要

红帽建议所有 Ceph 客户端运行与 Ceph 存储集群相同的版本。

先决条件

- 如果要升级的集群包含使用 **exclusive-lock** 功能的 Ceph 块设备镜像，请确保所有 Ceph 块设备用户都有将客户端列入黑名单的权限：

```
ceph auth caps client.<ID> mon 'allow r, allow command "osd blacklist"' osd '<existing-OSD-user-capabilities>'
```

升级监控节点

本节介绍将 Ceph 监控节点升级到更新版本的步骤。monitor 的数量必须是奇数。当您升级一个 monitor 时，存储群集仍会拥有仲裁。

流程

在存储集群中的每个 monitor 节点上执行以下步骤。一次仅升级一个 monitor 节点。

1. 如果使用软件存储库安装 Red Hat Ceph Storage 2，请禁用软件仓库：

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2-mon-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

2. 启用 Red Hat Ceph Storage 3 monitor 存储库：

```
[root@monitor ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-mon-els-rpms
```

3. 以 **root** 用户身份，停止 monitor 进程：

   语法

```
# service ceph stop <daemon_type>.<monitor_host_name>
```

   示例

```
# service ceph stop mon.node1
```

4. 以 **root** 用户身份，更新 **ceph-mon** 软件包：

```
# yum update ceph-mon
```

5. 以 **root** 用户身份，更新所有者和组权限：

   语法

```
# chown -R <owner>:<group> <path_to_directory>
```

   示例

```
# chown -R ceph:ceph /var/lib/ceph/mon
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown ceph:ceph /etc/ceph/ceph.client.admin.keyring
# chown ceph:ceph /etc/ceph/ceph.conf
# chown ceph:ceph /etc/ceph/rbdmap
```

注意

如果 Ceph 监控节点与 OpenStack 控制器节点在一起，则 Glance 和 Cinder 密钥环文件必须分别归 **glance** 和 **cinder** 所有。例如：

```
# ls -l /etc/ceph/
...
-rw-------.  1 glance glance      64 <date> ceph.client.glance.keyring
-rw-------.  1 cinder cinder      64 <date> ceph.client.cinder.keyring
...
```

6. 如果 SELinux 处于 enforcing 或 permissive 模式，请在下次重启时重新标记 SELinux 上下文。

```
# touch /.autorelabel
```

警告

重新标记可能需要很长时间才能完成，因为 SELinux 必须遍历每个文件系统并修复任何错误标记的文件。要排除要重新标记的目录，请在重启前将目录添加到 **/etc/selinux/fixfiles_exclude_dirs** 文件。

7. 以 **root** 用户身份，启用 **ceph-mon** 进程：

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@<monitor_host_name>
```

8. 以 **root** 用户身份，重启 monitor 节点：

```
# shutdown -r now
```

9. 监控节点启动后，在移动到下一个 monitor 节点前检查 Ceph 存储集群的运行状况：

```
# ceph -s
```

G.1. 手动安装 CEPH MANAGER

通常，在部署 Red Hat Ceph Storage 集群时，Ansible 自动化实用程序会安装 Ceph Manager 守护进程 (**ceph-mgr**)。但是，如果您不使用 Ansible 管理红帽 Ceph 存储，您可以手动安装 Ceph Manager。红帽建议在同一节点上并置 Ceph 管理器和 Ceph 监控守护进程。

先决条件

- 正常工作的 Red Hat Ceph Storage 集群
- **root** 或 **sudo** 访问权限
- **rhel-7-server-rhceph-3-mon-els-rpms** 存储库已启用

- 如果使用防火墙，需要在公共网络上打开端口 **6800-7300**

流程

在要部署 **ceph-mgr** 的节点上，以 **root 用户身份或通过 sudo** 实用程序，使用以下命令。

1. 安装 **ceph-mgr** 软件包：

```
[root@node1 ~]# yum install ceph-mgr
```

2. 创建 **/var/lib/ceph/mgr/ceph-*hostname*/** 目录：

```
mkdir /var/lib/ceph/mgr/ceph-hostname
```

使用部署 **ceph-mgr** 守护进程的节点的主机名替换 *hostname*，例如：

```
[root@node1 ~]# mkdir /var/lib/ceph/mgr/ceph-node1
```

3. 在新创建的目录中，为 **ceph-mgr** 守护进程创建一个身份验证密钥：

```
[root@node1 ~]# ceph auth get-or-create mgr.`hostname -s` mon 'allow profile mgr' osd
'allow *' mds 'allow *' -o /var/lib/ceph/mgr/ceph-node1/keyring
```

4. 将 **/var/lib/ceph/mgr/** 目录的所有者和组更改为 **ceph:ceph** ：

```
[root@node1 ~]# chown -R ceph:ceph /var/lib/ceph/mgr
```

5. 启用 **ceph-mgr** 目标：

```
[root@node1 ~]# systemctl enable ceph-mgr.target
```

6. 启用并启动 **ceph-mgr** 实例：

```
systemctl enable ceph-mgr@hostname
systemctl start ceph-mgr@hostname
```

使用部署 **ceph-mgr** 的节点的主机名替换 *hostname*，例如：

```
[root@node1 ~]# systemctl enable ceph-mgr@node1
[root@node1 ~]# systemctl start ceph-mgr@node1
```

7. 验证 **ceph-mgr** 守护进程是否已成功启动：

```
ceph -s
```

输出将在 **services** 部分下包括类似如下的行：

```
    mgr: node1(active)
```

8. 安装更多 **ceph-mgr** 守护进程以作为备用守护进程（如果当前活跃守护进程失败）处于活跃状态。

其他资源

- *安装 Red Hat Ceph Storage 的要求*

升级 OSD 节点

本节介绍将 Ceph OSD 节点升级到更新版本的步骤。

先决条件

在升级 OSD 节点时，一些 PG 可能会降级，因为 OSD 可能会停机或重新启动。要防止 Ceph 启动恢复过程，请在 monitor 节点上设置 **noout** 和 **norebalance** OSD 标志：

```
[root@monitor ~]# ceph osd set noout
[root@monitor ~]# ceph osd set norebalance
```

流程

对存储集群中的每个 OSD 节点上执行下列步骤。一次仅升级一个 OSD 节点。如果为 Red Hat Ceph Storage 2.3 执行基于 ISO 的安装，则跳过此第一步。

1. 以 **root** 用户身份，禁用 Red Hat Ceph Storage 2 存储库：

   ```
   # subscription-manager repos --disable=rhel-7-server-rhceph-2-osd-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
   ```

2. 启用 Red Hat Ceph Storage 3 OSD 存储库：

   ```
   [root@osd ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-osd-els-rpms
   ```

3. 以 **root** 用户身份，停止任何正在运行的 OSD 进程：

   语法

   ```
   # service ceph stop <daemon_type>.<osd_id>
   ```

   示例

   ```
   # service ceph stop osd.0
   ```

4. 以 **root** 用户身份，更新 **ceph-osd** 软件包：

   ```
   # yum update ceph-osd
   ```

5. 以 **root** 用户身份，更新新创建的目录和文件的所有者和组权限：

   语法

   ```
   # chown -R <owner>:<group> <path_to_directory>
   ```

   示例

   ```
   # chown -R ceph:ceph /var/lib/ceph/osd
   # chown -R ceph:ceph /var/log/ceph
   # chown -R ceph:ceph /var/run/ceph
   ```

```
# chown -R ceph:ceph /etc/ceph
```

注意

在一个带有大量磁盘的 Ceph 存储集群中，可以使用以下 **find** 命令，通过并行执行 **chown** 命令来加快修改所有者设置的过程：

```
# find /var/lib/ceph/osd -maxdepth 1 -mindepth 1 -print | xargs -P12 -n1 chown
-R ceph:ceph
```

6. 如果 SELinux 被设置为 enforcing 或 permissive 模式，则在文件中设置 SELinux 上下文的重新标记，以便在下次重启：

```
# touch /.autorelabel
```

警告

重新标记可能需要很长时间才能完成，因为 SELinux 必须遍历每个文件系统并修复任何错误标记的文件。要排除要重新标记的目录，请在重启前将目录添加到 **/etc/selinux/fixfiles_exclude_dirs** 文件。

注意

在每个放置组 (PG) 具有大量对象的环境中，使用目录枚举速度会降低，从而导致对性能造成负面影响。这是因为添加 xattr 查询来验证 SELinux 上下文。在挂载时设置上下文会删除对上下文的 xattr 查询，这可以提高磁盘的整体性能，特别对于较慢的磁盘。

将以下行添加到 **/etc/ceph/ceph.conf** 文件中的 **[osd]** 部分：

+

```
osd_mount_options_xfs=rw,noatime,inode64,context="system_u:object_r:ceph_
var_lib_t:s0"
```

7. 以 **root** 用户身份，重播内核的设备事件：

```
# udevadm trigger
```

8. 以 **root** 用户身份，启用 **ceph-osd** 进程：

```
# systemctl enable ceph-osd.target
# systemctl enable ceph-osd@<osd_id>
```

9. 以 **root** 用户身份，重启 OSD 节点：

```
# shutdown -r now
```

10. 移到下一 OSD 节点。

注意

如果设置了 **noout** 和 **norebalance** 标记，存储集群将处于 **HEALTH_WARN** 状态

```
$ ceph health
HEALTH_WARN noout,norebalance flag(s) set
```

升级 Ceph 存储集群后，请取消设置之前设置的 OSD 标志并验证存储集群状态。

在 monitor 节点上，升级所有 OSD 节点后，取消设置 **noout** 和 **norebalance** 标志：

```
# ceph osd unset noout
# ceph osd unset norebalance
```

此外，执行 **ceph osd require-osd-release <release>** 命令。此命令可确保没有更多具有 Red Hat Ceph Storage 2.3 的 OSD 添加到存储集群中。如果不运行此命令，存储状态将为 **HEALTH_WARN**。

```
# ceph osd require-osd-release luminous
```

其它资源

- 若要通过添加新 OSD 到存储集群来扩展存储容量，请参阅 Red Hat Ceph Storage 3 *管理指南*中的*添加 OSD* 部分。

升级 Ceph 对象网关节点

本节介绍将 Ceph 对象网关节点升级到更新版本的步骤。

先决条件

- 红帽建议将 Ceph 对象网关放在负载均衡器后面，如 HAProxy。如果您使用负载平衡器，请在没有提供请求时从负载均衡器中删除 Ceph 对象网关。
- 如果您使用自定义名称作为 region 池（在 **rgw_region_root_pool** 参数中指定的），请将 **rgw_zonegroup_root_pool** 参数添加到 Ceph 配置文件的 **[global]** 部分。将 **rgw_zonegroup_root_pool** 的值设置为与 **rgw_region_root_pool** 的值相同，例如：

```
[global]
rgw_zonegroup_root_pool = .us.rgw.root
```

流程

在存储群集中的每个 Ceph 对象网关节点上执行下列步骤。一次仅升级一个节点。

1. 如果您使用在线存储库安装 Red Hat Ceph Storage，请禁用 2 存储库。

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2.3-tools-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

2. 启用 Red Hat Ceph Storage 3 Tools 存储库：

```
[root@gateway ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

3. 停止 Ceph 对象网关进程(**ceph-radosgw**）：

```
# service ceph-radosgw stop
```

4. 更新 **ceph-radosgw** 软件包：

```
# yum update ceph-radosgw
```

5. 将新创建的 **/var/lib/ceph/radosgw/** 和 **/var/log/ceph/** 目录及其内容上的所有者和组权限更改为 **ceph**。

```
# chown -R ceph:ceph /var/lib/ceph/radosgw
# chown -R ceph:ceph /var/log/ceph
```

6. 如果将 SELinux 设置为在 enforcing 或 permissive 模式下运行，请指示它在下次引导时重新标记 SELinux 上下文。

```
# touch /.autorelabel
```

重要

重新标记可能需要很长时间才能完成，因为 SELinux 必须遍历每个文件系统并修复任何错误标记的文件。要排除要重新标记的目录，请在重启前将其添加到 **/etc/selinux/fixfiles_exclude_dirs** 文件中。

7. 启用 **ceph-radosgw** 进程。

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.<hostname>
```

将 **<hostname>** 替换为 Ceph 对象网关主机的名称，如 **gateway-node**。

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.gateway-node
```

8. 重新引导 Ceph 对象网关节点。

```
# shutdown -r now
```

9. 如果使用负载平衡器，请将 Ceph 对象网关节点重新添加到负载平衡器。

另请参阅

- *Red Hat Enterprise Linux 的 Ceph 对象网关指南*

升级 Ceph 客户端节点

Ceph 客户端是：

- Ceph 块设备

- OpenStack Nova 计算节点
- QEMU/KVM 管理程序
- 使用 Ceph 客户端侧库的任何自定义应用

红帽建议所有 Ceph 客户端运行与 Ceph 存储集群相同的版本。

先决条件

- 在升级软件包以避免发生意外错误时停止对 Ceph 客户端节点的所有 I/O 请求

流程

1. 如果您使用软件存储库安装 Red Hat Ceph Storage 2 客户端，请禁用存储库：

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2-tools-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

注意

如果为 Red Hat Ceph Storage 2 客户端执行基于 ISO 的安装，请跳过第一步。

2. 在客户端节点上，启用 Red Hat Ceph Storage Tools 3 存储库：

```
[root@gateway ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

3. 在客户端节点上，更新 **ceph-common** 软件包：

```
# yum update ceph-common
```

在升级 **ceph-common** 软件包后，重新启动依赖于 Ceph 客户端侧库的任何应用。

注意

如果您要升级运行 QEMU/KVM 实例的 OpenStack Nova 计算节点，或使用专用 QEMU/KVM 客户端，请停止并启动 QEMU/KVM 实例，因为在此情况下重新启动实例不起作用。

附录 H. 版本 2 和版本 3 之间的 ANSIBLE 变量更改

在 Red Hat Ceph Storage 3 中，位于 **/usr/share/ceph-ansible/group_vars/** 目录的配置文件中的某些变量已更改或已被删除。下表列出了所有更改：升级到版本 3 后，再次复制 **all.yml.sample** 和 **osds.yml.sample** 文件，以反映这些更改。详情请参阅 升级 Red Hat Ceph Storage 集群。

旧选项	新选项	File
ceph_rhcs_cdn_install	**ceph_repository_type: cdn**	**all.yml**
ceph_rhcs_iso_install	**ceph_repository_type: iso**	**all.yml**
ceph_rhcs	**ceph_origin: repository** 和 **ceph_repository: rhcs** (默认启用)	**all.yml**
journal_collocation	**osd_scenario：collocated**	**osds.yml**
raw_multi_journal	**osd_scenario: non-collocated**	**osds.yml**
raw_journal_devices	**dedicated_devices**	**osds.yml**
dmcrytpt_journal_collocation	**dmcrypt: true** + **osd_scenario: collocated**	**osds.yml**
dmcrypt_dedicated_journal	**dmcrypt: true** + **osd_scenario: non-collocated**	**osds.yml**

附录 I. 将现有 CEPH 集群导入到 ANSIBLE

您可以将 Ansible 配置为使用在没有 Ansible 的情况下部署的集群。例如，如果您将 Red Hat Ceph Storage 1.3 集群升级到版本 2，请按照以下步骤将其配置为使用 Ansible：

1. 从 1.3 手动升级到版本 2 后，在管理节点上安装和配置 Ansible。
2. 确保 Ansible 管理节点对集群中的所有 Ceph 节点进行免密码 **ssh** 访问。详情请查看 第 2.11 节 “为 Ansible 启用无密码 SSH”。
3. 以 **root** 用户身份，在 **/etc/ansible/** 目录中创建一个指向 Ansible **group_vars** 目录的符号链接：

```
# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

4. 以 **root** 用户身份，使用 **all.yml.sample** 文件中创建一个 **all.yml** 文件，并打开该文件进行编辑：

```
# cd /etc/ansible/group_vars
# cp all.yml.sample all.yml
# vim all.yml
```

5. 在 **group_vars/all.yml** 中，将 **generate_fsid** 设置为 **false**。
6. 通过执行 **ceph fsid** 获得当前集群 **fsid**。
7. 在 **group_vars/all.yml** 中设置检索到的 **fsid**。
8. 修改 **/etc/ansible/hosts** 中的 Ansible 清单，使其包含 Ceph 主机。在 **[mons]** 部分下添加监视器，在 **[osds]** 部分下的 OSD 和网关下的 **[rgws]** 部分下将其角色标识到 Ansible。
9. 确定 **ceph_conf_overrides** 已更新，使用用于 **all.yml** 文件中的 **[global]**、**[osd]**、**[mon]** 和 **[client]** 项的原始 **ceph.conf** 选项。
   在 **ceph_conf_overrides** 中不应添加 **osd journal**、**public_network** 和 **cluster_network** 等选项，因为它们已经是 **all.yml** 的一部分。仅应将不属于 **all.yml** 且位于原始 **ceph.conf** 中的选项添加到 **ceph_conf_overrides**。
10. 从 **/usr/share/ceph-ansible/** 目录运行 playbook。

```
# cd /usr/share/ceph-ansible/
# cp infrastructure-playbooks/take-over-existing-cluster.yml .
$ ansible-playbook take-over-existing-cluster.yml -u <username>
```

附录 J. 使用 ANSIBLE 清除 CEPH 集群

如果使用 Ansible 部署 Ceph 集群，并且希望清除集群，则使用位于 **infrastructure-playbooks** 目录中的 **purge-cluster.yml** Ansible playbook。

重要

清除 Ceph 集群将丢失存储在群集 OSD 上的数据。

在清除 Ceph 集群前...

检查 **osds.yml** 文件中的 **osd_auto_discovery** 选项。将此选项设置为 **true** 将导致清除失败。要防止失败，请在运行清除前执行以下步骤：

1. 在 **osds.yml** 文件中声明 OSD 设备。详情请查看 第 3.2 节 “安装 Red Hat Ceph Storage 集群” 。
2. 注释掉 **osds.yml** 文件中的 **osd_auto_discovery** 选项。

清除 Ceph 集群...

1. 以 **root** 用户身份，导航到 **/usr/share/ceph-ansible/** 目录：

```
# cd /usr/share/ceph-ansible
```

2. 以 **root** 用户身份，将 **purge-cluster.yml** Ansible playbook 复制到当前目录中：

```
# cp infrastructure-playbooks/purge-cluster.yml .
```

3. 运行 **purge-cluster.yml** Ansible playbook:

```
$ ansible-playbook purge-cluster.yml
```