手动安装

获取软件

获取Ceph软件有几种方法。最简单和最常用的方法是通过添加存储库来获取包，以便与包管理工具一起使用，例如APT或YUM。还可以从Ceph存储库中检索预编译的包。最后，可以检索原始码或克隆Ceph源代码存储库并自己构建Ceph。

获取包

要安装Ceph和其他支持软件，您需要从Ceph存储库中检索软件包。按照本指南获取包; 然后，继续安装Ceph对象存储。

获取包

获取包有两种方法：

- **添加存储库：**添加存储库是获取包的最简单方法，因为在大多数情况下，包管理工具将为您检索包和所有启用软件。但是，要使用此方法，群集中的每个 Ceph节点都必须具有Internet访问权限。
- **手动下载软件包：**如果您的环境不允许Ceph节点访问Internet，**手动**下载软件包是安装Ceph的便捷方式。

要求

所有Ceph部署都需要Ceph包（开发除外）。您还应该添加密钥和推荐的软件包。

- **密钥:(推荐)**无论是手动添加存储库还是下载包，都应下载密钥以验证包。如果您没有获得密钥，则可能会遇到安全警告。
- **Ceph :(必需)**所有Ceph部署都需要Ceph发布包，但使用开发包的部署除外（仅限开发，QA和前沿部署）。
- **Ceph开发:(可选)**如果您正在为Ceph开发，测试Ceph开发版本，或者如果您想从Ceph开发的最前沿功能，那么您可能会获得Ceph开发包。有关详细信息，请参阅 添加Ceph开发。

如果您打算手动下载软件包，请参阅部分下载软件包。

添加密钥

将密钥添加到系统的可信密钥列表中以避免出现安全警告。对于通用版本（例如hammer，jewel，luminous）和开发版本（release-name-rc1，release-name-rc2），使用release.asc密钥。

APT

要安装release.asc密钥，请执行以下操作：

```
wget -q -O- 'https://download.ceph.com/keys/release.asc' | sudo apt-key add -
```

RPM

要安装release.asc密钥，请执行以下操作：

```
sudo rpm --import 'https://download.ceph.com/keys/release.asc'
```

添加ceph

发布存储库使用release.asc密钥来验证包。要使用APT或YUM安装Ceph软件包，需要添加Ceph存储库。

您可以在以下位置找到Debian / Ubuntu（使用APT安装）的发行版：

```
https://download.ceph.com/debian-{release-name}
```

您可以在以下位置找到CentOS/RHEL（使用YUM安装）的发行版：

```
https://download.ceph.com/rpm-{release-name}
```

Ceph的主要版本总结如下：发布索引

每隔一个主要版本是长期稳定（LTS）版本。关键错误修正直到LTS退役。由于不再维护已退役的版本，我们建议用户定期升级其群集 - 最好是最新的LTS版本。

> **小技巧：**
> 对于非美国用户：您附近可能有一个镜像从哪里下载Ceph。有关更多信息，请参阅：Ceph镜像。
> **CN: China**: http://mirrors.ustc.edu.cn/ceph/

DEBIAN软件包

将Ceph软件包存储库添加到系统的APT源列表中。对于较新版本的Debian / Ubuntu，请在命令行上调用lsb_release -sc以获取短代号，并在以下命令中替换{codename}。

```
sudo apt-add-repository 'deb https://download.ceph.com/debian-luminous/{codename} main'
```

对于早期的Linux发行版，您可以执行以下命令：

```
echo deb https://download.ceph.com/debian-luminous/ $(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list
```

对于早期的Ceph版本，请将{release-name}替换为Ceph版本的名称。您可以在命令行上调用lsb_release -sc以获取短代号，并在以下命令中替换{codename}。

```
sudo apt-add-repository 'deb https://download.ceph.com/debian-{release-name}/ {codename} main'
```

对于较旧的Linux发行版，请替换{release-name}为发行版的名称：

```
echo deb https://download.ceph.com/debian-{release-name}/ $(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list
```

对于开发发布包，请将我们的包存储库添加到系统的APT源列表中。请参阅测试Debian存储库以获取支持的Debian和Ubuntu发行版的完整列表。

```
echo deb https://download.ceph.com/debian-testing/ $(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list
```

RPM包

对于主要版本，您可以在/etc/yum.repos.d 目录中添加Ceph条目。创建一个ceph.repo文件。在下面的例子中，用ceph的版本（例如hammer，jewel，luminous等）替换{ceph-

release}的主要版本，用Linux发行版（如，el7等）替换{distro}。您可以查看 https://download.ceph.com/rpm - {ceph-release} /目录以查看Ceph支持的发行版。一些Ceph包（例如，EPEL）必须优先于标准包，因此您必须确保设置 priority=2。

```
[ceph]
name=Ceph packages for $basearch
baseurl=https://download.ceph.com/rpm-{ceph-release}/{distro}/$basearch
enabled=1
priority=2
gpgcheck=1
gpgkey=https://download.ceph.com/keys/release.asc

[ceph-noarch]
 name=Ceph noarch packages
 baseurl=https://download.ceph.com/rpm-{ceph-release}/{distro}/noarch
 enabled=1
 priority=2
 gpgcheck=1
 gpgkey=https://download.ceph.com/keys/release.asc

 [ceph-source]
 name=Ceph source packages
 baseurl=https://download.ceph.com/rpm-{ceph-release}/{distro}/SRPMS
 enabled=0
 priority=2
 gpgcheck=1
 gpgkey=https://download.ceph.com/keys/release.asc
```

对于特定包，您可以通过按名称下载发行包来检索它们。我们的开发过程每3-4周就会生成一个新的Ceph版本。这些软件包比主要版本更快。开发包具有快速集成的新功能，同时在发布之前仍需要几周的QA。

存储库包会在本地系统上安装存储库详细信息以供使用yum。替换{distro}为您的Linux发行版，以及 {release}为Ceph的特定版本：

```
su -c 'rpm -Uvh https://download.ceph.com/rpms/{distro}/x86_64/ceph-{release}.el7.noarch.rpm'
```

您可以直接从以下位置下载RPM：

```
https://download.ceph.com/rpm-testing
```

添加CEPH开发

如果您正在开发Ceph并需要部署和测试特定的Ceph分支，请确保首先删除主要版本的存储库条目。

DEB包

我们自动为Ceph源代码库中的当前开发分支构建Ubuntu包。这些软件包仅供开发人员和QA使用。

将软件包存储库添加到系统的APT源列表中，但替换{BRANCH}为您要使用的分支（例如，wip-hack，master）。请参阅shaman页面以获取我们构建的完整分发列表。

```
curl -L https://shaman.ceph.com/api/repos/ceph/{BRANCH}/latest/ubuntu/$(lsb_release -sc)/repo/ | sudo tee /etc/apt/sources.list.d/shaman.list
```

> **注意**
> 如果存储库未就绪，则将返回HTTP 504

```
curl -L https://shaman.ceph.com/api/repos/ceph/master/53e772a45fdf2d211c0c383106a66e1feedec8fd/ubuntu/xenial/repo/ | sudo tee /etc/apt/sources.list.d/shaman.list
```

RPM包

对于当前开发分支，您可以向/etc/yum.repos.d目录添加Ceph条目 。该页面可用于检索回购文件的完整细节。它可以通过HTTP请求检索，例如：

```
curl -L https://shaman.ceph.com/api/repos/ceph/{BRANCH}/latest/centos/7/repo/ | sudo tee /etc/yum.repos.d/shaman.repo
```

latest在url中使用意味着它将确定哪个是最后一次构建的提交。或者，可以指定特定的sha1。对于CentOS 7和Ceph的主分支，它看起来像：

```
curl -L https://shaman.ceph.com/api/repos/ceph/master/53e772a45fdf2d211c0c383106a66e1feedec8fd/centos/7/repo/ | sudo tee /etc/apt/sources.list.d/shaman.list
```

下载包

如果您尝试在没有Internet访问的环境中安装在防火墙后面，则必须在尝试安装之前检索软件包（镜像了所有必需的依赖项）。

DEBIAN软件包

Ceph需要额外的第三方库。

- libaio1
- libsnappy1
- libcurl3
- curl
- libgoogle-perftools4
- google-perftools
- libleveldb1

存储库包会在本地系统上安装存储库详细信息以供使用apt。替换{release}为最新的Ceph版本。替换 {version}为最新的Ceph版本号。替换{distro}为您的Linux发行版代号。替换{arch}为CPU架构。

```
wget -q https://download.ceph.com/debian-{release}/pool/main/c/ceph/ceph_{version}{distro}_{arch}.deb
```

RPM包

Ceph需要额外的第三方库。要添加EPEL存储库，请执行以下命令：

```
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

Ceph需要以下包：

- snappy
- leveldb
- gdisk
- python-argparse
- gperftools-libs

目前为RHEL / CentOS7（el7）平台构建了软件包。存储库包会在本地系统上安装存储库详细信息以供使用yum。替换{distro}您的发行版。

```
su -c 'rpm -Uvh https://download.ceph.com/rpm-luminous/{distro}/noarch/ceph-{version}.{distro}.noarch.rpm'
```

例如，对于CentOS 7（el7）：

```
su -c 'rpm -Uvh https://download.ceph.com/rpm-luminous/el7/noarch/ceph-release-1-0.el7.noarch.rpm'
```

您可以直接从以下位置下载RPM：

```
https://download.ceph.com/rpm-luminous
```

对于早期的Ceph版本，请将{release-name}名称替换为Ceph版本的名称。您可以在命令行上调用lsb_release -sc以获取短代号。

```
su -c 'rpm -Uvh https://download.ceph.com/rpm-{release-
```

下载CEPH的发行TAR包

随着Ceph开发的进展，Ceph团队发布了新版本的源代码。您可以在此处下载Ceph版本的源代码TAR包。

克隆CEPH源代码库

您可以通过转到github Ceph Repository，选择一个分支（master默认情况下），然后单击 **Download ZIP**按钮来克隆Ceph源代码的Ceph分支。

要克隆整个git存储库，请安装并配置git。

安装git

在Debian / Ubuntu上安装git，请执行：

```
sudo apt-get install git
```

在CentOS / RHEL上安装git，请执行：

```
sudo yum install git
```

您还必须拥有一个github帐户。如果您还没有 github帐户，请访问github.com并注册。按照Set Up Git中设置git的说明进行操作 。

添加SSH密钥（可选）

如果您打算将代码提交给Ceph或使用SSH（git@github.com:ceph/ceph.git）进行克隆，则必须为github生成SSH密钥。

要为其生成SSH密钥github，请执行：

```
ssh-keygen
```

获取要添加到github帐户的密钥（以下示例假定您使用了默认文件路径）：

```
cat .ssh/id_rsa.pub
```

复制公钥。

转到您的github帐户，点击“帐户设置”（即“工具”图标）；然后，单击左侧导航栏上的“SSH密钥”。

单击“SSH密钥”列表中的“添加SSH密钥”，输入密钥的名称，粘贴生成的密钥，然后按“添加密钥”按钮。

克隆源

要克隆Ceph源代码存储库，请执行：

```
git clone --recursive https://github.com/ceph/ceph.git
```

一旦git clone执行，你应该有一个Ceph库的完整副本。

为确保维护存储库中包含的子模块的最新副本。运行git status将告诉您子模块是否已过期。

```
cd ceph
git status
```

如果您的子模块已过期，请运行：

```
git submodule update --force --init --recursive
```

选择分支

克隆源代码和子模块后，默认情况下，Ceph存储库将位于分支master上，即不稳定的开发分支。你也可以选择其他分支。

- master：不稳定的开发分支。
- stable：错误修复分支。
- next：发布候选分支。

```
git checkout master
```

编译ceph

您可以通过检索Ceph源代码并自行构建来获取Ceph软件。要构建Ceph，您需要设置开发环境，编译Ceph，然后安装在用户空间或构建软件包并安装软件包。

构建先决条件

查看此部分以了解Linux / Unix发行版是否有特定的先决条件。

在构建Ceph源代码之前，需要安装几个库和工具：

```
./install-deps.sh
```

> **注意**
> 一些支持Google内存分析器工具的发行版可能使用不同的包名称（例如libgoogle-perftools4）。

编译ceph

Ceph是使用cmake构建的。要构建Ceph，请cd到克隆的Ceph存储库并执行以下命令：

```
cd ceph
./do_cmake.sh
cd build
make
```

> **注意**
> 默认情况下，do_cmake.sh将构建一个ceph的调试版本，对于某些工作负载，它的执行速度可能会慢5倍。如果您想构建ceph可执行文件的发布版本，请将'-DCMAKE_BUILD_TYPE = RelWithDebInfo'传递给do_cmake.sh。
> **超线程**
> 您可以使用它来执行多个作业，具体取决于您的系统。例如，对于双核处理器可以更快地构建。make -jmake -j4

请参阅安装构建以在用户空间中安装构建。

编译CEPH包

要构建包，必须克隆Ceph存储库。您可以使用dpkg-buildpackage（Debian / Ubuntu）或rpmbuild（RPM Package Manager ）从最新代码创建安装包。

在多核CPU上构建时，请使用核心-j数量和内核数量* 2.例如，使用-j4双核处理器来加速构建。

高级包工具（APT）

要.deb为Debian / Ubuntu 创建包，请确保已克隆 Ceph存储库，安装了Build Prerequisites并已安装 debhelper：

```
sudo apt-get install debhelper
```

一旦安装了debhelper，就可以构建软件包：

```
sudo dpkg-buildpackage
```

对于多处理器CPU，请使用该-j选项来加速构建。

RPM包管理器

要创建.rpm包，请确保已克隆Ceph存储库，安装了Build Prerequisites并已安装rpm-build和 rpmdevtools：

```
yum install rpm-build rpmdevtools
```

安装工具后，设置RPM编译环境：

```
rpmdev-setuptree
```

获取RPM编译环境的源tar：

```
wget -P ~/rpmbuild/SOURCES/ https://download.ceph.com/tarballs/ceph-<version>.tar.bz2
```

或者来自欧盟的镜像：

```
wget -P ~/rpmbuild/SOURCES/ http://eu.ceph.com/tarballs/ceph-<version>.tar.bz2
```

提取specfile：

```
tar --strip-components=1 -C ~/rpmbuild/SPECS/ --no-anchored -xvjf ~/rpmbuild/SOURCES/ceph-<version>.tar.bz2 "ceph.spec"
```

构建RPM包：

```
rpmbuild -ba ~/rpmbuild/SPECS/ceph.spec
```

对于多处理器CPU，请使用该-j选项来加速构建。

安装软件

一旦你有Ceph软件（或添加了存储库），安装软件很容易。在群组的每个Ceph节点上安装包。您可以使用 ceph-deploy为您的存储群集安装Ceph，或使用包管理工具。如果您打算安装Ceph对象网关或QEMU，您应该为RHEL / CentOS和其他使用Yum的发行版安装Yum Priorities。

安装CEPH DEPLOY

该ceph-deploy工具使您能够设置和卸载Ceph集群，用于开发，测试和概念验证项目。

APT

使用apt安装ceph-deploy，执行以下命令：

```
sudo apt-get update && sudo apt-get install ceph-deploy
```

RPM

使用yum安装ceph-deploy，执行以下命令：

```
sudo yum install ceph-deploy
```

安装CEPH存储集群

本指南介绍了手动安装Ceph软件包。此过程仅适用于不适用部署工具，如安装用户 ceph-deploy，chef，juju等。

您还可以使用ceph-deploy安装Ceph软件包，这可能更方便，因为您可以ceph使用单个命令在多个主机上安装。

用APT安装

一旦您将发布或开发包添加到APT，您应该更新APT的数据库并安装Ceph：

```
sudo apt-get update && sudo apt-get install ceph ceph-mds
```

用RPM安装

要使用RPM安装Ceph，请执行以下步骤：

1. 安装yum-plugin-priorities。

```
sudo yum install yum-plugin-priorities
```

2. 确保/etc/yum/pluginconf.d/priorities.conf存在。

3. 确保priorities.conf启用插件。

```
[main]
enabled = 1
```

4. 确保您的YUM ceph.repo条目包括priority=2。有关详情，请参阅获取包

```
[ceph]
name=Ceph packages for $basearch
baseurl=https://download.ceph.com/rpm-{ceph-release}/{distro}/$basearch
enabled=1
priority=2
gpgcheck=1
gpgkey=https://download.ceph.com/keys/release.asc

[ceph-noarch]
name=Ceph noarch packages
baseurl=https://download.ceph.com/rpm-{ceph-release}/{distro}/noarch
enabled=1
priority=2
gpgcheck=1
gpgkey=https://download.ceph.com/keys/release.asc

[ceph-source]
name=Ceph source packages
baseurl=https://download.ceph.com/rpm-{ceph-release}/{distro}/SRPMS
enabled=0
priority=2
gpgcheck=1
gpgkey=https://download.ceph.com/keys/release.asc
```

5. 安装必备软件包：

```
sudo yum install snappy leveldb gdisk python-argparse gperftools-libs
```

添加了发行版或开发包，或添加了 ceph.repo文件后/etc/yum.repos.d，即可安装Ceph软件包。

```
sudo yum install ceph
```

安装构建

如果从源代码构建Ceph，可以通过执行以下命令在用户空间中安装Ceph：

```
sudo make install
```

如果您在本地安装Ceph，make则会将可执行文件放入 usr/local/bin。您可以将Ceph配置文件添加到 usr/local/bin目录以从单个目录运行Ceph。

安装CEPH对象网关

从*firefly*（v0.80）开始，Ceph对象网关在Civetweb上运行（嵌入到ceph-radosgw守护进程中）而不是Apache和FastCGI。使用Civetweb简化了Ceph对象网关的安装和配置。

> **注意**
> 要运行Ceph对象网关服务，您应该有一个正在运行的Ceph存储群集，并且网关主机应该可以访问公共网络。
> 在版本0.80中，Ceph对象网关不支持SSL。您可以使用SSL设置反向代理服务器，以将HTTP请求作为HTTP请求分发给CivetWeb。

执行预安装程序

请参阅Preflight并在Ceph对象网关节点上执行预安装过程。具体来说，您应该禁用Ceph Deploy用户的requiretty，将SELinux设置为Permissive并设置具有无密码的Ceph Deploy用户sudo。对于Ceph对象网关，您需要打开Civetweb将在生产中使用的端口。

> **注意**
> Civetweb 7480默认在端口上运行。

安装CEPH对象网关

从管理服务器的工作目录中，在Ceph对象网关节点上安装Ceph对象网关包。例如：

```
ceph-deploy install --rgw <gateway-node1> [<gateway-node2> ...]
```

该ceph-common软件包是一个依赖，那么ceph-deploy将安装它。该cephCLI工具旨在为管理员。要使Ceph对象网关节点成为管理员节点，请从管理服务器的工作目录中执行以下命令：

```
ceph-deploy admin <node-name>
```

创建网关实例

从管理服务器的工作目录中，在Ceph对象网关上创建Ceph对象网关的实例。例如：

```
ceph-deploy rgw create <gateway-node1>
```

网关运行后，您应该能够在端口7480 上使用未经身份验证的请求访问它，如下所示：

```
http://client-node:7480
```

如果网关实例工作正常，您应该收到如下响应：

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
 <Owner>
 <ID>anonymous</ID>
 <DisplayName></DisplayName>
 </Owner>
 <Buckets>
 </Buckets>
</ListAllMyBucketsResult>
```

如果您在任何时候遇到麻烦并且想要重新开始，请执行以下操作以清除配置：

```
ceph-deploy purge <gateway-node1> [<gateway-node2>]
ceph-deploy purgedata <gateway-node1> [<gateway-node2>]
```

如果执行清除操作，则必须重新安装Ceph。

更改默认端口

Civetweb 7480默认在端口上运行。要更改默认端口（例如，更改为端口80），请在管理服务器的工作目录中修改Ceph配置文件。添加标题为“[client.rgw.<gateway-node>]的节，并替换<gateway-node>为Ceph对象网关节点的短节点名称”的部分（即hostname -s）。

> **注意**
> 由于11.0.1版本中，Ceph的对象网关**不**支持SSL。有关如何设置它的信息，请参阅将SSL与Civetweb一起使用。

例如，如果您的节点名称是gateway-node1，请在以下部分之后添加如下所示的[global]部分：

```
[client.rgw.gateway-node1]
rgw_frontends = "civetweb port=80"
```

> **注意**
> 确保port=<port-number>在rgw_frontends键/值对之间不留空格。该[client.rgw.gateway-node1] 标题将Ceph配置文件的这一部分标识为配置Ceph存储集群客户端，其中客户端类型是Ceph对象网关（即rgw），并且实例的名称是 gateway-node1。

将更新的配置文件推送到Ceph对象网关节点（和其他Ceph节点）：

```
ceph-deploy --overwrite-conf config push <gateway-node> [<other-nodes>]
```

要使新端口设置生效，请重新启动Ceph对象网关：

```
sudo systemctl restart ceph-radosgw.service
```

最后，检查以确保您选择的端口在节点的防火墙（例如，端口80）上打开。如果未打开，请添加端口并重新加载防火墙配置。如果您使用firewalld守护程序，请执行：

```
sudo firewall-cmd --list-all
sudo firewall-cmd --zone=public --add-port 80/tcp --permanent
sudo firewall-cmd --reload
```

如果您使用iptables，执行：

```
sudo iptables --list
sudo iptables -I INPUT 1 -i <iface> -p tcp -s <ip-address>/<netmask> --dport 80 -j ACCEPT
```

使用Ceph对象网关节点的相关值替换<iface>和<ip-address>/<netmask>。完成配置iptables后，请确保使更改持久化，以便在Ceph对象网关节点重新启动时生效。执行：

```
sudo apt-get install iptables-persistent
```

终端UI将打开。选择yes提示将当前IPv4iptables规则保存到/etc/iptables/rules.v4，当前IPv6 iptables规则/etc/iptables/rules.v6。

IPv4您在前面步骤中设置的iptables规则将被加载到/etc/iptables/rules.v4并将在重新启动后保持不变。

如果IPv4在安装后添加新的iptables规则， iptables-persistent则必须将其添加到规则文件中。在这种情况下，以root用户身份执行以下操作：

```
iptables-save > /etc/iptables/rules.v4
```

使用SSL CIVETWEB

在使用SSL与civetweb之前，需要一个证书，该证书将与用于访问Ceph对象网关的主机名相匹配。您可能希望获得具有*备用名称*字段的一个以获得更大的灵活性。如果您打算使用S3样式的子域（将通配符添加到DNS），则需要使用*通配符*证书。

Civetweb要求在一个文件中提供服务器密钥，服务器证书和任何其他CA或中间证书。这些项目中的每一项都必须是*pem*形式。由于组合文件包含密钥，因此应保护其免受未经授权的访问。

要配置ssl操作，请附加s到端口号。例如：

```
[client.rgw.gateway-node1]
rgw_frontends = civetweb port=443s ssl_certificate=/etc/ceph/private/keyandcert.pem
```

在Luminous中新添加的功能。

此外，通过+在配置中将它们分开，可以使civetweb绑定到多个端口。这允许ssl和非ssl连接都由单个rgw实例托管的用例。例如：

```
[client.rgw.gateway-node1]
rgw_frontends = civetweb port=80+443s ssl_certificate=/etc/ceph/private/keyandcert.pem
```

其他CIVETWEB配置选项

可以在文件的**Ceph对象网关**部分中为嵌入式Civetweb Web服务器调整一些其他配置选项ceph.conf。可以在HTTP前端中找到支持的选项列表，包括示例。

从APACHE 迁移到CIVETWEB

如果您使用Ceph Storage v0.80或更高版本在Apache和FastCGI上运行Ceph对象网关，那么您已经在运行Civetweb - 它从ceph-radosgw守护程序开始， 默认情况下它在端口7480上运行，因此它不会与您的Apache和FastCGI安装以及其他常用的Web服务端口冲突。迁移到使用Civetweb涉及删除Apache安装。然后，您必须从Ceph配置文件中删除Apache和FastCGI设置并重置rgw_frontends为Civetweb。

返回参考安装Ceph对象网关的说明 ceph-deploy，请注意配置文件只有一个设置 rgw_frontends（假设您选择更改默认端口）。该ceph-deploy实用程序为您生成数据目录和密钥环 - 将密钥环置于其中/var/lib/ceph/radosgw/{rgw-intance}。守护程序在默认位置查找，而您可能在Ceph配置文件中指定了不同的设置。由于您已经拥有密钥和数据

目录，因此如果您使用的是默认路径以外的其他内容，则需要在Ceph配置文件中维护这些路径。

基于Apache的部署的典型Ceph对象网关配置文件类似于以下内容：

在Red Hat Enterprise Linux上：

```
[client.radosgw.gateway-node1]
host = {hostname}
keyring = /etc/ceph/ceph.client.radosgw.keyring
rgw socket path = ""
log file = /var/log/radosgw/client.radosgw.gateway-node1.log
rgw frontends = fastcgi socket\_port=9000 socket\_host=0.0.0.0
rgw print continue = false
```

在Ubuntu上：

```
[client.radosgw.gateway-node]
host = {hostname}
keyring = /etc/ceph/ceph.client.radosgw.keyring
rgw socket path = /var/run/ceph/ceph.radosgw.gateway.fastcgi.sock
log file = /var/log/radosgw/client.radosgw.gateway-node1.log
```

要修改它以便与Civetweb一起使用，只需删除特定于Apache的设置，例如rgw_socket_path和rgw_print_continue。然后，更改 rgw_frontends设置以反映Civetweb而不是Apache FastCGI前端，并指定要使用的端口号。例如：

```
[client.radosgw.gateway-node1]
host = {hostname}
keyring = /etc/ceph/ceph.client.radosgw.keyring
log file = /var/log/radosgw/client.radosgw.gateway-node1.log
rgw_frontends = civetweb port=80
```

最后，重新启动Ceph对象网关。在Red Hat Enterprise Linux上执行：

```
sudo systemctl restart ceph-radosgw.service
```

在Ubuntu上执行：

```
sudo service radosgw restart id=rgw.<short-hostname>
```

如果您使用的是未打开的端口号，则还需要在防火墙上打开该端口。

配置桶分片

Ceph对象网关将存储桶索引数据存储在index_pool默认值.rgw.buckets.index中。有时，用户喜欢在一个存储桶中放置许多对象（数十万到数百万个对象）。如果不使用网关管理界面为每个存储桶设置最大对象数的配额，则当用户将大量对象放入存储桶时，存储桶索引会遭受严重的性能下降。

在Ceph 0.94中，您可以对存储桶索引进行分片，以便在每个存储桶允许大量对象时帮助防止性能瓶颈。该rgw_override_bucket_index_max_shards设置允许您设置每个桶的最大分片数。默认值为0，表示默认情况下桶索引分片处于关闭状态。

要打开桶索引分片，请设置rgw_override_bucket_index_max_shards 为大于的值0。

对于简单配置，您可以添加rgw_override_bucket_index_max_shards 到Ceph配置文件中。添加它[global]以创建系统范围的值。您也可以在Ceph配置文件中为每个实例设置它。

在Ceph配置文件中更改了桶分片配置后，重新启动网关。在Red Hat Enterprise Linux上执行：

```
sudo systemctl restart ceph-radosgw.service
```

在Ubuntu上执行：

```
sudo service radosgw restart id=rgw.<short-hostname>
```

对于联合配置，每个区域可能具有不同index_pool 的故障转移设置。要使区域组区域的值保持一致，您可以rgw_override_bucket_index_max_shards在网关的区域组配置中进行设置。例如：

```
radosgw-admin zonegroup get > zonegroup.json
```

打开zonegroup.json文件并编辑bucket_index_max_shards每个命名区域的设置。保存zonegroup.json文件并重置zonegroup。例如：

```
radosgw-admin zonegroup set < zonegroup.json
```

更新区域组后，更新并提交期间。例如：

```
radosgw-admin period update --commit
```

> **注意**
> 将索引池（对于每个区域，如果适用）映射到基于SSD的OSD的CRUSH规则也可以有助于桶索引性能。

将通配符添加到DNS

要将Ceph用于S3样式的子域（例如，bucket-name.domain-name.com），您需要将通配符添加到与ceph-radosgw守护程序一起使用的DNS服务器的DNS记录中 。

还必须在Ceph配置文件中使用该设置指定DNS的地址。rgw dns name ={hostname}

对于dnsmasq，添加以下地址设置，并在主机名前添加一个点（.）：

```
address=/.{hostname-or-fqdn}/{host-ip-address}
```

例如：

```
address=/.gateway-node1/192.168.122.75
```

对于bind，请在DNS记录中添加通配符。例如：

```
$TTL 604800
@ IN SOA gateway-node1. root.gateway-node1. (
 2 ; Serial
 604800 ; Refresh
```

```
 86400 ; Retry
 2419200 ; Expire
 604800 ) ; Negative Cache TTL
;
@ IN NS gateway-node1.
 @ IN A 192.168.122.113
 * IN CNAME @
```

重新启动DNS服务器并使用子域ping您的服务器，以确保您的DNS配置按预期工作：

```
ping mybucket.{hostname}
```

例如：

```
ping mybucket.gateway-node1
```

添加调试（如果需要）

完成设置过程后，如果遇到配置问题，可以[global]在Ceph配置文件的部分添加调试并重新启动网关以帮助解决任何配置问题。例如：

```
[global]
#append the following in the global section.
debug ms = 1
debug rgw = 20
```

使用网关

要使用REST接口，首先要为S3接口创建一个初始Ceph对象网关用户。然后，为Swift接口创建一个子用户。然后，您需要验证创建的用户是否能够访问网关。

为S3 ACCESS创建RADOSGW用户

一个radosgw用户需要创建并授予访问权限。该命令将提供有关其他命令选项的信息。

man radosgw-admin

要创建用户，请在以下位置执行以下操作：gateway host

```
sudo radosgw-admin user create --uid="testuser" --display-name="First Use
r"
```

命令的输出将如下所示：

```
{
 "user_id": "testuser",
 "display_name": "First User",
 "email": "",
 "suspended": 0,
 "max_buckets": 1000,
 "subusers": [],
 "keys": [{
 "user": "testuser",
  "access_key": "I0PJDPCIYZ665MW88W9R",
```

```
 "secret_key": "dxaXZ8U90SXydYzyS5ivamEP20hkLSUViiaR+ZDA"
 }],
 "swift_keys": [],
 "caps": [],
 "op_mask": "read, write, delete",
 "default_placement": "",
 "placement_tags": [],
 "bucket_quota": {
 "enabled": false,
 "max_size_kb": -1,
 "max_objects": -1
 },
 "user_quota": {
 "enabled": false,
 "max_size_kb": -1,
 "max_objects": -1
 },
 "temp_url_keys": []
}
```

> **注意**
> 访问验证的值keys->access_key和keys->secret_key。

创建一个SWIFT用户

如果需要这种访问，则需要创建Swift子用户。创建Swift用户是一个两步过程。第一步是创建用户。第二是创建密钥。

执行以下步骤：gateway host

创建Swift用户：

```
sudo radosgw-admin subuser create --uid=testuser --subuser=testuser:swift
--access=full
```

输出将如下所示：

```
{
 "user_id": "testuser",
 "display_name": "First User",
 "email": "",
 "suspended": 0,
 "max_buckets": 1000,
```

```
 "subusers": [{
 "id": "testuser:swift",
 "permissions": "full-control"
 }],
 "keys": [{
 "user": "testuser:swift",
 "access_key": "3Y1LNW4Q6X0Y53A52DET",
 "secret_key": ""
 }, {
 "user": "testuser",
 "access_key": "I0PJDPCIYZ665MW88W9R",
 "secret_key": "dxaXZ8U90SXydYzyS5ivamEP20hkLSUViiaR+ZDA"
 }],
 "swift_keys": [],
 "caps": [],
 "op_mask": "read, write, delete",
 "default_placement": "",
 "placement_tags": [],
 "bucket_quota": {
 "enabled": false,
 "max_size_kb": -1,
 "max_objects": -1
 },
 "user_quota": {
 "enabled": false,
 "max_size_kb": -1,
 "max_objects": -1
 },
 "temp_url_keys": []
 }
```

创建密钥:

```
 sudo radosgw-admin key create --subuser=testuser:swift --key-type=swift --gen-secret
```

输出将如下所示:

```
{
 "user_id": "testuser",
 "display_name": "First User",
 "email": "",
 "suspended": 0,
 "max_buckets": 1000,
```

```
"subusers": [{
"id": "testuser:swift",
"permissions": "full-control"
 }],
 "keys": [{
 "user": "testuser:swift",
 "access_key": "3Y1LNW4Q6X0Y53A52DET",
 "secret_key": ""
 }, {
 "user": "testuser",
 "access_key": "I0PJDPCIYZ665MW88W9R",
 "secret_key": "dxaXZ8U90SXydYzyS5ivamEP20hkLSUViiaR+ZDA"
 }],
 "swift_keys": [{
 "user": "testuser:swift",
 "secret_key": "244+fz2gSqoHwR3lYtSbIyomyPHf3i7rgSJrF\/IA"
 }],
 "caps": [],
 "op_mask": "read, write, delete",
 "default_placement": "",
 "placement_tags": [],
 "bucket_quota": {
 "enabled": false,
 "max_size_kb": -1,
 "max_objects": -1
 },
 "user_quota": {
 "enabled": false,
 "max_size_kb": -1,
 "max_objects": -1
 },
 "temp_url_keys": []
}
```


访问验证

测试S3访问

您需要编写并运行Python测试脚本以验证S3访问。S3访问测试脚本将连接到radosgw，创建一个新存储桶并列出所有存储桶。对值aws_access_key_id和 aws_secret_access_key从的值取access_key和secret_key由返回radosgw-admin命令。

执行以下步骤：

1. 您需要安装python-boto包：

```
sudo yum install python-boto
```

2. 创建Python脚本：

```
vi s3test.py
```

3. 将以下内容添加到文件中：

```
import boto.s3.connection

access_key = 'I0PJDPCIYZ665MW88W9R'
secret_key = 'dxaXZ8U90SXydYzyS5ivamEP20hkLSUViiaR+ZDA'
conn = boto.connect_s3(
 aws_access_key_id=access_key,
 aws_secret_access_key=secret_key,
 host='{hostname}', port={port},
 is_secure=False,
calling_format=boto.s3.connection.OrdinaryCallingFormat(),
 )

bucket = conn.create_bucket('my-new-bucket')
for bucket in conn.get_all_buckets():
 print "{name} {created}".format(
 name=bucket.name,
 created=bucket.creation_date,
 )
```

替换{hostname}为已配置网关服务gateway host的主机的主机名，即。替换{port}为您使用Civetweb的端口号。

4. 运行脚本：

```
python s3test.py
```

输出将如下所示：

```
my-new-bucket 2015-02-16T17:09:10.000Z
```

测试快速访问

可以通过swift命令行客户端验证Swift访问。该man swift命令将提供有关可用命令行选项的更多信息。

要安装swift客户端，请执行以下命令。在Red Hat Enterprise Linux上：

```
sudo yum install python-setuptools
sudo easy_install pip
sudo pip install --upgrade setuptools
sudo pip install --upgrade python-swiftclient
```

基于Debian的发行版：

```
sudo apt-get install python-setuptools
```

```
sudo easy_install pip
sudo pip install --upgrade setuptools
sudo pip install --upgrade python-swiftclient
```

要测试swift访问，请执行以下命令：

```
swift -V 1 -A http://{IP ADDRESS}:{port}/auth -U testuser:swift -K '{swift_secret_key}' list
```

替换{IP ADDRESS}为网关服务器的公共IP地址，替换{swift_secret_key}为radosgw-admin key create 命令为swift用户输出的值。将{port}替换为您使用Civetweb的端口号（例如，默认值为7480）。如果不更换端口，则默认为端口。

例如：

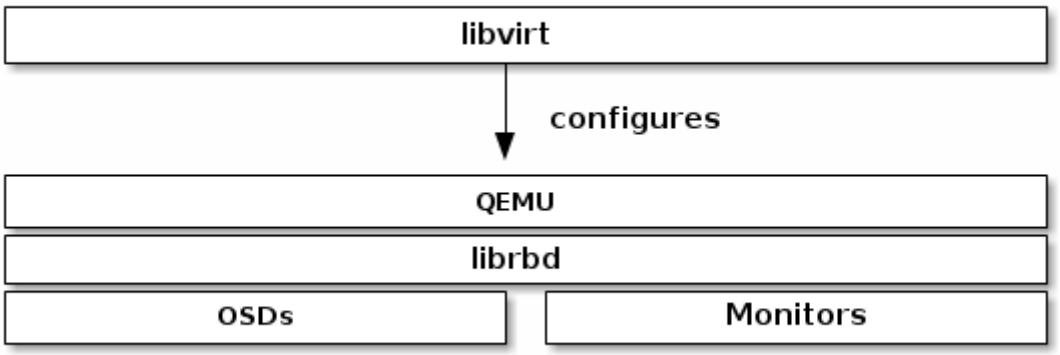
```
swift -V 1 -A http://10.19.143.116:7480/auth -U testuser:swift -K '244+fz2gSqoHwR3lYtSbIyomyPHf3i7rgSJrF/IA' list
```

输出应该是：

```
my-new-bucket
```

为块设备安装虚拟化

如果您打算将Ceph Block Devices和Ceph Storage Cluster用作虚拟机（VM）或云平台的后端， 则QEMU / KVM和 libvirt软件包对于启用VM和云平台非常重要。VM的示例包括：QEMU / KVM，XEN，VMWare，LXC，VirtualBox等。云平台的示例包括OpenStack，CloudStack，OpenNebula等。



安装QEMU

QEMU KVM可以与Ceph Block Devices进行交互librbd，这是将Ceph与云平台结合使用的一个重要特性。安装QEMU后，请参阅QEMU和阻止设备以了解用途。

DEBIAN软件包

QEMU软件包已合并到Ubuntu 12.04 Precise Pangolin及更高版本中。要安装QEMU，请执行以下命令：

```
sudo apt-get install qemu
```

RPM包

要安装QEMU，请执行以下命令：

1. 更新您的存储库。

```
sudo yum update
```

2. 为Ceph安装QEMU。

```
sudo yum install qemu-kvm qemu-kvm-tools qemu-img
```

3. 安装其他QEMU包（可选）：

```
sudo yum install qemu-guest-agent qemu-guest-agent-win32
```

构建QEMU

要从源构建QEMU，请使用以下过程：

```
cd {your-development-directory}
git clone git://git.qemu.org/qemu.git
cd qemu
./configure --enable-rbd
make; make install
```

安装LIBVIRT

要libvirt与Ceph一起使用，您必须拥有正在运行的Ceph存储群集，并且必须已安装并配置了QEMU。请参阅将libvirt与Ceph Block Device一起使用。

DEBIAN软件包

libvirt软件包被整合到Ubuntu 12.04 Precise Pangolin和更高版本的Ubuntu中。要libvirt在这些发行版上安装，请执行以下操作：

```
sudo apt-get update && sudo apt-get install libvirt-bin
```

RPM包

要libvirt与Ceph存储群集一起使用，您必须具有正在运行的Ceph存储群集，并且还必须安装具有rbd格式支持的QEMU版本。有关详细信息，请参阅安装QEMU。

libvirt软件包已合并到最近的CentOS / RHEL发行版中。要安装libvirt，请执行以下操作：

```
sudo yum install libvirt
```

建筑LIBVIRT

要从libvirt源构建，请克隆libvirt存储库并使用 AutoGen生成构建。然后，执行make并完成安装。例如：make install

```
git clone git://libvirt.org/libvirt.git
cd libvirt
./autogen.sh
make
sudo make install
```

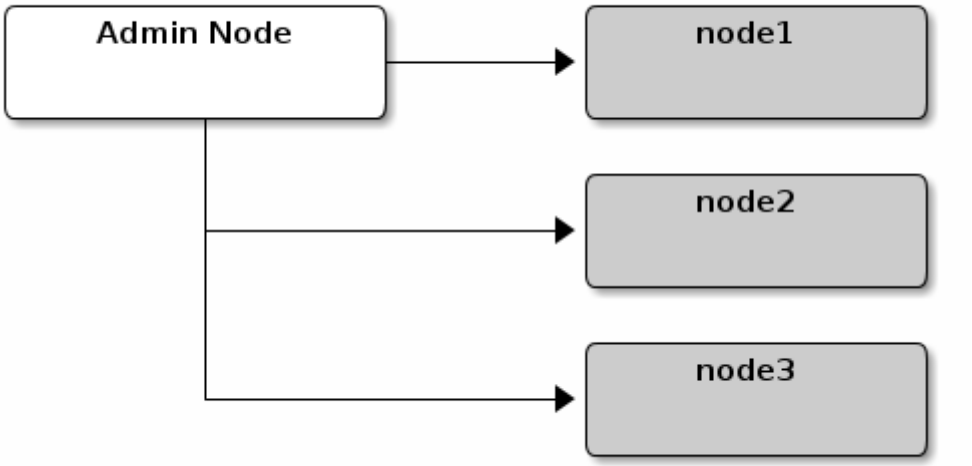
有关详细信息，请参阅libvirt安装。

- 安装Virtualization for Block

手动部署

所有Ceph集群都需要至少一个监视器，并且至少需要与集群中存储的对象的副本一样多的OSD。引导初始监视器是部署Ceph存储群集的第一步。监视器部署还为整个群集设置了重要标准，例如池的副本数，每个OSD的放置组数，心跳间隔，是否需要身份验证等。大多数这些值都是默认设置的，因此它是在设置群集进行生产时了解它们很有用。

遵循与安装（快速）相同的配置，我们将设置一个群集node1作为监控节点，node2并node3用于OSD节点。



监控引导

引导监视器（理论上是Ceph存储集群）需要许多东西：

- **唯一标识符：**它fsid是群集的唯一标识符，代表Ceph存储群集主要用于Ceph文件系统的日期的文件系统ID。Ceph现在也支持本机接口，块设备和对象存储网关接口，所以fsid有点用词不当。
- **群集名称：** Ceph群集具有群集名称，这是一个没有空格的简单字符串。默认群集名称是ceph，但您可以指定其他群集名称。当您使用多个群集并且需要清楚地了解正在使用哪个群集时，覆盖默认群集名称尤其有用。

例如，在多站点配置中运行多个群集时，群集名称（例如us-west，us-east）标识当前CLI会话的群集。**注意：**为了识别群集名称命令行界面上，指定与所述群集名称Ceph的配置文件（例如，ceph.conf，us-west.conf，us-east.conf，等等）。另请参阅CLI用法（ceph --cluster {cluster-name}）。

- **监视器名称：**群集中的每个监视器实例都具有唯一的名称。在通常的做法中，Ceph Monitor名称是主机名（我们建议每个主机使用一个Ceph Monitor，而不要将Ceph OSD守护进程与Ceph监视器混合使用）。您可以使用hostname -s检索短主机名。

- **监视器映射：**引导初始监视器需要您生成监视器映射。监视器映射需要fsid群集名称（或使用默认值），以及至少一个主机名及其IP地址。
- **监视密钥环**：监视器通过密钥相互通信。您必须生成带有监视器密钥的密钥环，并在引导初始监视器时提供它。
- **管理员密钥环**：要使用cephCLI工具，您必须拥有client.admin用户。因此，您必须生成管理员用户和密钥环，并且还必须将client.admin用户添加到监视器密钥环。

上述要求并不意味着创建Ceph配置文件。但是，作为最佳实践，我们建议创建Ceph配置文件并设置fsid，mon initialmembers and mon host 。

您也可以在运行时获取并设置所有监视器设置。但是，Ceph配置文件可能只包含覆盖默认值的设置。将设置添加到Ceph配置文件时，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轻松地维护集群。

过程如下：

1. 登录初始监控节点：

```
ssh {hostname}
```

例如：

```
ssh node1
```

2. 确保您有Ceph配置文件的目录。默认情况下，Ceph使用/etc/ceph。安装时ceph，安装程序将/etc/ceph自动创建目录。

```
ls /etc/ceph
```

注意：部署工具可能在清除群集时删除此目录（例如，ceph-deploy purgedata {node-name}ceph-deploy purge{node-name}）。

3. 创建Ceph配置文件。默认情况下，Ceph使用 ceph.conf，其中ceph反映了群集名称。

```
sudo vim /etc/ceph/ceph.conf
```

4. fsid为您的群集生成唯一ID（即 fsid）。

```
uuidgen
```

5. 将唯一ID添加到Ceph配置文件中。

```
fsid = {UUID}
```

例如：

```
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
```

6. 将初始监视器添加到Ceph配置文件中。

```
mon initial members = {hostname}[,{hostname}]
```

例如：

```
mon initial members = node1
```

7. 将初始监视器的IP地址添加到Ceph配置文件并保存文件。

```
mon host = {ip-address}[,{ip-address}]
```

例如：

```
mon host = 192.168.0.1
```

注意：您可以使用IPv6地址而不是IPv4地址，但必须设置ms bindipv6为true。有关网络配置的详细信息，请参阅网络配置。

8. 为群集创建密钥环并生成监控密钥。

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

9. 生成管理员密钥环，生成client.admin用户并将用户添加到密钥环。

```
sudo ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n client.admin --cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow *' --cap mgr 'allow *'
```

10. 生成bootstrap-osd密钥环，生成client.bootstrap-osd用户并将用户添加到密钥环。

```
sudo ceph-authtool --create-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring --gen-key -n client.bootstrap-osd --cap mon 'profile bootstrap-osd'
```

11. 将生成的密钥添加到ceph.mon.keyring。

```
sudo ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring
sudo ceph-authtool /tmp/ceph.mon.keyring --import-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring
```

12. 使用主机名，主机IP地址和FSID生成监控映射。保存为/tmp/monmap：

```
monmaptool --create --add {hostname} {ip-address} --fsid {uuid} /tmp/monmap
```

例如：

```
monmaptool --create --add node1 192.168.0.1 --fsid a7f64266-0894-4f1e-a635-d0aeaca0e993 /tmp/monmap
```

13. 在监视器主机上创建默认数据目录（或多个目录）。

```
sudo mkdir /var/lib/ceph/mon/{cluster-name}-{hostname}
```

例如：

```
sudo -u ceph mkdir /var/lib/ceph/mon/ceph-node1
```

有关详细信息，请参阅Monitor Config Reference - Data。

14. 使用监视器映射和密钥环填充监视器守护程序。

```
sudo -u ceph ceph-mon [--cluster {cluster-name}] --mkfs -i {hostname} --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring
```

例如：

```
sudo -u ceph ceph-mon --mkfs -i node1 --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring
```

15. Ceph配置文件的设置。常见设置包括以下内容：

```
[global]
fsid = {cluster-id}
mon initial members = {hostname}[, {hostname}]
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 copies in a degraded s
tate.
 osd pool default pg num = {n}
 osd pool default pgp num = {n}
 osd crush chooseleaf type = {n}
```

在上面的示例中，[global]配置的部分可能如下所示：

```
[global]
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
mon initial members = node1
mon host = 192.168.0.1
public network = 192.168.0.0/24
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd journal size = 1024
 osd pool default size = 3
 osd pool default min size = 2
 osd pool default pg num = 333
 osd pool default pgp num = 333
 osd crush chooseleaf type = 1
```

16. 启动监视器。

对于大多数发行版，服务现在通过systemd启动：

```
sudo systemctl start ceph-mon@node1
```

对于较旧的Debian / CentOS / RHEL，请使用sysvinit：

```
sudo /etc/init.d/ceph start mon.node1
```

17. 验证监视器是否正在运行。

```
ceph -s
```

您应该看到您启动的监视器已启动并正在运行的输出，并且您应该看到一个运行状况错误，指示放置组处于非活动状态。它应该看起来像这样：

```
cluster:
 id: a7f64266-0894-4f1e-a635-d0aeaca0e993
 health: HEALTH_OK

services:
 mon: 1 daemons, quorum node1
 mgr: node1(active)
 osd: 0 osds: 0 up, 0 in

data:
 pools: 0 pools, 0 pgs
 objects: 0 objects, 0 bytes
 usage: 0 kB used, 0 kB / 0 kB avail
 pgs:
```

注意：添加OSD并启动它们后，放置组健康错误应该消失。有关详细信息，请参阅添加OSD。

MANAGER守护程序配置

在运行ceph-mon守护程序的每个节点上，还应该设置ceph-mgr守护程序。

请参阅ceph-mgr管理员指南

添加的OSD

初始监视器运行后，应添加OSD。在有足够的OSD来处理对象的副本数量之前，您的群集无法达到active + clean 状态（例如，osd pool default size = 2需要至少两个OSD）。引导监视器后，您的群集具有默认的CRUSH映射; 但是，CRUSH映射没有映射到Ceph节点的任何Ceph OSD守护进程。

简短形式

Ceph提供了该ceph-volume实用程序，可以准备逻辑卷，磁盘或分区提供给ceph使用。该ceph-volume实用程序通过递增索引来创建OSD ID。此外，ceph-volume还会将新OSD添加到主机下的CRUSH映射中。执行CLI详细信息。该实用程序自动执行下面的长表格的步骤。要使用短格式过程创建前两个OSD，请执行以下操作并执行以下操作：

1. 创建OSD。

```
ssh {node-name}
sudo ceph-volume lvm create --data {data-path}
```

例如：

```
ssh node1
sudo ceph-volume lvm create --data /dev/hdd1
```

或者，创建过程可以分为两个阶段（准备和激活）：

1. 准备OSD。

```
ssh {node-name}
sudo ceph-volume lvm prepare --data {data-path} {data-path}
```

例如：

```
ssh node1
sudo ceph-volume lvm prepare --data /dev/hdd1
```

一旦准备，该OSD的ID和FSID需要激活。这些可以通过列出当前服务器中的OSD来获得：

```
sudo ceph-volume lvm list
```

2. 激活OSD：

```
sudo ceph-volume lvm activate {ID} {FSID}
```

例如：

```
sudo ceph-volume lvm activate 0 a7f64266-0894-4f1e-a635-d0aeaca0e993
```

文件存储

1. 创建OSD。

```
ssh {node-name}
sudo ceph-volume lvm create --filestore --data {data-path} --journal {jou
rnal-path}
```

例如：

```
ssh node1
sudo ceph-volume lvm create --filestore --data /dev/hdd1 --journal /dev/h
dd2
```

或者，创建过程可以分为两个阶段（准备和激活）：

1. 准备OSD。

```
ssh {node-name}
sudo ceph-volume lvm prepare --filestore --data {data-path} --journal {jo
urnal-path}
```

例如：

```
ssh node1
sudo ceph-volume lvm prepare --filestore --data /dev/hdd1 --journal /dev/
hdd2
```

一旦准备，该OSD的ID和FSID需要激活。这些可以通过列出当前服务器中的OSD来获得：

```
sudo ceph-volume lvm list
```

2. 激活OSD：

```
sudo ceph-volume lvm activate --filestore {ID} {FSID}
```

例如：

```
sudo ceph-volume lvm activate --filestore 0 a7f64266-0894-4f1e-a635-d0aeaca0e993
```

长表

如果没有任何帮助程序实用程序的好处，请创建一个OSD并使用以下过程将其添加到群集和CRUSH映射中。要使用长格式过程创建前两个OSD，请对每个OSD执行以下步骤。

> **注意**
> 此过程未描述使用dm-crypt“密码箱”的dm-crypt之上的部署。

1. 连接到OSD主机并成为root用户。

```
ssh {node-name}
sudo bash
```

2. 为OSD生成UUID。

```
UUID=$(uuidgen)
```

3. 为OSD生成cephx密钥。

```
OSD_SECRET=$(ceph-authtool --gen-print-key)
```

4. 创建OSD。请注意，如果您需要重复使用以前销毁的OSD ID，则可以提供ceph osd new作为附加参数。我们假设client.bootstrap-osd密钥出现已经存在。您也可以在存在该密钥的其他主机上执行client.admin命令：

```
ID=$(echo "{\"cephx_secret\": \"$OSD_SECRET\"}" | \
 ceph osd new $UUID -i - \
 -n client.bootstrap-osd -k /var/lib/ceph/bootstrap-osd/ceph.keyring)
```

也可以在JSON中包含一个crush_device_class属性来设置默认值以外的初始类（ssd或hdd基于自动检测的设备类型）。

5. 在新OSD上创建默认目录。

```
mkdir /var/lib/ceph/osd/ceph-$ID
```

6. 如果OSD用于OS驱动器以外的驱动器，请准备与Ceph一起使用，并将其安装到刚刚创建的目录中。

```
mkfs.xfs /dev/{DEV}
mount /dev/{DEV} /var/lib/ceph/osd/ceph-$ID
```

7. 将密钥写入OSD密钥环文件。

```
ceph-authtool --create-keyring /var/lib/ceph/osd/ceph-$ID/keyring \
 --name osd.$ID --add-key $OSD_SECRET
```

8. 初始化OSD数据目录。

```
ceph-osd -i $ID --mkfs --osd-uuid $UUID
```

9. 修复所有权

```
chown -R ceph:ceph /var/lib/ceph/osd/ceph-$ID
```

10. 将OSD添加到Ceph后，终稿OSD就在您的配置中。但是，它还没有运行。您必须先启动新的OSD才能开始接收数据。

对于现代systemd发行版：

```
systemctl enable ceph-osd@$ID
systemctl start ceph-osd@$ID
```

例如：

```
systemctl enable ceph-osd@12
systemctl start ceph-osd@12
```

添加

在下面的说明中，{id}是一个任意名称，例如机器的主机名。

1. 创建mds数据目录：

```
mkdir -p /var/lib/ceph/mds/{cluster-name}-{id}
```

2. 创建一个密钥环：

```
ceph-authtool --create-keyring /var/lib/ceph/mds/{cluster-name}-{id}/keyring --gen-key -n mds.{id}
```

3. 导入密钥环并设置上限：

```
ceph auth add mds.{id} osd "allow rwx" mds "allow" mon "allow profile mds" -i /var/lib/ceph/mds/{cluster}-{id}/keyring
```

4. 添加到ceph.conf。：

```
[mds.{id}]
host = {id}
```

5. 以手动方式启动守护进程：

```
ceph-mds --cluster {cluster-name} -i {id} -m {mon-hostname}:{mon-port} [-f]
```

6. 以正确的方式启动守护进程（使用ceph.conf条目）：

```
service ceph start
```

7. 如果启动守护程序失败并显示以下错误：

```
mds.-1.0 ERROR: failed to authenticate: (22) Invalid argument
```

然后确保你没有在全局部分的ceph.conf中设置密钥环; 把它移到客户端部分; 或添加特定于此mds守护程序的密钥环设置。并验证您在mds数据目录中看到相同的密钥和ceph auth get mds.{id}输出。

8. 现在您已准备好创建Ceph文件系统。

总结

启动并运行两个OSD后，您可以通过执行以下操作来观察放置组对等：

```
ceph -w
```

要查看树，请执行以下操作：

```
ceph osd tree
```

你应该看到看起来像这样的输出：

```
# id weight type name up/down reweight
```

```
-1 2 root default
-2 2 host node1
0 1 osd.0 up 1
-3 1 host node2
1 1 osd.1 up 1
```

要添加（或删除）其他监视器，请参阅添加/删除监视器。要添加（或删除）其他Ceph OSD守护进程，请参阅添加/删除OSD。

升级

随着Ceph的新版本变得可用，您可以升级群集以利用新功能。升级群集之前，请阅读升级文档。有时升级Ceph需要您遵循升级顺序。Ceph的每个版本都可能有其他步骤。在使用升级过程之前，请参阅发行版的发行说明文档以确定群集的特定于发行版的过程。

摘要

您可以在群集联机并投入使用时升级Ceph群集中的守护程序！某些类型的守护进程依赖于其他守护进程。例如，Ceph元数据服务器和Ceph对象网关依赖于Ceph监视器和Ceph OSD守护进程。我们建议按此顺序升级：

1. Ceph部署
2. Ceph监视器
3. Ceph OSD守护进程
4. Ceph元数据服务器
5. Ceph对象网关

作为一般规则，我们建议升级特定类型的所有ceph-mon守护进程（例如，所有守护进程，所有ceph-osd守护进程等），以确保它们都在同一版本中。我们还建议您在尝试在发布中执行新功能之前升级群集中的所有守护程序。

升级过程都比较简单，但不要看你的释放的发行说明文件在升级之前。基本过程包括三个步骤：

1. 使用ceph-deploy您的管理节点上升级为多个主机（使用包ceph-deployinstall命令），或登录到每个主机和升级包Ceph的使用你的发行版的包管理器。例如，在升级监视器时，ceph-deploy语法可能如下所示：

```
ceph-deploy install --release {release-name} ceph-node1[ ceph-node2]
ceph-deploy install --release firefly mon1 mon2 mon3
```

注意：该ceph-deploy install命令会将指定节点中的软件包从旧版本升级到您指定的版本。没有ceph-deploy upgrade命令。

2. 登录每个Ceph节点并重新启动每个Ceph守护程序。有关详细信息，请参阅操作群集
3. 确保您的群集健康。有关详细信息，请参阅监控群集

> **重要**
> 升级后台程序后，无法降级它。

CEPH DEPLOY

在升级Ceph守护进程之前，请升级该ceph-deploy工具。

```
sudo pip install -U ceph-deploy
```

要么：

```
sudo apt-get install ceph-deploy
```

要么：

```
sudo yum install ceph-deploy python-pushy
```

升级过程

以下部分描述了升级过程。

> **重要**
> Ceph的每个版本都可能有一些额外的步骤。请参阅您的版本发行说明文件的详细信息**之前**开始升级守护进程。

升级监视器

要升级监视器，请执行以下步骤：

1. 升级每个守护程序实例的Ceph包。

您可以使用ceph-deploy一次性处理所有监视节点。例如：

```
ceph-deploy install --release {release-name} ceph-node1[ ceph-node2]
ceph-deploy install --release hammer mon1 mon2 mon3
```

您还可以在每个节点上使用Linux发行版的软件包管理器。要在每个Debian / Ubuntu主机上手动升级软件包，请执行以下步骤：

```
ssh {mon-host}
sudo apt-get update && sudo apt-get install ceph
```

在CentOS / Red Hat主机上，执行以下步骤：

```
ssh {mon-host}
sudo yum update && sudo yum install ceph
```

2. 重启每台显示器。对于Ubuntu发行版，请使用：

```
sudo restart ceph-mon id={hostname}
```

对于CentOS / Red Hat / Debian发行版，请使用：

```
sudo /etc/init.d/ceph restart {mon-id}
```

对于部署的CentOS / Red Hat发行版ceph-deploy，通常使用监视器ID mon.{hostname}。

3. 确保每个监视器重新加入仲裁：

```
ceph mon stat
```

确保您已完成所有Ceph监视器的升级周期。

升级OSD

要升级Ceph OSD守护程序，请执行以下步骤：

1. 升级Ceph OSD守护进程包。

您可以使用ceph-deploy一次性处理所有Ceph OSD守护程序节点。例如：

```
ceph-deploy install --release {release-name} ceph-node1[ ceph-node2]
ceph-deploy install --release hammer osd1 osd2 osd3
```

您还可以使用每个节点上的软件包管理器使用您的发行版软件包管理器升级软件包 。

对于Debian / Ubuntu主机，请在每台主机上执行以下步骤：

```
ssh {osd-host}
sudo apt-get update && sudo apt-get install ceph
```

对于CentOS / Red Hat主机，请执行以下步骤：

```
ssh {osd-host}
sudo yum update && sudo yum install ceph
```

2. 重新启动OSD，N是OSD的编号。对于Ubuntu，请使用：

```
sudo restart ceph-osd id=N
```

对于主机上的多个OSD，您可以使用Upstart重新启动所有OSD。

```
sudo restart ceph-osd-all
```

对于CentOS / Red Hat / Debian发行版，请使用：

```
sudo /etc/init.d/ceph restart N
```

3. 确保每个升级的Ceph OSD守护程序重新加入群集：

```
ceph osd stat
```

确保您已完成所有Ceph OSD守护进程的升级周期。

升级元数据服务器

要升级Ceph元数据服务器，请执行以下步骤：

1. 升级Ceph Metadata Server包。您可以使用ceph-deploy一次解决所有Ceph Metadata Server节点，或使用每个节点上的包管理器。例如：

```
ceph-deploy install --release {release-name} ceph-node1
ceph-deploy install --release hammer mds1
```

要手动升级软件包，请在每个Debian / Ubuntu主机上执行以下步骤：

```
ssh {mon-host}
sudo apt-get update && sudo apt-get install ceph-mds
```

或者在CentOS / Red Hat主机上执行以下步骤：

```
ssh {mon-host}
sudo yum update && sudo yum install ceph-mds
```

2. 重启元数据服务器。对于Ubuntu，请使用：

```
sudo restart ceph-mds id={hostname}
```

对于CentOS / Red Hat / Debian发行版，请使用：

```
sudo /etc/init.d/ceph restart mds.{hostname}
```

对于部署的集群ceph-deploy，名称通常是您在创建时指定的名称或主机名。

3. 确保元数据服务器已启动并正在运行：

```
ceph mds stat
```

升级客户端

一旦在Ceph集群上升级了软件包并重新启动了守护程序，我们建议您在客户机节点上升级ceph-common和客户端库（librbd1和librados2）。

1. 升级包：

```
ssh {client-host}
apt-get update && sudo apt-get install ceph-common librados2 librbd1 python-rados python-rbd
```

2. 确保您拥有最新版本：

```
ceph --version
```

如果您没有最新版本，则可能需要卸载，自动删除依赖项并重新安装。