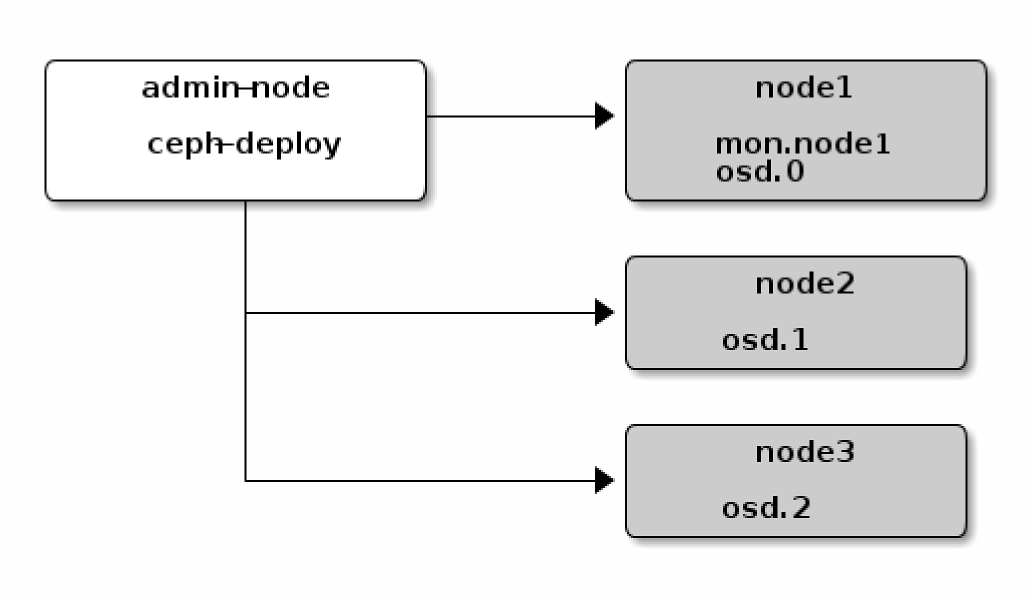
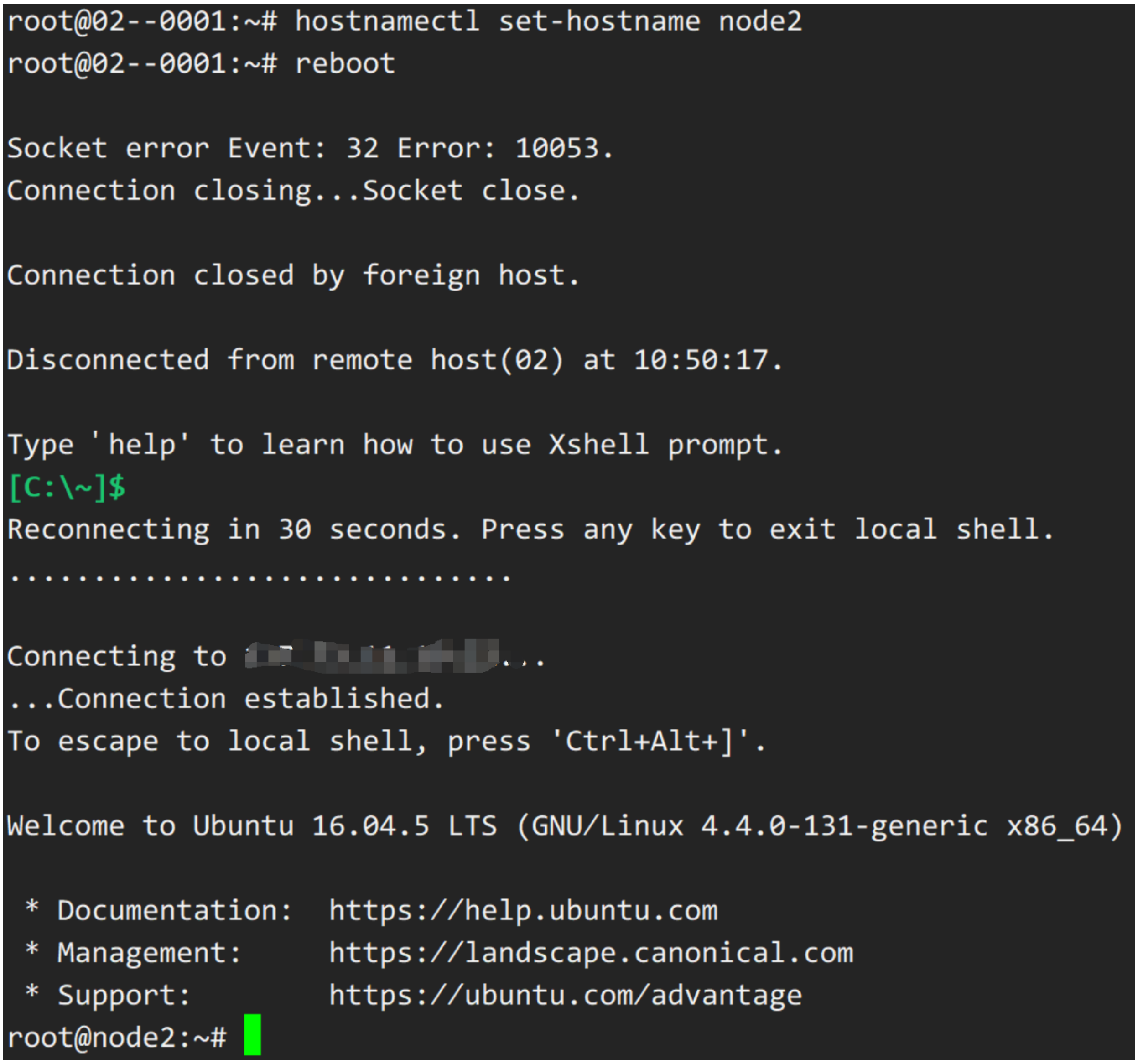
预配置

该ceph-deploy工具在管理节点上的目录之外运行 。任何具有网络连接、python环境和ssh（如Linux）的主机都应该可以工作。

在下面的描述中，节点指的是单个机器。

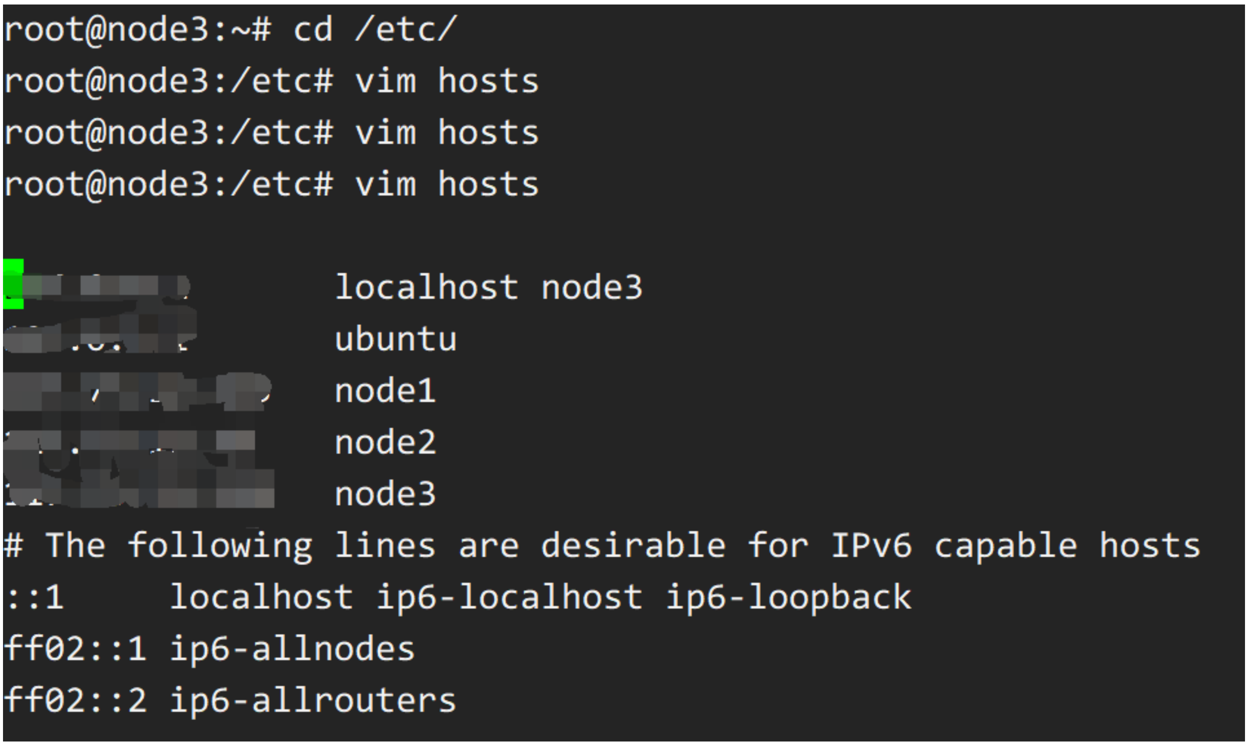


重命名主机并重启



```
root@02--0001:~# hostnamectl set-hostname node2
root@02--0001:~# reboot

Socket error Event: 32 Error: 10053.
Connection closing...Socket close.

Connection closed by foreign host.

Disconnected from remote host(02) at 10:50:17.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$
Reconnecting in 30 seconds. Press any key to exit local shell.
..............................

Connecting to [redacted]...
...Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-131-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
root@node2:~# 
```

配置hosts

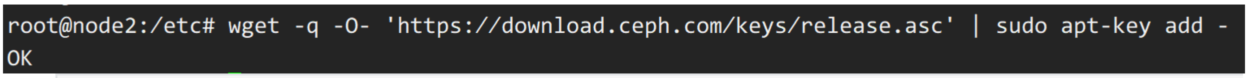


```
root@node3:~# cd /etc/
root@node3:/etc# vim hosts
root@node3:/etc# vim hosts
root@node3:/etc# vim hosts

[redacted]        localhost node3
[redacted]        ubuntu
[redacted]        node1
[redacted]        node2
[redacted]        node3
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

ceph-deploy安装

将Ceph存储库添加到ceph-deploy管理节点。然后，安装 ceph-deploy。

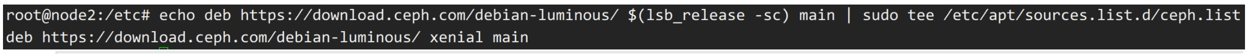
1. 添加发布密钥：



```
root@node2:/etc# wget -q -O- 'https://download.ceph.com/keys/release.asc' | sudo apt-key add -
OK
```

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

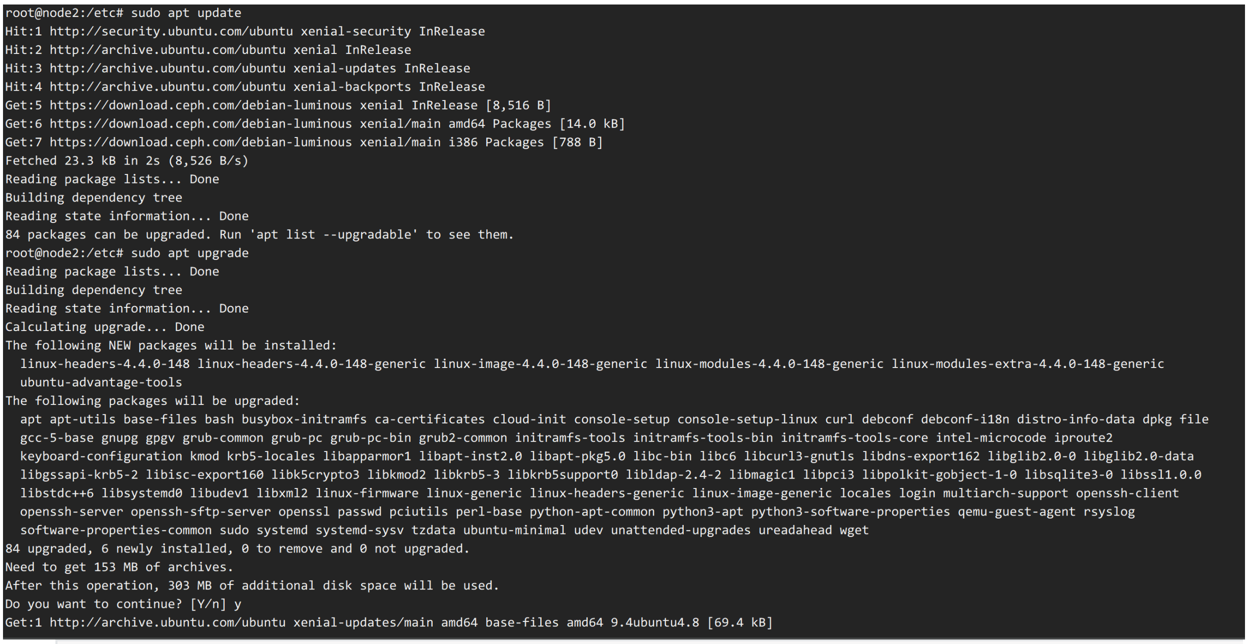
2. 将Ceph包添加到存储库。使用下面的命令并替换{ceph-stable-release}为稳定的Ceph版本（如luminous。）例如：



```
root@node2:/etc# echo deb https://download.ceph.com/debian-luminous/ $(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list
deb https://download.ceph.com/debian-luminous/ xenial main
```

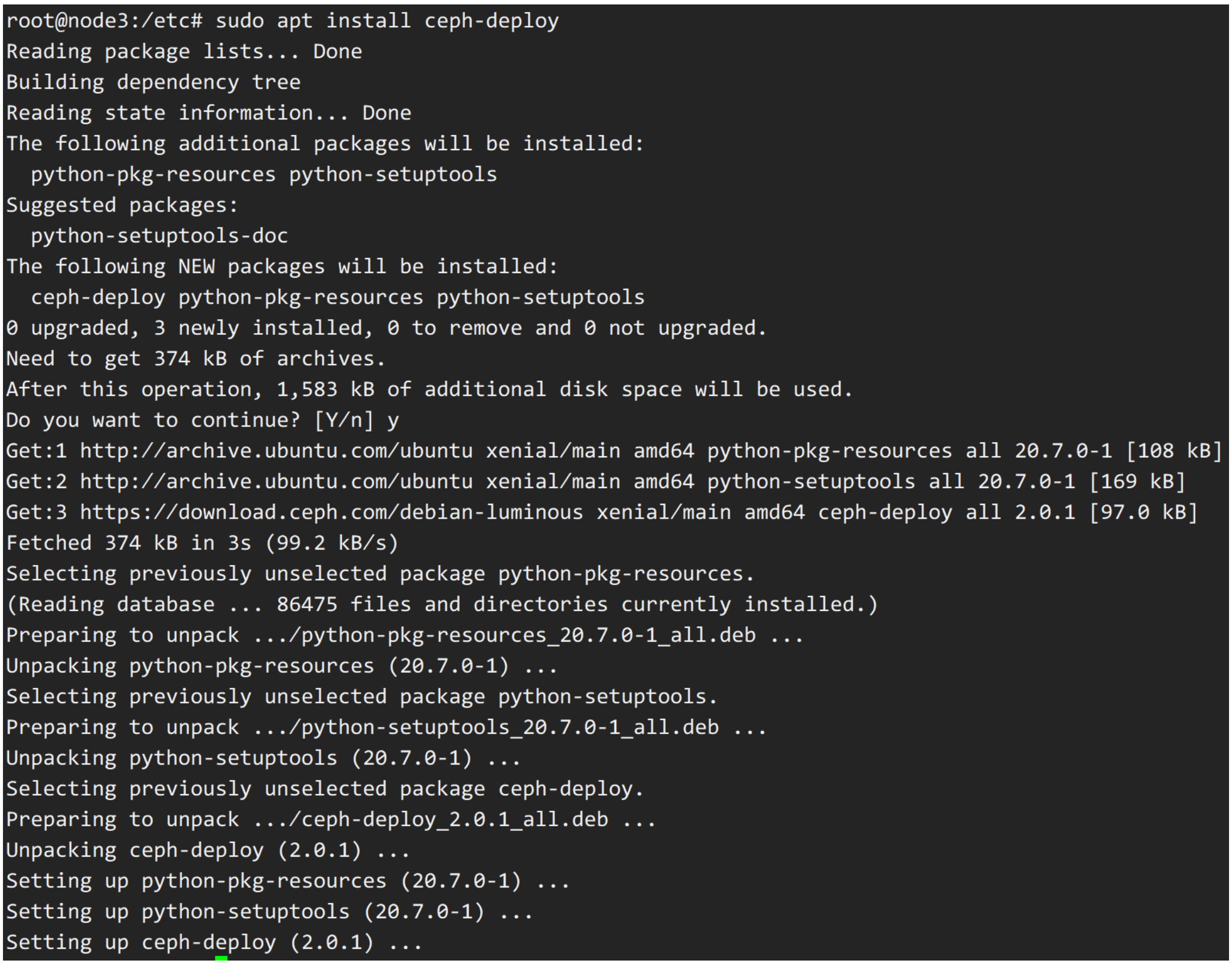
```
echo deb https://download.ceph.com/debian-{ceph-stable-release}/ $(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list
```

3. 更新您的存储库并安装ceph-deploy：



```
root@node2:/etc# sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu xenial InRelease
Hit:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu xenial-backports InRelease
Get:5 https://download.ceph.com/debian-luminous xenial InRelease [8,516 B]
Get:6 https://download.ceph.com/debian-luminous xenial/main amd64 Packages [14.0 kB]
Get:7 https://download.ceph.com/debian-luminous xenial/main i386 Packages [788 B]
Fetched 23.3 kB in 2s (8,526 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
84 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@node2:/etc# sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  linux-headers-4.4.0-148 linux-headers-4.4.0-148-generic linux-image-4.4.0-148-generic linux-modules-4.4.0-148-generic linux-modules-extra-4.4.0-148-generic
  ubuntu-advantage-tools
The following packages will be upgraded:
  apt apt-utils base-files bash busybox-initramfs ca-certificates cloud-init console-setup console-setup-linux curl debconf debconf-i18n distro-info-data dpkg file
  gcc-5-base gnupg gpgv grub-common grub-pc grub-pc-bin grub2-common initramfs-tools initramfs-tools-bin initramfs-tools-core intel-microcode iproute2
  keyboard-configuration kmod krb5-locales libapparmor1 libapt-inst2.0 libapt-pkg5.0 libc-bin libc6 libcurl3-gnutls libdns-export162 libglib2.0-0 libglib2.0-data
  libgssapi-krb5-2 libisc-export160 libk5crypto3 libkmod2 libkrb5-3 libkrb5support0 libldap-2.4-2 libmagic1 libpci3 libpolkit-gobject-1-0 libsqlite3-0 libssl1.0.0
  libstdc++6 libsystemd0 libudev1 libxml2 linux-firmware linux-generic linux-headers-generic linux-image-generic locales login multiarch-support openssh-client
  openssh-server openssh-sftp-server openssl passwd pciutils perl-base python-apt-common python3-apt python3-software-properties qemu-guest-agent rsyslog
  software-properties-common sudo systemd systemd-sysv tzdata ubuntu-minimal udev unattended-upgrades ureadahead wget
84 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 153 MB of archives.
After this operation, 303 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 base-files amd64 9.4ubuntu4.8 [69.4 kB]
```

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



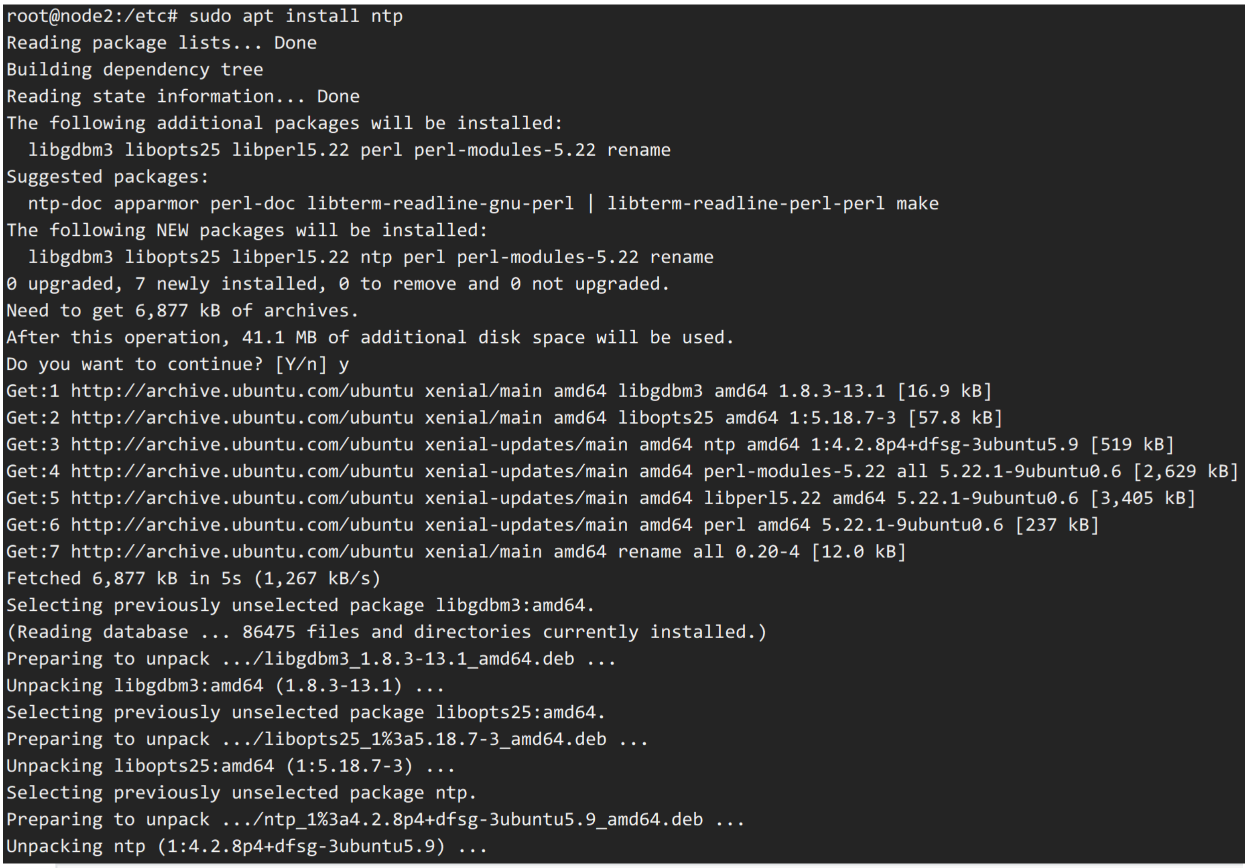
```
root@node3:/etc# sudo apt install ceph-deploy
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python-pkg-resources python-setuptools
Suggested packages:
  python-setuptools-doc
The following NEW packages will be installed:
  ceph-deploy python-pkg-resources python-setuptools
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 374 kB of archives.
After this operation, 1,583 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu xenial/main amd64 python-pkg-resources all 20.7.0-1 [108 kB]
Get:2 http://archive.ubuntu.com/ubuntu xenial/main amd64 python-setuptools all 20.7.0-1 [169 kB]
Get:3 https://download.ceph.com/debian-luminous xenial/main amd64 ceph-deploy all 2.0.1 [97.0 kB]
Fetched 374 kB in 3s (99.2 kB/s)
Selecting previously unselected package python-pkg-resources.
(Reading database ... 86475 files and directories currently installed.)
Preparing to unpack .../python-pkg-resources_20.7.0-1_all.deb ...
Unpacking python-pkg-resources (20.7.0-1) ...
Selecting previously unselected package python-setuptools.
Preparing to unpack .../python-setuptools_20.7.0-1_all.deb ...
Unpacking python-setuptools (20.7.0-1) ...
Selecting previously unselected package ceph-deploy.
Preparing to unpack .../ceph-deploy_2.0.1_all.deb ...
Unpacking ceph-deploy (2.0.1) ...
Setting up python-pkg-resources (20.7.0-1) ...
Setting up python-setuptools (20.7.0-1) ...
Setting up ceph-deploy (2.0.1) ...
```

Ceph节点设置

管理节点必须具有对Ceph节点的无密码SSH访问。当ceph-deploy以用户身份登录到Ceph节点时，该特定用户必须具有无密码sudo权限。

安装NTP

我们建议在Ceph节点上安装NTP（特别是在Ceph Monitor节点上）以防止时钟漂移引起的问题。



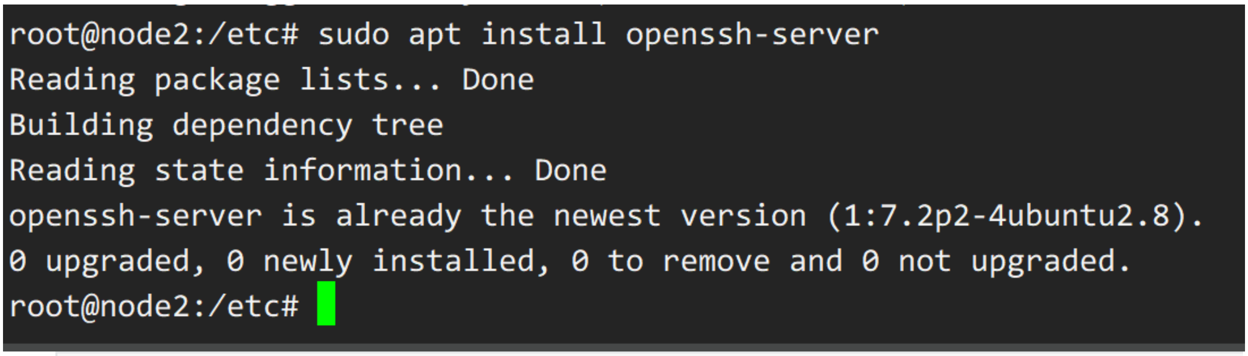
```
root@node2:/etc# sudo apt install ntp
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libgdbm3 libopts25 libperl5.22 perl perl-modules-5.22 rename
Suggested packages:
  ntp-doc apparmor perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl make
The following NEW packages will be installed:
  libgdbm3 libopts25 libperl5.22 ntp perl perl-modules-5.22 rename
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 6,877 kB of archives.
After this operation, 41.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu xenial/main amd64 libgdbm3 amd64 1.8.3-13.1 [16.9 kB]
Get:2 http://archive.ubuntu.com/ubuntu xenial/main amd64 libopts25 amd64 1:5.18.7-3 [57.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 ntp amd64 1:4.2.8p4+dfsg-3ubuntu5.9 [519 kB]
Get:4 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 perl-modules-5.22 all 5.22.1-9ubuntu0.6 [2,629 kB]
Get:5 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 libperl5.22 amd64 5.22.1-9ubuntu0.6 [3,405 kB]
Get:6 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 perl amd64 5.22.1-9ubuntu0.6 [237 kB]
Get:7 http://archive.ubuntu.com/ubuntu xenial/main amd64 rename all 0.20-4 [12.0 kB]
Fetched 6,877 kB in 5s (1,267 kB/s)
Selecting previously unselected package libgdbm3:amd64.
(Reading database ... 86475 files and directories currently installed.)
Preparing to unpack .../libgdbm3_1.8.3-13.1_amd64.deb ...
Unpacking libgdbm3:amd64 (1.8.3-13.1) ...
Selecting previously unselected package libopts25:amd64.
Preparing to unpack .../libopts25_1%3a5.18.7-3_amd64.deb ...
Unpacking libopts25:amd64 (1:5.18.7-3) ...
Selecting previously unselected package ntp.
Preparing to unpack .../ntp_1%3a4.2.8p4+dfsg-3ubuntu5.9_amd64.deb ...
Unpacking ntp (1:4.2.8p4+dfsg-3ubuntu5.9) ...
```

```
sudo apt install ntp
sudo apt-get install -y ntp ntpdate ntp-doc
ntpdate 0.us.pool.ntp.org
hwclock --systohc
systemctl enable ntp
systemctl start ntp
```

安装SSH服务器

对于**所有** Ceph节点，执行以下步骤：

1. 在每个Ceph节点上安装SSH服务器（如有必要）：



```
root@node2:/etc# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.2p2-4ubuntu2.8).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@node2:/etc#
```

```
sudo apt install openssh-server
```

2. 确保SSH服务器在**所有** Ceph节点上运行。

创建Ceph部署用户

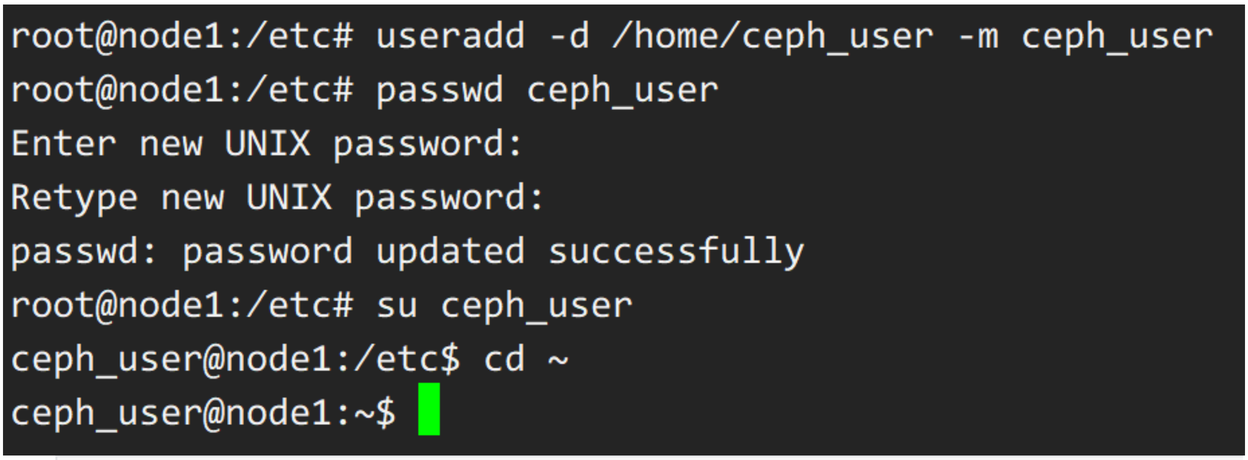
该ceph-deploy实用程序必须以具有无密码sudo权限的用户身份登录到Ceph节点，因为它需要在不提示输入密码的情况下安装软件和配置文件。

最新版本的ceph-deploy支持--username选项，因此可以指定任何具有无密码的用户sudo（包括root，但**不**建议这样做）。要使用ceph-deploy--username{username}ceph-deploy，所指定的用户必须具有对Ceph节点的无密码SSH访问权限，因为 不会提示您输入密码。

我们建议在**所有**集群节点上创建一个特定的用户用于ceph-deploy。请**不要**使用“ceph”作为用户名。集群统一的用户名可以提高易用性（不要求），但你应该避免使用明显的用户名，因为黑客通常使用蛮力破解它们（例如root， admin，{productname}）。以下过程描述了如何创建无密码sudo用户（替换 {username}成您定义的用户名）。

> **注意**
> 从Infernalis版本开始， “ceph”用户名是为Ceph守护进程保留的。如果Ceph节点上已存在“ceph”用户，则必须在尝试升级之前删除该用户。

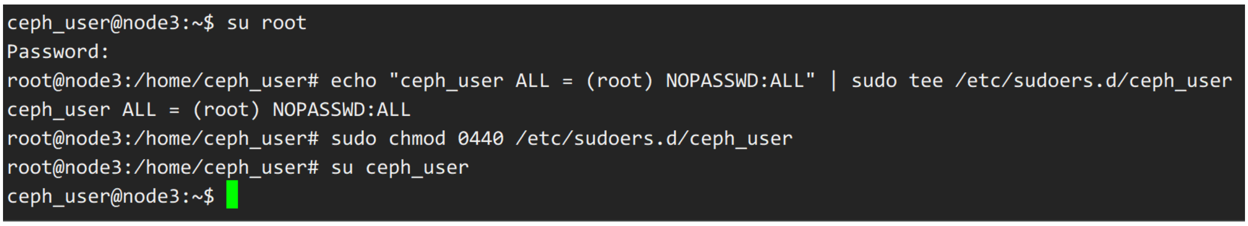
1. 在每个Ceph节点上创建一个新用户。



```
root@node1:/etc# useradd -d /home/ceph_user -m ceph_user
root@node1:/etc# passwd ceph_user
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@node1:/etc# su ceph_user
ceph_user@node1:/etc$ cd ~
ceph_user@node1:~$
```

```
ssh user@ceph-server
sudo useradd -d /home/{username} -m {username}
sudo passwd {username}
```

2. 对于添加到每个Ceph节点的新用户，请确保该用户具有 sudo权限。



```
ceph_user@node3:~$ su root
Password:
root@node3:/home/ceph_user# echo "ceph_user ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/ceph_user
ceph_user ALL = (root) NOPASSWD:ALL
root@node3:/home/ceph_user# sudo chmod 0440 /etc/sudoers.d/ceph_user
root@node3:/home/ceph_user# su ceph_user
ceph_user@node3:~$
```

```
echo "{username} ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/{username}
sudo chmod 0440 /etc/sudoers.d/{username}
```

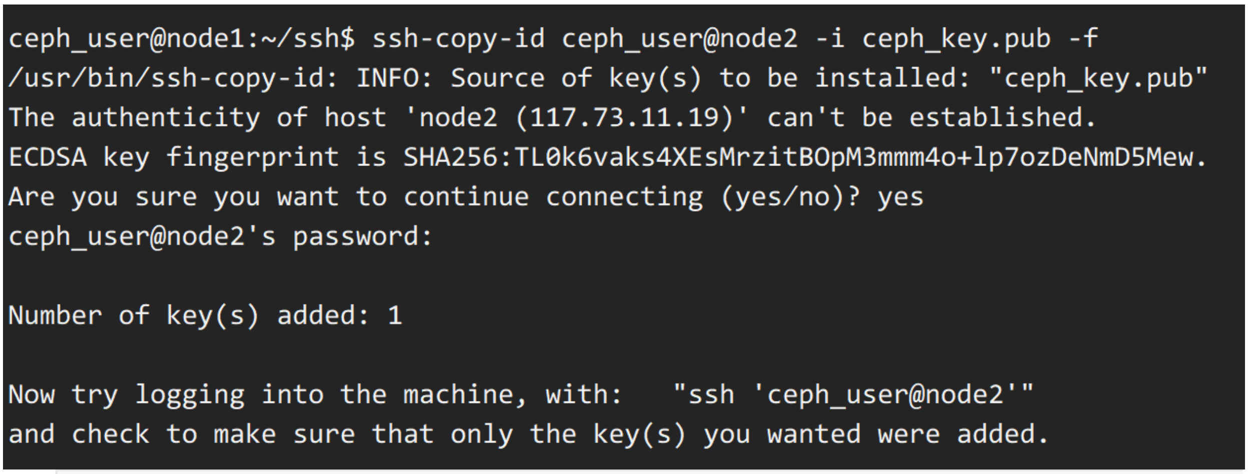
启用无密码SSH

由于ceph-deploy不会提示输入密码，因此必须在admin节点上生成SSH密钥，并将公钥分发给每个Ceph节点。ceph-deploy将尝试为初始监视器生成SSH密钥。

1. 生成SSH密钥，不要使用sudo或 root用户。将文件名和密码保留为空：

```
ssh-keygen

Generating public/private key pair.
Enter file in which to save the key (/ceph-admin/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /ceph-admin/.ssh/id_rsa.
Your public key has been saved in /ceph-admin/.ssh/id_rsa.pub.
```

2. 将密钥复制到每个Ceph节点，替换{username}为您创建Ceph Deploy用户的用户名。

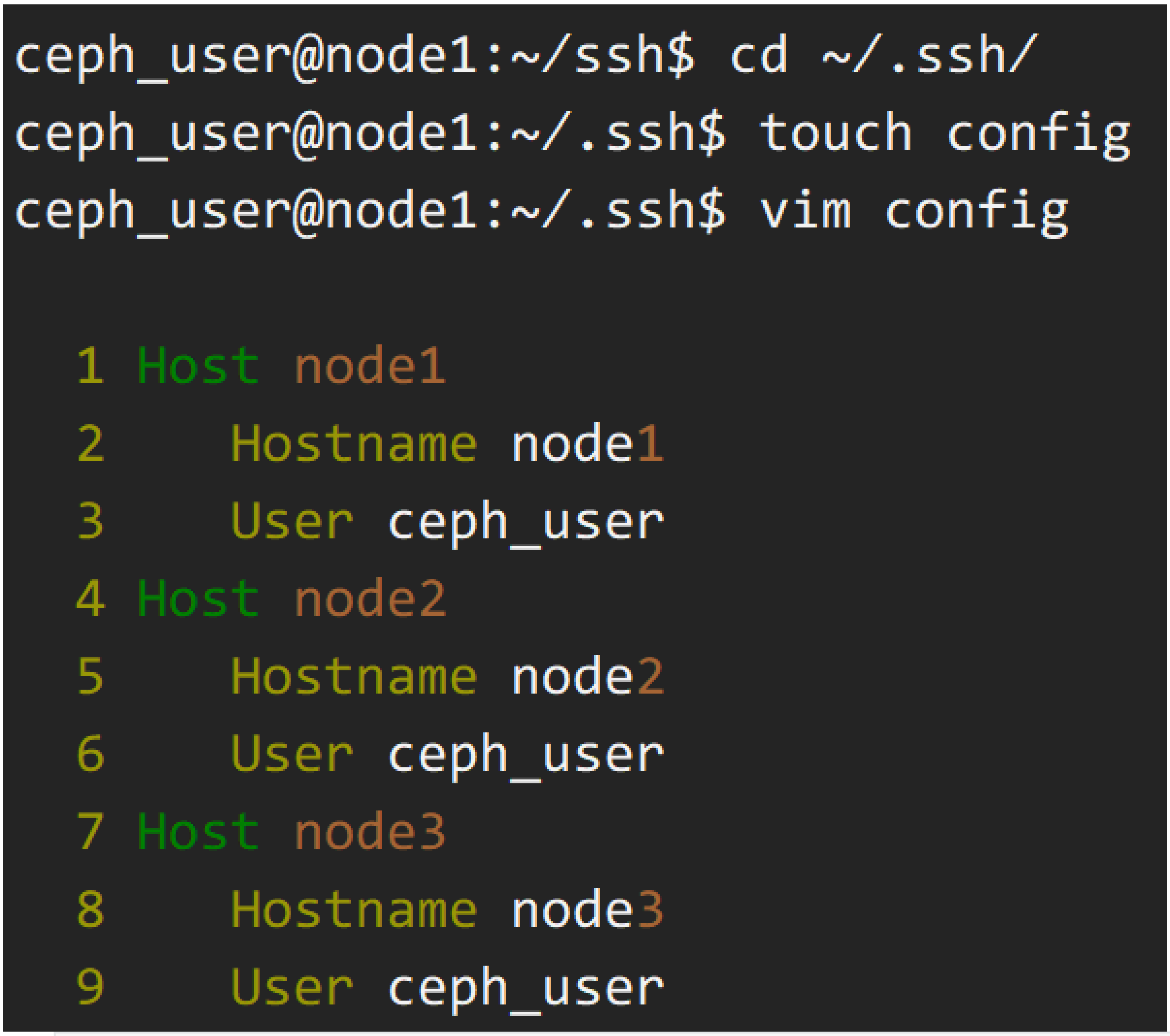


```
ceph_user@node1:~/ssh$ ssh-copy-id ceph_user@node2 -i ceph_key.pub -f
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "ceph_key.pub"
The authenticity of host 'node2 (117.73.11.19)' can't be established.
ECDSA key fingerprint is SHA256:TL0k6vaks4XEsMrzitBOpM3mmm4o+lp7ozDeNmD5Mew.
Are you sure you want to continue connecting (yes/no)? yes
ceph_user@node2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ceph_user@node2'"
and check to make sure that only the key(s) you wanted were added.
```

```
ssh-copy-id {username}@node1
ssh-copy-id {username}@node2
ssh-copy-id {username}@node3
```

3. （推荐）修改ceph-deploy管理节点的~/.ssh/config文件，以便ceph-deploy能够以您创建的用户身份登录Ceph节点，而无需在每次执行时指定。替换{username}为您创建Ceph Deploy用户的用户名。注意需要修改文件权限

```
chmod 644 ~/.ssh/config
```



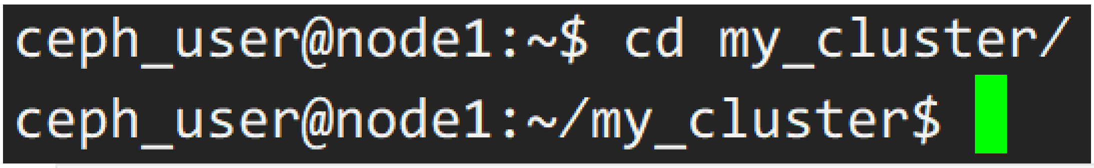
```
ceph_user@node1:~/ssh$ cd ~/.ssh/
ceph_user@node1:~/.ssh$ touch config
ceph_user@node1:~/.ssh$ vim config
```

```
Host node1
    Hostname node1
    User ceph_user
Host node2
    Hostname node2
    User ceph_user
Host node3
    Hostname node3
    User ceph_user
```

```
Host node1
  Hostname node1
  User {username}
Host node2
  Hostname node2
  User {username}
Host node3
  Hostname node3
  User {username}
```

存储群集快速入门

作为第一个练习，创建一个带有一个Ceph Monitor和三个Ceph OSDs守护进程的Ceph存储集群。一旦群集达到active + clean 状态，通过添加第四个Ceph OSD守护进程，元数据服务器和另外两个Ceph监视器来扩展它。为获得最佳结果，请在管理节点上创建一个目录，以维护为群集生成的配置文件和密钥。



```
ceph_user@node1:~$ cd my_cluster/
ceph_user@node1:~/my_cluster$
```

```
mkdir my-cluster
cd my-cluster
```

该ceph-deploy实用程序将文件输出到当前目录。执行时确保在ceph-deploy目录中。

> **注意：**
> 如果你登录为不同的用户时，不要用sudo运行ceph-deploy或root用户下运用它，因为它不会需要在远程主机上发出sudo命令。

重新开始（尝试过，未截图）

如果您遇到麻烦并且想要重新开始，请执行以下操作以清除Ceph软件包，并清除其所有数据和配置：

```
ceph-deploy purge {ceph-node} [{ceph-node}]
ceph-deploy purgedata {ceph-node} [{ceph-node}]
ceph-deploy forgetkeys
rm ceph.*
```

如果执行purge，则必须重新安装Ceph。最后一个rm 命令删除在先前安装期间由本地ceph-deploy写出的所有文件。

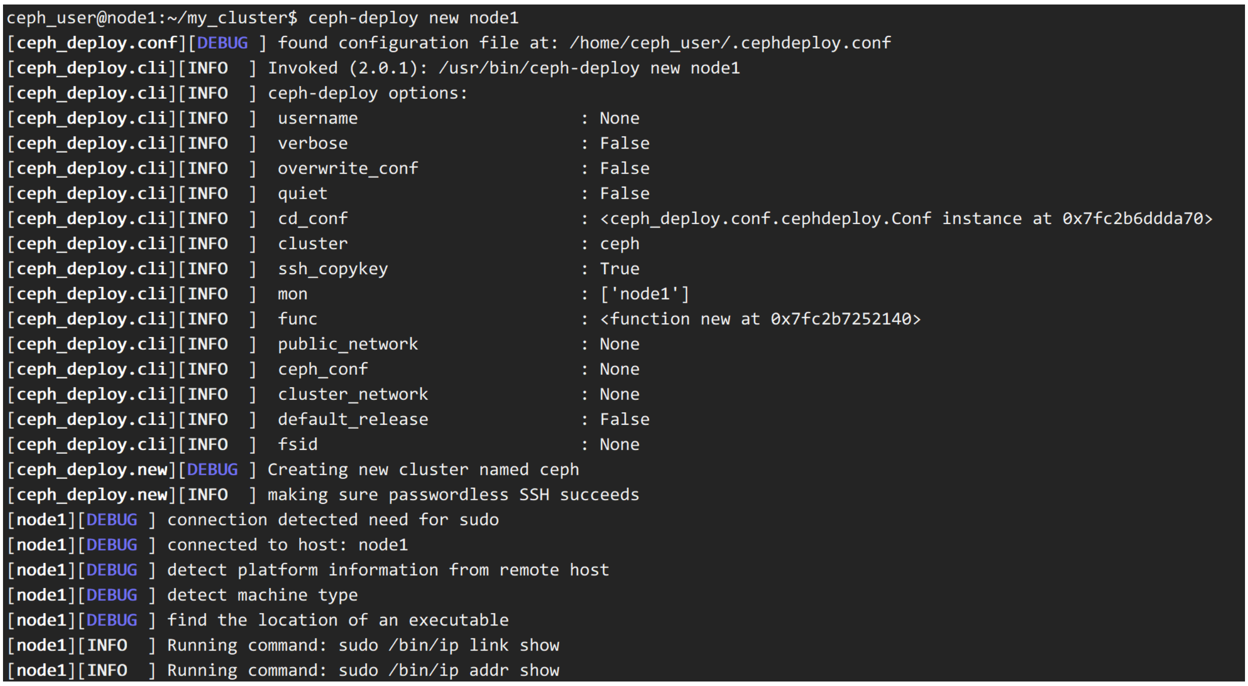
创建一个集群

在管理节点上保存配置详细信息的目录中，使用执行以下步骤ceph-deploy。

1. 创建群集。

```
ceph-deploy new {initial-monitor-node(s)}
```

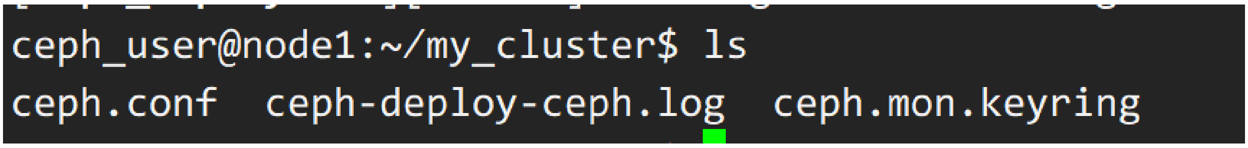
将节点指定为hostname，fqdn或hostname：fqdn。例如：



```
ceph_user@node1:~/my_cluster$ ceph-deploy new node1
[ceph_deploy.conf][DEBUG ] found configuration file at: /home/ceph_user/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (2.0.1): /usr/bin/ceph-deploy new node1
[ceph_deploy.cli][INFO  ] ceph-deploy options:
[ceph_deploy.cli][INFO  ]  username                      : None
[ceph_deploy.cli][INFO  ]  verbose                       : False
[ceph_deploy.cli][INFO  ]  overwrite_conf                : False
[ceph_deploy.cli][INFO  ]  quiet                         : False
[ceph_deploy.cli][INFO  ]  cd_conf                       : <ceph_deploy.conf.cephdeploy.Conf instance at 0x7fc2b6ddda70>
[ceph_deploy.cli][INFO  ]  cluster                       : ceph
[ceph_deploy.cli][INFO  ]  ssh_copykey                   : True
[ceph_deploy.cli][INFO  ]  mon                           : ['node1']
[ceph_deploy.cli][INFO  ]  func                          : <function new at 0x7fc2b7252140>
[ceph_deploy.cli][INFO  ]  public_network                : None
[ceph_deploy.cli][INFO  ]  ceph_conf                     : None
[ceph_deploy.cli][INFO  ]  cluster_network               : None
[ceph_deploy.cli][INFO  ]  default_release               : False
[ceph_deploy.cli][INFO  ]  fsid                          : None
[ceph_deploy.new][DEBUG ] Creating new cluster named ceph
[ceph_deploy.new][INFO  ] making sure passwordless SSH succeeds
[node1][DEBUG ] connection detected need for sudo
[node1][DEBUG ] connected to host: node1
[node1][DEBUG ] detect platform information from remote host
[node1][DEBUG ] detect machine type
[node1][DEBUG ] find the location of an executable
[node1][INFO  ] Running command: sudo /bin/ip link show
[node1][INFO  ] Running command: sudo /bin/ip addr show
```

```
ceph-deploy new node1
```

用ls和cat检查ceph-deploy当前目录中的输出。您应该看到Ceph配置文件（ceph.conf），监视器密钥密钥环（ceph.mon.keyring）和新集群的日志文件。有关其他详细信息，请参阅ceph-deploy new -h。



```
ceph_user@node1:~/my_cluster$ ls
ceph.conf  ceph-deploy-ceph.log  ceph.mon.keyring
```

2. 如果您有多个网络接口，请在Ceph配置文件[global]的部分下添加 public network 设置。有关详细信息，请参阅网络配置参考。

很关键！！！！！！！！ 需要配置全局的/etc/ceph/ceph.conf，需区分内网和外网。

```
public network = {ip-address}/{bits}
```

例如：

```
public network = 10.1.2.0/24
```

在10.1.2.0/24（或10.1.2.0/255.255.255.0）网络中使用IP。

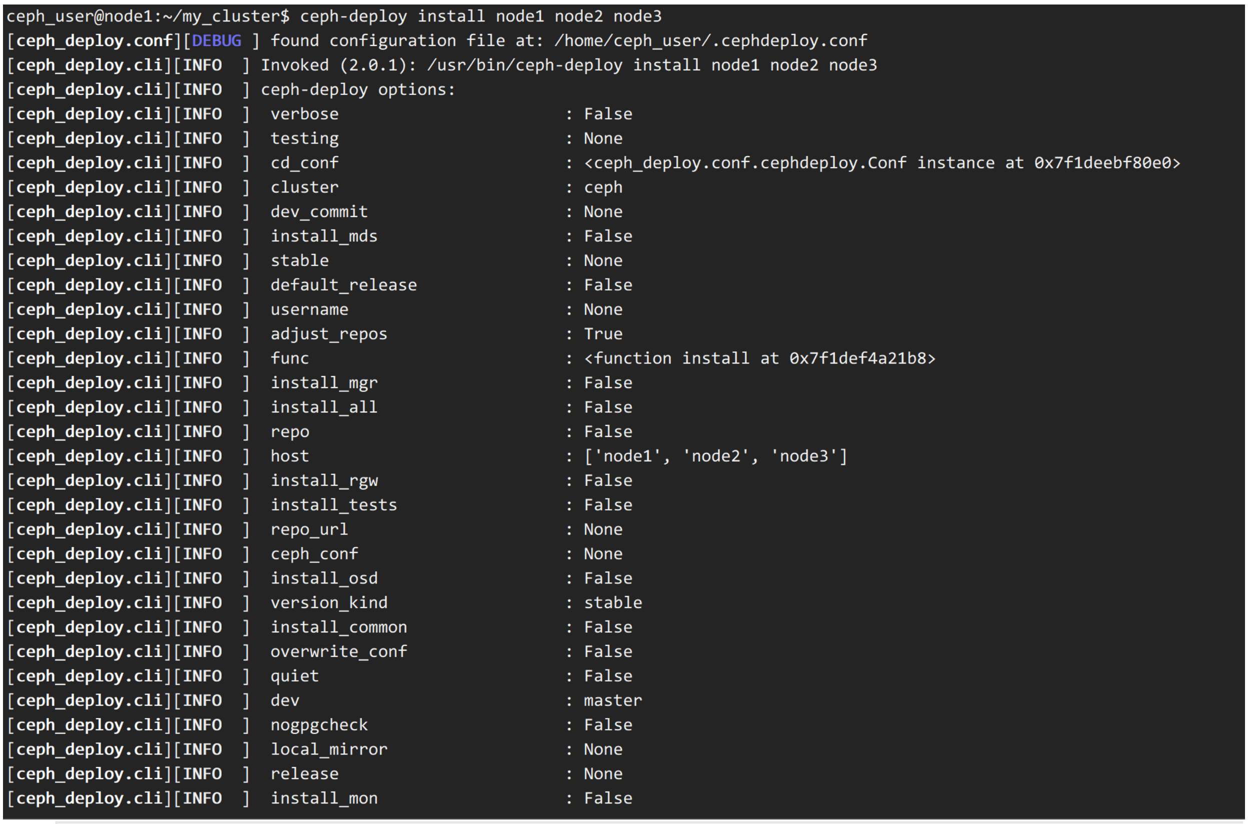
3. 如果要在IPv6环境中部署，请ceph.conf在本地目录中添加以下内容：（不需要）

```
echo ms bind ipv6 = true >> ceph.conf
```

4. 安装Ceph包：

```
ceph-deploy install {ceph-node} [...]
```

例如：

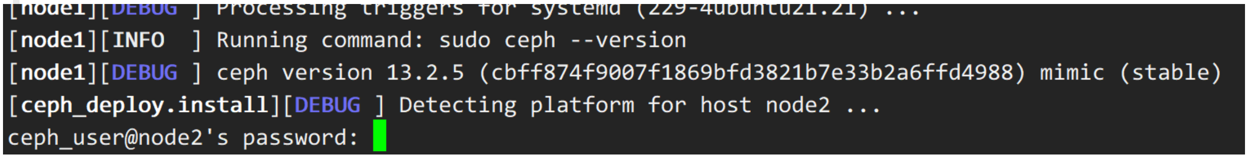


```
ceph_user@node1:~/my_cluster$ ceph-deploy install node1 node2 node3
[ceph_deploy.conf][DEBUG ] found configuration file at: /home/ceph_user/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (2.0.1): /usr/bin/ceph-deploy install node1 node2 node3
[ceph_deploy.cli][INFO  ] ceph-deploy options:
[ceph_deploy.cli][INFO  ]  verbose                       : False
[ceph_deploy.cli][INFO  ]  testing                       : None
[ceph_deploy.cli][INFO  ]  cd_conf                       : <ceph_deploy.conf.cephdeploy.Conf instance at 0x7f1deebf80e0>
[ceph_deploy.cli][INFO  ]  cluster                       : ceph
[ceph_deploy.cli][INFO  ]  dev_commit                    : None
[ceph_deploy.cli][INFO  ]  install_mds                   : False
[ceph_deploy.cli][INFO  ]  stable                        : None
[ceph_deploy.cli][INFO  ]  default_release               : False
[ceph_deploy.cli][INFO  ]  username                      : None
[ceph_deploy.cli][INFO  ]  adjust_repos                  : True
[ceph_deploy.cli][INFO  ]  func                          : <function install at 0x7f1def4a21b8>
[ceph_deploy.cli][INFO  ]  install_mgr                   : False
[ceph_deploy.cli][INFO  ]  install_all                   : False
[ceph_deploy.cli][INFO  ]  repo                          : False
[ceph_deploy.cli][INFO  ]  host                          : ['node1', 'node2', 'node3']
[ceph_deploy.cli][INFO  ]  install_rgw                   : False
[ceph_deploy.cli][INFO  ]  install_tests                 : False
[ceph_deploy.cli][INFO  ]  repo_url                      : None
[ceph_deploy.cli][INFO  ]  ceph_conf                     : None
[ceph_deploy.cli][INFO  ]  install_osd                   : False
[ceph_deploy.cli][INFO  ]  version_kind                  : stable
[ceph_deploy.cli][INFO  ]  install_common                : False
[ceph_deploy.cli][INFO  ]  overwrite_conf                : False
[ceph_deploy.cli][INFO  ]  quiet                         : False
[ceph_deploy.cli][INFO  ]  dev                           : master
[ceph_deploy.cli][INFO  ]  nogpgcheck                    : False
[ceph_deploy.cli][INFO  ]  local_mirror                  : None
[ceph_deploy.cli][INFO  ]  release                       : None
[ceph_deploy.cli][INFO  ]  install_mon                   : False
```

```
ceph-deploy install node1 node2 node3
```

该ceph-deploy实用程序将在每个节点上安装Ceph。

> 注意：弄完ssh需要reboot（reboot后仍旧需要输入密码）可能不需要重启，这部分可能之前操作有误



```
[node1][DEBUG ] Processing triggers for systemd (229-4ubuntu21.21) ...
[node1][INFO  ] Running command: sudo ceph --version
[node1][DEBUG ] ceph version 13.2.5 (cbff874f9007f1869bfd3821b7e33b2a6ffd4988) mimic (stable)
[ceph_deploy.install][DEBUG ] Detecting platform for host node2 ...
ceph_user@node2's password: 
```

5. 部署初始监视器并收集密钥：

```
ceph-deploy mon create-initial
```

完成此过程后，您的本地目录应具有以下密钥环：

- ceph.client.admin.keyring
- ceph.bootstrap-mgr.keyring
- ceph.bootstrap-osd.keyring
- ceph.bootstrap-mds.keyring
- ceph.bootstrap-rgw.keyring
- ceph.bootstrap-rbd.keyring
- ceph.bootstrap-rbd-mirror.keyring

可采用以下命令查看密钥

```
ceph-deploy gatherkeys node1
```

> **注意**
> 如果此过程失败并显示类似于“无法找到/etc/ceph/ceph.client.admin.keyring”的消息，请确保ceph.conf中为监控节点列出的IP是公共IP，而不是私有IP 。

6. 使用ceph-deploy配置文件和管理密钥复制到您的管理节点和你的Ceph的节点，以便每次执行命令ceph CLI时无需指定监视地址和 ceph.client.admin.keyring。

```
ceph-deploy admin {ceph-node(s)}
```

例如：

```
ceph-deploy admin node1 node2 node3
```

注意此处分配完密钥时，需要手动设置权限。

在所有节点运行下面的命令，改变密钥文件权限。

```
sudo chmod 644 /etc/ceph/ceph.client.admin.keyring
```

7. 部署管理器守护程序。（仅适用于luminous+ builds）：

```
ceph-deploy mgr create node1 *Required only for luminous+ builds, i.e >= 12.x builds*
```

8. 添加三个OSDs。出于这些说明的目的，我们假设您在每个节点中都有一个未使用的磁盘/dev/vdb。 *确保设备当前未使用且不包含任何重要数据。*

```
ceph-deploy osd create --data {device} {ceph-node}
```

例如：

```
ceph-deploy osd create --data /dev/vdb node1
```

```
ceph-deploy osd create --data /dev/vdb node2
ceph-deploy osd create --data /dev/vdb node3
```

> **注意**
> 如果要在LVM卷上创建OSD，则参数 --data *必须*是volume_group/lv_name，而不是卷的块设备的路径。

9. 检查群集的运行状况。

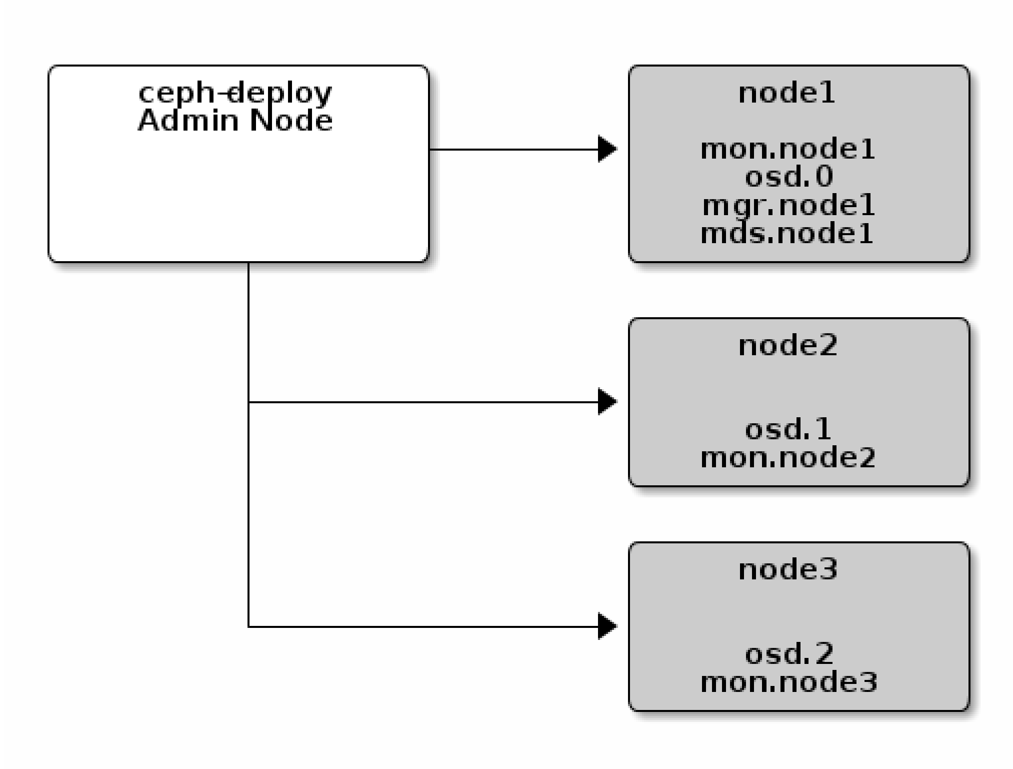
```
ssh node1 sudo ceph health
```

您的群集应该报告HEALTH_OK。您可以使用以下命令查看更完整的群集状态：

```
ssh node1 sudo ceph -s
```

扩展您的群集

启动并运行基本群集后，下一步是展开群集。添加Ceph元数据服务器node1。然后添加Ceph Monitor和Ceph Manager node2，node3以提高可靠性和可用性。



添加元数据服务器

要使用CephFS，您至少需要一个元数据服务器。执行以下操作以创建元数据服务器：

```
ceph-deploy mds create {ceph-node}
```

例如：

```
ceph-deploy mds create node1
```

添加监视器

Ceph存储集群需要至少运行一个Ceph Monitor和Ceph Manager。为了实现高可用性，Ceph存储集群通常运行多个Ceph监视器，因此单个Ceph监视器的故障不会导致Ceph存储

集群崩溃。Ceph使用Paxos算法，该算法需要大多数监视器（即大于*N／2*，其中*N*是监视器的数量）才能形成仲裁集。虽然这不是必需的，但监视器的数量往往更好。

将两个Ceph监视器添加到您的群集：

```
ceph-deploy mon add {ceph-nodes}
```

例如：

```
ceph-deploy mon add node2 node3
```

一旦你添加了新的Ceph监视器，Ceph将开始同步监视器并形成一个仲裁集。您可以通过执行以下操作来检查仲裁状态：

```
ceph quorum_status --format json-pretty
```

> **小技巧：**
> 当您使用多个监视器运行Ceph时，您应该在每个监视器主机上安装和配置NTP。确保监视器是NTP成员。

添加管理员

Ceph Manager守护进程以活动/备用模式运行。部署其他管理器守护程序可确保在一个守护程序或主机发生故障时，另一个守护程序或主机可以在不中断服务的情况下接管。

要部署其他管理器守护程序：

```
ceph-deploy mgr create node2 node3
```

您应该在以下输出中看到备用管理器：

```
ssh node1 sudo ceph -s
```

添加RGW实例

要使用Ceph的Ceph对象网关组件，必须部署RGW实例。执行以下命令以创建RGW的新实例：

```
ceph-deploy rgw create {gateway-node}
```

例如：

```
ceph-deploy rgw create node1
```

默认情况下，RGW实例将监听端口7480.可以通过在运行RGW的节点上编辑ceph.conf来更改此设置，如下所示：

```
[client]
rgw frontends = civetweb port=80
```

要使用IPv6地址，请使用：

```
[client]
rgw frontends = civetweb port=[::]:80
```

存储/检索对象数据

要将对象数据存储在Ceph存储集群中，Ceph客户端必须：

1. 设置对象名称
2. 指定一个池

Ceph客户端检索最新的集群映射，CRUSH算法计算如何将对象映射到放置组，然后计算如何动态地将放置组分配给Ceph OSD守护进程。要查找对象位置，您只需要对象名称和池名称。例如：

```
ceph osd map {poolname} {object-name}
```

> **练习：找到一个对象**
> 作为练习，让我们创建一个对象。使用命令行上的rados put命令指定对象名称，包含某些对象数据的测试文件的路径和池名称。例如：

```
echo {Test-data} > testfile.txt
ceph osd pool create mytest 8
rados put {object-name} {file-path} --pool=mytest
rados put test-object-1 testfile.txt --pool=mytest
```

> 要验证Ceph存储集群是否存储了该对象，请执行以下命令：

```
rados -p mytest ls
```

> 现在，确定对象位置：

```
ceph osd map {pool-name} {object-name}
ceph osd map mytest test-object-1
```

> Ceph应该输出对象的位置。例如：

```
osdmap e537 pool 'mytest' (1) object 'test-object-1' -> pg 1.d1743484 (1.4) -> up [1,0] acting [1,0]
```

> 要删除测试对象，只需使用该rados rm命令将其删除即可。
> 例如：

```
rados rm test-object-1 --pool=mytest
```

> 要删除mytest池：

```
ceph osd pool rm mytest
```

> （出于安全原因，您需要根据提示提供其他参数;删除池会破坏数据。需要改参数）
> 随着集群的发展，对象位置可能会动态变化。Ceph动态重新平衡的一个好处是Ceph使您不必手动执行数据迁移或平衡。