

云计算导论实践--ceph 存储

一、环境配置

1.1 主机名解析设置

修改各服务器的/etc/hosts 文件，向其添加如下内容，使得各服务器间可以通过主机名相互识别，避免通信时以 ip 地址作为身份认证，较为麻烦。

```
192.168.0.218 node01
192.168.0.213 node02
192.168.0.44 node03

::1      localhost        localhost.localdomain  localhost6      localhost6.localdomain6
127.0.0.1 localhost        localhost.localdomain  localhost4      localhost4.localdomain4
127.0.0.1      node03      node03
192.168.0.218 node01
192.168.0.213 node02
192.168.0.44 node03
```

1.2 服务器购买

在华为云购买云服务器，因为考虑云服务器只用作实验，因此计费方式选取按需计费以减少资金消耗

开启私有云，设置网段为 172.16.0.0/16，子网网段为 172.16.249.0/24。依照实验材料设置三个云服务器的局域网 ip 地址。各服务器 ip、账密以及其他设置如下所示：

服务器一:Ip 地址:192.168.0.218、主机:node01、账密:root:admin@123

云服务器名称	node01	ID	6a0c27cd-2ac3-45c3-b25b-220458399022
镜像	CentOS 7.9 64bit	计费模式	按需计费
当前规格	通用入门型 t6.large.2 2vCPUs 4GiB	区域	华北-北京四

服务器二:Ip 地址:192.168.0.213、主机名:node02、账密:root:admin@123

云服务器名称	node02	ID	e86f7445-bf57-44dc-84a5-1250b95f80c1
镜像	CentOS 7.9 64bit	计费模式	按需计费
当前规格	t6.large.2 2vCPUs 4GiB	区域	华北-北京四

服务器三:Ip 地址:192.168.0.44、主机名:node03、账密:root:admin@123

云服务器名称	node03	ID	cdb7b65d-4ba6-4cbd-99e4-537fba2f695c
镜像	CentOS 7.9 64bit	计费模式	按需计费
当前规格	通用入门型 t6.large.2 2vCPUs 4GiB	区域	华北-北京四

1.3ssh 免密登录设置

修改各服务器的/etc/ssh/sshd_config 文件，将如下几个配置取消注释并修改相应选项。

```
AuthorizedKeysFile .ssh/authorized_keys
PermitRootLogin yes
PasswordAuthentication yes
```

```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no

AuthorizedKeysFile .ssh/authorized_keys
PermitRootLogin yes
PasswordAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes

# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials no
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
#GSSAPIEnableK5users no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
-- INSERT --
```

在各个服务器下执行 `ssh-keygen -t rsa` 生成 ssh 连接所需要的公私钥文件以及认证需要的 `authorized_keys` 文件。

```
[root@node03 ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Yw7+kojKHPY+ygqYwuVCMhDkRywgYecQnJ7Z5a9n9Q0 root@node03
The key's randomart image is:
+---[RSA 2048]---+
|O*+o
|+=. .
|o.=oo
|.+. .
|o. . o S
|=oo . = o E
|== o .o.o . o
|B =...o+ .
|*oo.o..
+---[SHA256]-----+
[root@node03 ~]#
```

将 node02 和 node03 服务器的 `id_rsa.pub` 文件内容复制到 node01 服务器的 `authorized_keys` 中，保存文件之后即可完成 node02 和 node03 免密登录 node01，同理完成 node02 和 node03 的免密登录操作。完成之后，node01 服务器的 `authorized_keys` 中应当含有 node2 和 node3 服务器 `id_rsa.pub` 的内容。

1.4 查看时区

在各个服务器上执行 `timedatectl` 查看各自所在时区，保证三个服务器所处同一个时区---Asia/Shanghai。

```
[root@k8s-master ~]# timedatectl
Local time: Mon 2025-06-02 01:32:41 CST
Universal time: Sun 2025-06-01 17:32:41 UTC
RTC time: Sun 2025-06-01 17:32:39
Time zone: Asia/Shanghai (CST, +0800)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
[root@k8s-master ~]#
```

```

[root@k8s-node ~]# timedatectl
Local time: Mon 2025-06-02 01:33:13 CST
Universal time: Sun 2025-06-01 17:33:13 UTC
RTC time: Sun 2025-06-01 17:33:13
Time zone: Asia/Shanghai (CST, +0800)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
[root@k8s-node ~]#

[root@node03 ~]# timedatectl
Local time: Mon 2025-06-02 01:34:16 CST
Universal time: Sun 2025-06-01 17:34:16 UTC
RTC time: Sun 2025-06-01 17:34:16
Time zone: Asia/Shanghai (CST, +0800)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
[root@node03 ~]#

```

1.5 关闭防火墙

为了避免之后的实验用到的端口被防火墙拦截导致实验没办法正常进行，我们实现关闭各自服务器的防火墙，之后依据实验进度再在华为云控制台修改安全组对于端口的过滤/开放规则。

```

root@node01:~/.ssh# systemctl stop ufw
root@node01:~/.ssh# systemctl status ufw
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Thu 2023-05-25 17:00:22 CST; 11s ago
     Docs: man:ufw(8)
   Process: 412 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SUCCESS)
   Process: 2075 ExecStop=/lib/ufw/ufw-init stop (code=exited, status=0/SUCCESS)
   Main PID: 412 (code=exited, status=0/SUCCESS)
    CPU: 2ms

May 25 16:32:32 node01 systemd[1]: Starting Uncomplicated firewall...
May 25 16:32:32 node01 systemd[1]: Finished Uncomplicated firewall.
May 25 17:00:22 node01 systemd[1]: Stopping Uncomplicated firewall...
May 25 17:00:22 node01 ufw-init[2075]: Skip stopping firewall: ufw (not enabled)
May 25 17:00:22 node01 systemd[1]: ufw.service: Deactivated successfully.
May 25 17:00:22 node01 systemd[1]: Stopped Uncomplicated firewall.
root@node01:~/.ssh#

```

二、配置 ceph

2.1 安装 ceph

在各服务器中执行 `aptinstallceph-v` 以安装 ceph，由于华为云的云服务器 apt 源默认使用国内源，因此我们无需进行换源操作，如果遇到无法正常下载 ceph 的情况，使用 `apt-getupdate` 更新源即可。

安装完成后尝试执行 `ceph-v` 来查看安装的 ceph 的版本，如果命令成功执行则表示 ceph 正常安装，可以看到三台机器的 ceph 版本统一，均是 17.2.5 版本。

```

root@node02:~# ceph -v
ceph version 17.2.5 (98318ae89f1a893a6ded3a640405cddb33e08757) quincy (stable)
root@node02:~#

```

2.2 node01 相关 mon 配置

2.2.1 生成 uuid 并创建 ceph.conf

使用 `uuidgen` 生成 uuid，将其作为 node01 在集群中的唯一标识。

```
root@116.63.143.111 x
root@node81:~# uuidgen
24956837-3602-4276-b7a6-616d4785bb54
root@node81:~#
```

编辑/etc/ceph/ceph.conf，其内容如下：

```
[global]
fsid=24956837-3602-4276-b7a6-616d4785bb54
moninitialmembers=node01
monhost=172.16.249.78
publicnetwork=172.16.249.0/24
clusternetwork=192.168.100.0/24
authclusterrequired=cephx
authservicerequired=cephx
authclientrequired=cephx
osdjournalsize=1024osdpooldefaultsize=3
osdpooldefaultminsize=2osdpooldefaultpgnum=16
osdpooldefaultpgpnum=16
osdcrushchooseleatype=1
```

2.2.2 创建 mon 令牌环

使用如下命令生成令牌环。紧跟在--create-keyring 之后的参数指定令牌环文件存放在/tmp/ceph.mon.keyring；--gen-key 参数为指定实体名生成新私钥；-n 参数指定要操作的实体名为 mod.；.--cap 指定在 mon 子系统下授予该密钥完整的超级用户权限。

```
root@node81:~# ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon, --cap mon 'allow *'
creating /tmp/ceph.mon.keyring
root@node81:~#
```

由于令牌环文件存放在 tmp 目录下，云服务器会在每次关闭机器后清理 tmp 目录下的文件，因此为了避免令牌环文件被清除，需要修改

/usr/lib/tmpfiles.d/tmp.conf 内容如下所示。

```
#Thisfileispartofsystemd.
#systemdisfreesoftware;youcanredistributeitand/ormodifyit
#underthetermsoftheGNULesserGeneralPublicLicenseaspublishedby#theFreeSoftwareFound
ation;eitherversion2.1oftheLicense,or
#(atyouroption)anylaterversion.
#Seetmpfiles.d(5)fordetails
#Cleartmpdirectoriesseparately,tomakethemeasiertooverride#D/tmp1777rootroot-
X/tmp/ceph.*
X/tmp/monmap
#q/var/tmp1777rootroot30d
```

2.2.3 创建管理令牌环

使用如下命令生成令牌环。紧跟在--create-keyring 之后的参数指定令牌环文件存放在/etc/ceph/ceph.client.admin.keyring; --gen-key 参数为指定实体名生成新私钥; -n 参数指定要操作的实体名为 client.admin; --cap 指定在 mon、osd、mds、mgr 子系统下授予该密钥完整的超级用户权限。

```
root@node01:~# 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 *'
creating /etc/ceph/ceph.client.admin.keyring
root@node01:~#
```

2.2.4 创建 osd 引导令牌环

使用如下命令生成令牌环。紧跟在--create-keyring 之后的参数指定令牌环文件存放在/var/lib/ceph/bootstrap-osd/ceph.keyring; --gen-key 参数为指定实体名生成新私钥; -n 参数指定要操作的实体名为 client.bootstrap-osd; --cap 指定在 mon 子系统下授予用户引导 OSD 的权限, 在 mgr 子系统下读取权限。

```
root@node01:~# ceph-authtool --create-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring --gen-key -n client.bootstrap-osd --cap mon 'profile bootstrap' --cap mgr 'allow r'
creating /var/lib/ceph/bootstrap-osd/ceph.keyring
root@node01:~#
```

2.2.5 将 mon 令牌环添加给管理和 osd 引导令牌环中

```
root@node01:~# 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
root@node01:~# ceph-authtool /tmp/ceph.mon.keyring --import-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring
importing contents of /var/lib/ceph/bootstrap-osd/ceph.keyring into /tmp/ceph.mon.keyring
root@node01:~#
```

2.2.5 修改用户组

```
root@node01:~# chown ceph:ceph /tmp/ceph.mon.keyring
root@node01:~#
```

2.2.6 生成 monmap

```
root@node01:~# monmaptool --create --add node01 172.16.249.78 --fsid 24956837-3682-4276-b7a6-616d4785bb54 /tmp/monmap
monmaptool: monmap file /tmp/monmap
setting min_mon_release = octopus
monmaptool: set fsid to 24956837-3682-4276-b7a6-616d4785bb54
monmaptool: writing epoch 0 to /tmp/monmap (1 monitors)
root@node01:~#
```

2.2.7 创建 mon 目录

```
root@node01:~# sudo -u ceph mkdir /var/lib/ceph/mon/ceph-node01
root@node01:~# chown -R ceph /var/lib/ceph/mon/ceph-node01/
root@node01:~#
```

2.2.8 初始化 mon 节点守护进程

```
root@node01:~# sudo -u ceph ceph-mon --mkfs -i node01 --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring
root@node01:~#
```

```
root@node01:~# systemctl start ceph-mon@node01
root@node01:~# systemctl enable ceph-mon@node01
Created symlink /etc/systemd/system/ceph-aon.target.wants/ceph-mon@node01.service -- /lib/systemd/systemd/ceph-mon@node01.service.
root@node01:~# systemctl status ceph-mon@node01
● ceph-mon@node01.service - Ceph cluster monitor daemon
   Loaded: loaded (/lib/systemd/system/ceph-mon@node01.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-05-29 11:12:13 CST; 14s ago
     Main PID: 1649 (ceph-mon)
       Tasks: 24
      Memory: 12.3M
         CPU: 103ms
    CGroup: /system.slice/ceph-mon@node01.service
            └─1649 /usr/bin/ceph-mon -f --cluster ceph --id node01 --setuser ceph --setgroup ceph

May 29 11:12:13 node01 systemd[1]: Started Ceph cluster monitor daemon.
root@node01:~#
```


2.2.9 将/etc/ceph/下所有文件拷贝至其他节点

```
root@node01:~# scp /etc/ceph/* node02:/etc/ceph/
ceph.client.admin.keyring                                100M 151 312.5KB/s 00:00
ceph.conf                                                 100M 448 1.3MB/s 00:00
rbdmap                                                    100M  92 205.3KB/s 00:00
root@node01:~# scp /etc/ceph/* node03:/etc/ceph/
The authenticity of host 'node03 (172.16.249.99)' can't be established.
ED25519 key fingerprint is SHA256:M7faC0IEMc2wyOanapHnIlyQ22g2NanFVChusE(HDfo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'node03' (ED25519) to the list of known hosts.
ceph.client.admin.keyring                                100M 151 323.1KB/s 00:00
ceph.conf                                                 100M 448 1.0MB/s 00:00
rbdmap                                                    100M  92 275.2KB/s 00:00
root@node01:~#
```

2.3 node02 相关 mon 配置

2.3.1 创建 mon 目录

```
root@node02:~# sudo -u ceph mkdir /var/lib/ceph/mon/ceph-node02
root@node02:~#
```

2.3.2 在临时目录获取监视器密钥环

```
root@node02:~# ceph auth get mon. -o /tmp/ceph.mon.keyring
exported keyring for mon.
root@node02:~# ls /tmp
ceph.mon.keyring
hstoredata_root
snap-private-tmp
systemd-private-41102fd95e8143739131fa43592d9bc8-chrony.service-kkQztz
systemd-private-41102fd95e8143739131fa43592d9bc8-ModemManager.service-UXicUe
systemd-private-41102fd95e8143739131fa43592d9bc8-systemd-logind.service-CypDMB
systemd-private-41102fd95e8143739131fa43592d9bc8-systemd-resolved.service-LoM7jU
wrapper-794-1-in
wrapper-794-1-out
root@node02:~#
```

2.3.3 获取监视器运行图

```
root@node02:~# ceph mon getmap -o /tmp/ceph.mon.map
got monmap epoch 2
root@node02:~# ls /tmp
ceph.mon.keyring  ayatond-private-41102fd95e8143739131fa43592d9bc8-chrony.service-kkQztz  wrapper-794-1-in
ceph.mon.map      ayatond-private-41102fd95e8143739131fa43592d9bc8-ModemManager.service-UXicUe  wrapper-794-1-out
hstoredata_root  ayatond-private-41102fd95e8143739131fa43592d9bc8-ayatond-logind.service-CypDMB
snap-private-tmp  ayatond-private-41102fd95e8143739131fa43592d9bc8-ayatond-resolved.service-LoM7jU
root@node02:~#
```

2.3.4 修改 ceph.mon.keyring 属主和属组为 ceph

```
root@node02:~# chown ceph.ceph /tmp/ceph.mon.keyring
root@node02:~# ls -al /tmp/ceph.mon.keyring
-rw-r--r-- 1 ceph ceph 77 May 29 11:20 /tmp/ceph.mon.keyring
root@node02:~#
```

2.3.5 初始化 mon

```
root@node02:~# sudo -u ceph ceph-mon --mkfs -i node02 --monmap /tmp/ceph.mon.map --keyring /tmp/ceph.mon.keyring
root@node02:~#
```

2.3.6 启动服务

```

root@node02:~# systemctl start ceph-mon@node02
root@node02:~# systemctl enable ceph-mon@node02
Created symlink /etc/systemd/system.target.wants/ceph-mon@node02.service -- /lib/systemd/system/ceph-mon@.service.
root@node02:~# systemctl status ceph-mon@node02
● ceph-mon@node02.service - Ceph cluster monitor daemon
   Loaded: loaded (/lib/systemd/system/ceph-mon@.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-05-29 11:24:06 CST; 13s ago
     Main PID: 1729 (ceph-mon)
        Task: 24
     Memory: 12.9M
        CPU: 115ms
     CGroup: /system.slice/ceph-mon@node02.service
             └─1729 /usr/bin/ceph-mon -f --cluster ceph --id node02 --setuser ceph --setgroup ceph

May 29 11:24:06 node02 systemd[1]: Started Ceph cluster monitor daemon.
root@node02:~#

```

2.4 node03 相关 mon 配置

2.4.1 创建 mon 目录

```

root@node03:~# sudo -u ceph mkdir /var/lib/ceph/mon/ceph-node82
root@node03:~#

```

2.4.2 在临时目录获取监视器密钥环

```

root@node03:~# ceph auth get mon. -o /tmp/ceph.mon.keyring
exported keyring for mon.
root@node03:~#

```

2.4.3 获取监视器运行图

```

root@node03:~# ceph mon getmap -o /tmp/ceph.mon.map
got monmap epoch 3
root@node03:~#

```

2.4.4 初始化 mon

```

root@node03:~# sudo -u ceph ceph-mon --mkfs -i node03 --monmap /tmp/ceph.mon.map --keyring /tmp/ceph.mon.keyring
root@node03:~#

```

2.4.5 启动服务

```

root@node03:~# systemctl start ceph-mon@node03
root@node03:~# systemctl enable ceph-mon@node03
Created symlink /etc/systemd/system.target.wants/ceph-mon@node03.service -- /lib/systemd/system/ceph-mon@.service.
root@node03:~# systemctl status ceph-mon@node03
● ceph-mon@node03.service - Ceph cluster monitor daemon
   Loaded: loaded (/lib/systemd/system/ceph-mon@.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-05-29 11:30:07 CST; 11s ago
     Main PID: 1791 (ceph-mon)
        Task: 24
     Memory: 14.0M
        CPU: 122ms
     CGroup: /system.slice/ceph-mon@node03.service
             └─1791 /usr/bin/ceph-mon -f --cluster ceph --id node03 --setuser ceph --setgroup ceph

May 29 11:30:07 node03 systemd[1]: Started Ceph cluster monitor daemon.
root@node03:~#

```

2.5 node01 OSD 设置

2.5.1 导入引导密钥环

```

root@node01:/etc/ceph# ceph auth get client.bootstrap-osd -o /var/lib/ceph/bootstrap-osd/ceph.keyring
exported keyring for client.bootstrap-osd
root@node01:/etc/ceph# ceph auth get client.bootstrap-osd -o /etc/ceph/ceph.client.bootstrap-osd.keyring
exported keyring for client.bootstrap-osd
root@node01:/etc/ceph#

```

2.5.2 消除分区信息

```
root@node01:/etc/ceph# dd if=/dev/zero of=/dev/vdb bs=512K count=1
1+8 records in
1+8 records out
524288 bytes (524 kB, 512 KiB) copied, 0.000855138 s, 613 MB/s
root@node01:/etc/ceph#
```

2.5.3 创建 osd

在消除分区信息后，创建 osd

```
--> ceph-volume lvm activate successful for osd ID: 0
--> ceph-volume lvm create successful for: /dev/vdb
```

2.5.4 列出与 Ceph 关联的逻辑卷和设备

查看创建的 osd 的 fsid 以及 id 便于后续激活。

```
root@node01:/etc/ceph# sudo ceph-volume lvm list

===== osd.0 =====

[block]          /dev/ceph-aac098f1-bela-457a-9dab-237c9b648d7b/osd-block-34aaef7-7226-4aa0-0600-18cf0a20703

block device      /dev/ceph-aac098f1-bela-457a-9dab-237c9b648d7b/osd-block-34aaef7-7226-4aa0-0600-18cf0a20701
block uuid        WJZCgP-d18Z-vk2Y-18Bz-aWE2-K0q2-h2aH0v
cephx keybox secret
cluster fsid      24966837-3682-4276-b7a6-616d4785bb54
cluster name      ceph
crush device class
oncrushpin        0
bad fsid          34aaef7-7226-4aa0-0600-18cf0a20701
osd id            0
osdepec affinity
type              block
vdo               0
devices           /dev/vdb

root@node01:/etc/ceph#
```

2.5.5 准备就绪的 osd 进行激活

使用上一步骤中得到的 id 和 fsid 进行激活。

```
root@node01:/etc/ceph# sudo ceph-volume lvm activate 0 34aaef7-7226-4aa0-0600-18cf0a20703
Running command: /usr/bin/ceph -R ceph:/var/lib/ceph/osd/ceph-0
Running command: /usr/bin/cneph-bluestore-tool --cluster-ceph priu-osd-dir --dev /dev/dm-0-aac098f1-bela-457a-9dab-237c9b648d7b/osd-block-34aaef7-7226-4aa0-0600-18cf0a20703 --path /var/lib/ceph/osd/ceph-0 --no-mon-config
Running command: /usr/bin/lv -anf /dev/ceph-aac098f1-bela-457a-9dab-237c9b648d7b/osd-block-34aaef7-7226-4aa0-0600-18cf0a20703 /var/lib/ceph/osd/ceph-0/block
Running command: /usr/bin/chown -h ceph:ceph /var/lib/ceph/osd/ceph-0/block
Running command: /usr/bin/chown -R ceph:ceph /dev/dm-0
Running command: /usr/bin/ceph -R ceph:/var/lib/ceph/osd/ceph-0
Running command: /usr/bin/ceph-volume lvm enable ceph-volume lvm-0-34aaef7-7226-4aa0-0600-18cf0a20703
Running command: /usr/bin/ceph-volume lvm enable --cuntine ceph-osd0
Running command: /usr/bin/ceph-volume start ceph-osd0
--> ceph-volume lvm activate successful for osd ID: 0
```

2.5.6 启动服务并修改相应文件

启动服务后，会在/etc/systemd/system/ceph-osd.target.wants/目录下生成相应 osd 的设置文件。

```
root@node01:/etc/ceph# systemctl enable ceph-osd0
root@node01:/etc/ceph# systemctl start ceph-osd0
root@node01:/etc/ceph# systemctl status ceph-osd0
● ceph-osd0.service - Ceph object storage daemon osd.0
   Loaded: loaded (/lib/systemd/system/ceph-osd0.service; vendor preset: enabled)
   Active: active (running) since Mon 2023-05-29 22:41:47 CST; 1min 18s ago
     Main PID: 12002 (ceph-osd)
        Tasks: 82
       Memory: 63.7M
          CPU: 717ms
     CGroup: /system.slice/system-ephel2d.service/ceph-osd0.service
             └─12002 /usr/bin/ceph-osd -f --cluster ceph --id 0 --setuid ceph --setgroup ceph

May 29 22:41:47 node01 systemd[1]: Starting Ceph object storage daemon osd.0...
May 29 22:41:47 node01 systemd[1]: Started Ceph object storage daemon osd.0.
May 29 22:41:48 node01 ceph-osd[12002]: 2023-05-29T22:41:48.553-0800 742664da48c8 -1 Falling back to public interface
May 29 22:41:58 node01 ceph-osd[12002]: 2023-05-29T22:41:58.593-0800 7498646a45e0 -1 osd.0 @ log_id_monitors trace
May 29 22:41:54 node01 ceph-osd[12002]: 2023-05-29T22:41:54.756-0800 7498646a45e0 -1 osd.0 @ waiting for initio! osd0p
May 29 22:41:54 node01 ceph-osd[12002]: 2023-05-29T22:41:54.744-0800 7498646a45e0 -1 osd.0 @ set_nuno_affinity unable to identify public interface
Linux 1-10110 ceph01
```

这里注释掉生成的配置文件中的 StartLimitInterval 选项，该选项限制 osd 两次启动

的时间间隔不能小于 30min，不利于配置环境时环境报错的情况。

```
StartLimitBurst=3
#StartLimitInterval=30min
TasksMax=infinity
```

当服务启动发生错误导致 osd 结点处于 down 状态时，第一时间应当查看 /var/log/ceph/{osd-isd}.log 文件进行错误排查，日志文件中错误内容具体如下：

```
2023-05-29T11:48:42.268+0800 Tfd985bd5c8 -1 unable to find any IPv4 addresses in networks '192.168.100.0/24' interfaces ''
2023-05-29T11:48:42.268+0800 Tfd985bd5c8 -1 Failed to pick cluster address.
```

可以发现错误的原因在于 osd 结点没办法寻找到隶属于 192.168.100.0/24 网段的 ip 地址，这是由于我们在 ceph 的配置文件/etc/ceph/ceph.conf 中填写了 cluster-ip 字段，并将该字段的值填写为 192.168.100.0/24，使得 osd 结点在启动过程中会寻找该网段的 ip 地址，而我们机器并未设置某一个网卡的 ip 地址隶属于该网段从而报错。

注释掉配置文件的 cluster-ip，然后安装 ceph-deploy，使用 ceph-deploy--overwrite-conf config push node1 将配置文件重新下发，然后重新启动 osd 结点即可。

```
# ceph-adm@2:~$ ceph object storage daemon nad.2
Loaded: loaded (/lib/systemd/system/ceph-adm2.service; enabled; vendor preset: enabled)
Active: active (running) since Mon 2023-05-29 22:13:13 CST; 22min ago
Main PID: 6049 (ceph-adm)
Tasks: 62
Memory: 78.8M
CPU: 3.156s
CGroup: /system.slice/system-ceph.slice/ceph-adm2.service
          └─6065 /usr/bin/ceph-adm -f --cluster ceph --ad 2 --set-addr ceph --set-group ceph

May 20 22:13:13 node01 systemd[1]: Starting Ceph object storage daemon nad.2...
May 20 22:13:13 node01 systemd[1]: Started Ceph object storage daemon nad.2.
May 10 22:13:14 node01 ceph-adm[6049]: 2023-05-20T22:11:14.112-0800 7f8ab132f5c2 -1 Felling back to public interface
May 19 22:13:16 node01 ceph-adm[6049]: 2023-05-20T22:11:10.284-0800 7f8ab132f5c2 -1 oad.2 @ lag to monitor trun
May 29 22:13:26 node01 ceph-adm[6049]: 2023-05-29T22:13:26.744-0800 7f8ab132f5c2 -1 oad.2 @ waiting for initial oadnap
May 29 22:13:20 node01 ceph-adm[6049]: 2023-05-29T22:17:29.740-0800 7f8ab132f5c2 -1 oad.2 @ set-addr ceph --set-group ceph
linna 1-16/16 (CMD)
```

值得一提的是，osd 结点的 id 号是逐步递增的，即使前面部署的结点因为某些问题处于 down 状态，因此如果希望可以删除掉存在问题的结点然后重新部署。

2.6node02OSD 设置

2.6.1 导入引导密钥环

```
root@node02:~# ceph auth get client.bootstrap-osd -o /var/lib/ceph/bootstrap-osd/ceph.keyring
exported keyring for client.bootstrap-osd
root@node02:~# ceph auth get client.bootstrap-osd -o /etc/ceph/ceph.client.bootstrap-osd.keyring
exported keyring for client.bootstrap-osd
root@node02:~#
```

2.6.2 消除分区信息

```
root@node02:~# dd if=/dev/zero of=/dev/vdb bs=512K count=1
1+0 records in
1+0 records out
524288 bytes (524 kB, 512 KiB) copied, 0.00699124 s, 75.0 MB/s
```

2.6.3 创建 osd

```
→ ceph-volume lvm activate successful for osd ID: 1
→ ceph-volume lvm create successful for: /dev/vdb
root@node02:~#
```

2.6.4 列出与 Ceph 关联的逻辑卷和设备

```

root@node02:~# sudo ceph-volume lvm list

===== oid.1 =====

[block1] /dev/ceph-8201f731-5836-4120-8fc6-68e78e6c87a0/osd-block-dd466871-cn12-40ee-6505-748e75d07117

    block device      /dev/ceph-d2a1f731-5836-4120-6fc6-60a78a6c8740/osd-block-dd400873-cn12-40ee-6095-748a75d0711f
    block uuid        aN0NT0-1MV4-71tb-VXzF-vfTq-TabM-157tyt
    cephx lockbox sneret
    cluster Taid       24956837-3602-4276-b7a6-610d4785bb54
    cluster name       ceph
    crush device class
    encrypted          B
    oad fsid            dd466871-cn12-40ee-b595-748e75d671ff
    oad id              1
    pedspec affinity
    type               block
    vde                 B
    devices             /dev/vdb

root@node02:~#

```

2.6.5 准备就绪的 osd 进行激活

```

root@node01:~# sudo ceph-volume lvs activate 2 501d58c-fes3-4786-be83-178w/001c53
Running command: /usr/bin/chopt -H cephadm /usr/lib/ceph/ceph-2
Running command: /usr/bin/ceph-bluestore-test --cluster=ceph prime-cep-dir --dvv /dev/ceph-7a0c0e9f-f5bb-4b3d-5016-b18a734f24en/osd-block-5e6d8dec-fa
e3-4786-0d13-570e/001c53 --path /var/lib/ceph/osd/ceph-2 --oo-acn-cootig
Running command: /usr/bin/ln -s /dev/ceph-ls1c8aer-f08a-1b8e-3610-11ba72A42Aen/osd-block-50ba8bae-fec3-4786-bd83-378a/5351c53 /var/lib/ceph/ceph-2/81ach
Running command: /usr/bin/chopt -H cephadm /usr/lib/ceph/ceph-2/block
Running command: /usr/bin/chopt -H cephadm /dev/dm-9
Running command: /usr/bin/chopt -H cephadm /usr/lib/ceph/ceph-2
Running command: /usr/bin/systemctl enable ceph-volume@lvs-2-384d08e-fec3-4786-0d13-378a-f6551c53
Running command: /usr/bin/systemctl enable --unit=ceph-oad02
Running command: /usr/bin/systemctl start ceph-oad02
--- ceph-volume lvs activate successful for and ID: 2

```

2.6.6 启动服务并修改相应文件

这里同样需要注释掉配置文件中的 cluster-ip。

```

root@node02:~# systemctl status ceph-oad01
● ceph-oad01.service - Ceph object storage daemon oad.1
   Loaded: Loadnd (/lib/systemd/system/ceph-oad01.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-05-29 22:52:07 CST; 4d ago
   Process: 3918 ExecStartPre=/usr/lib/ceph/ceph-oad-prestart.sh --cluster 5(CLUSTER) --id 1 (code=exited, status=0/SUCCESS)
   Main PID: 3922 (ceph-oad)
   Tasks: 63
   Memory: 28.3M
   CPU: 230ms
   CGroup: /system.slice/system-ceph.slice/ceph-oad01.service
           └─3922 /usr/bin/ceph-oad -f --cluster ceph --id 1 --setuser ceph --antgroup ceph

May 29 22:52:07 node02 systemd[1]: Starting Ceph object storage daemon oad.1...
May 29 22:52:07 node02 systemd[1]: Started Ceph object storage daemon oad.1.
May 29 22:52:08 node02 ceph-oad[3922]: 2023-05-29T22:52:08.075+0800 7f6143b1d5c0 -1 Falling back to public interface
May 29 22:52:10 node02 ceph-oad[3922]: 2023-05-29T22:52:10.167+0800 7f6143b1d5c0 -1 oad.1 @ log_to_monitors: true

```

2.7node03OSD 设置

2.7.1 导入引导密钥环

```

root@node03:~# ceph auth get client.bootstrap-osd -o /var/lib/ceph/bootstrap-osd/ceph.keyring
exported keyring for client.bootstrap-osd
root@node03:~# ceph auth get client.bootstrap-osd -o /etc/ceph/ceph.client.bootstrap-osd.keyring
exported keyring for client.bootstrap-osd
root@node03:~#

```

2.7.2 消除分区信息

```

root@node03:~# dd if=/dev/zero of=/dev/vdb bs=512K count=1
1+0 records in
1+0 records out
524288 bytes (524 kB, 512 KiB) copied, 8.98614927 s, 85.3 MB/s
root@node03:~#

```

2.7.3 创建 osd

```

--> ceph-volume lvm activate successful for osd ID: 2
--> ceph-volume lvm create successful for: /dev/vdb
root@node03:~#

```

2.7.4 列出与 Ceph 关联的逻辑卷和设备

```

root@node03:~# sudo ceph-volume lvm list

===== esd.2 =====

(black) /dov/ceph-Ta0c804f-f0bb-4b6d-9618-61b0724f2400/esd-block-5d4da6ac-fac3-4786-bd83-378076951c53

    block device          /dev/ceph-Ta9c8e4f-f0bb-4b6d-0018-61be724f24e0/osd-block-5d4da6ac-fac3-4786-bd83-378076951c53
    block uuid            E8p17i-qPNe-2t9A-PtRe-nje2-NU66-agA2ah
    cephx lockbox secret
    cluster-fnid          24956837-3602-A275-b7a6-616d4785bb54
    cluator name           ceph
    crush device class
    encrypted              0
    osd fsid               5d4da6ac-fac3-4706-bd83-378076951c55
    bsd id                  2
    oadapec affinity
    type                    block
    vdo                     8
    devices                 /dev/vdb

```

2.7.5 准备就绪的 osd 进行激活

```

root@node01:~# auto ceph-volume lvm activate 2 5d4da6ac-fac3-4786-bd83-378076951c53
Running command: /usr/bin/chown -R ceph:ceph /var/lib/ceph/osd/ceph-2
Running command: /usr/bin/ceph-bluestore-tool --cluster-uuid=24956837-3602-A275-b7a6-616d4785bb54 --fsid=5d4da6ac-fac3-4786-bd83-378076951c55 --path=/var/lib/ceph/osd/ceph-2 --osd-uuid=5d4da6ac-fac3-4786-bd83-378076951c55 /var/lib/ceph/osd/ceph-2/activate
Running command: /usr/bin/ceph-bluestore-tool --cluster-uuid=24956837-3602-A275-b7a6-616d4785bb54 --fsid=5d4da6ac-fac3-4786-bd83-378076951c55 --path=/var/lib/ceph/osd/ceph-2 --osd-uuid=5d4da6ac-fac3-4786-bd83-378076951c55 /var/lib/ceph/osd/ceph-2/activate
Running command: /usr/bin/chown -R ceph:ceph /var/lib/ceph/osd/ceph-2
Running command: /usr/bin/chown -R ceph:ceph /var/lib/ceph/osd/ceph-2
Running command: /usr/bin/systemctl enable ceph-volume@lvm-2-5d4da6ac-fac3-4786-bd83-378076951c53
Running command: /usr/bin/systemctl enable --unit=ceph-osd@2
Running command: /usr/bin/systemctl start ceph-osd@2
--- ceph-volume lvm activate successful for osd ID: 2

```

2.7.6 启动服务并修改相应文件

```

root@node01:~# systemctl enable ceph-osd@2
Created symlink /etc/systemd/system/ceph-osd.target.wants/ceph-osd@2.service to /usr/lib/systemd/system/ceph-osd@2.service.
root@node01:~# systemctl start ceph-osd@2
root@node01:~# systemctl status ceph-osd@2
* ceph-osd@2.service - Ceph object storage daemon on osd.2
   Loaded: loaded (/usr/lib/systemd/system/ceph-osd@2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-05-29 22:50:23 CST; 1min 20s ago
     Main PID: 3114 (ceph-osd)
        CGroup: /systemd/system/ceph-osd@2.service
               └─3114 /usr/bin/ceph-osd -f --cluster-ceph --id 2 --setuser ceph --setgroup ceph

May 29 22:50:21 node01 systemd[1]: Starting Ceph object storage daemon on osd.2...
May 29 22:56:21 node01 systemd[1]: Started Ceph object storage daemon on osd.2.
May 29 22:56:25 node01 ceph-osd[3114]: 2021-05-20T22:56:25.673-0800 713073421508 -1 Falling back to public interface
May 29 22:56:25 node01 ceph-osd[3114]: 2023-05-20T22:56:25.617-0800 713073421508 -1 osd.2 0 Log to monitors true
May 29 22:56:27 node01 ceph-osd[3114]: 2021-05-20T22:56:27.113-0800 713073421508 -1 osd.2 0 waiting for initial oadaus
May 29 22:56:27 node01 ceph-osd[3114]: 2021-05-20T22:56:27.129-0800 713073421508 -1 osd.2 31 but none of them is able to identify public interface
Link 1-10/16 CEND

```

当三个 osd 结点全部部署完毕后，使用 `cephostree` 查看个节点情况，可见都处于 up 状态，成功部署。

```

root@node03:~# ceph osd tree

ID CLASS WEIGHT TYPE NAME STATUS REWEIGHT PRI-AFF
-1 0.11728 root default
-3 0.03909 host node01
0 hdd 0.03909 osd.0 up 1.00000 1.00000
-5 0.03909 host node02
1 hdd 0.03909 osd.1 up 1.00000 1.00000
-7 0.03909 host node03
2 hdd 0.03909 osd.2 up 1.00000 1.00000

```

2.8MGR 添加

2.8.1 创建 mgr 密钥环

```

root@node01:~# ceph auth get-or-create mgr.node01 mon 'allow profile mgr' osd 'allow *' nds 'allow *'
[ mgr.node01 ]
key = AQA7vnrK5MLcLRAASn4bu60qvNGrZ2PhLzdemN==

```

2.8.2 创建 mgr 节点目录

创建 mgr 节点目录，节点目录默认名称为集群名称+节点名称。

```
root@node01:~# sudo -u ceph mkdir /var/lib/ceph/mgr/ceph-node01
root@node01:~#
```

2.8.3 获取 mgr 密钥环

获取 mgr 密钥环，导出到节点目录下，名称为默认为 keyring

```
root@node01:~# ceph auth get mgr.node01 -o /var/lib/ceph/mgr/ceph-node01/keyring
exported keyring for mgr.node01
root@node01:~#
```

2.8.4 启动服务

```
root@node01:~# systemctl daemon-reload
root@node01:~# systemctl start ceph-argr@node01
root@node01:~# systemctl status ceph-argr@node01
* ceph-argr@node01.service - Ceph cluster manager daemon
   Loaded: loaded (/lib/systemd/ceph-argr@node01.service; disabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-05-29 21:01:20 CST; 5s ago
     Main PID: 11470 (ceph-argr)
        Tasks: 11 (limit: 2000)
      Memory: 213.0M
         CPU: 4.001s
    CGroup: /system.slice/systemd-ceph@node01.service
            └─11470 /usr/bin/ceph-argr -f --cluster ceph --id node01 --setuid ceph --setgid ceph

May 29 23:01:31 node01 ceph-argr@node01: 2021-05-29T23:01:31.230-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module telegraf has missing NOTIFY_TYPES member
May 29 23:01:31 node01 ceph-argr@node01: 2021-05-29T23:01:31.814-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module volumen has missing NOTIFY_TYPES member
May 29 23:01:31 node01 ceph-argr@node01: 2021-05-29T23:01:31.838-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module tnet_orchestrator has missing NOTIFY_TYPES member
May 29 23:01:31 node01 ceph-argr@node01: 2021-05-29T23:01:31.900-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module og_autostaler has missing NOTIFY_TYPES member
May 29 23:01:32 node01 ceph-argr@node01: 2021-05-29T23:01:32.210-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module alorta has missing NOTIFY_TYPES member
May 29 23:01:32 node01 ceph-argr@node01: 2021-05-29T23:01:32.314-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module iabbis has missing NOTIFY_TYPES member
May 29 23:01:32 node01 ceph-argr@node01: 2021-05-29T23:01:32.514-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module inflax has missing NOTIFY_TYPES member
May 29 23:01:32 node01 ceph-argr@node01: 2021-05-29T23:01:32.608-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module ead support has missing NOTIFY_TYPES member
May 29 23:01:32 node01 ceph-argr@node01: 2021-05-29T23:01:32.788-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module toelist has missing NOTIFY_TYPES member
May 29 23:01:32 node01 ceph-argr@node01: 2021-05-29T23:01:31.840-0000 7f8e7e7c1dc8 -1 mgrIpy1 Module status has missing NOTIFY_TYPES member
root@node01:~#
```

成功启动服务之后可以发现集群概况中，多出了 mgr 服务，服务位于 node01 结点。

```
root@node01:~# ceph -s
cluster:
   id:           24956837-3602-4276-b7a6-616d4785bb54
  health: HEALTH_OK

services:
   mon: 3 daemons, quorum node01,node02,node03 (age 39m)
   mgr: node01(active, since 22s)
   osd: 3 osds: 3 up (since 5m), 3 in (since 21m)

date:
  pools:   1 pools, 1 pgs
 objects: 2 objects, 449 KiB
  usage:   49 MiB used, 120 GiB / 120 GiB avail
   pgs:    1 active+clean

io:
  client:   767 B/s rd, 22 KiB/s wr, 8 op/s rd, 1 op/s wr

root@node01:~#
```

2.9 添加 dashboard 对集群进行基本情况监视。

2.9.1 启用 dashboard

使用 aptinstall-yceph-mgr-dashboard 成功安装 ceph-mgr-dashboard 之后启用 dashboard。


```
root@node01:~# ceph mgr module enable dashboard
root@node01:~#
```

2.9.2 自建证书供 ssl 使用

```
root@node01:~# ceph dashboard create-self-signed-cert
Self-signed certificate created
```

[illegible]

2.9.3 导入证书文件和私钥文件

```
root@node01:~# ceph dashboard set-ssl-certificate -i dashboard.crt
SSL certificate updated
root@node01:~# ceph dashboard set-ssl-certificate-key -i dashboard.key
SSL certificate key updated
```

2.9.4 设置dashbaord地址和端口

```
root@node81:~# ceph config set mgr mgr/dashboard/server_addr 172.15.249.78
root@node81:~# ceph config set mgr mgr/dashboard/server_port 8888
root@node81:~# ceph config set mgr mgr/dashboard/ssl_server_port 8443
root@node81:~#
```

2.9.5 创建管理员用户admin

首先创建管理员的账户密码文件，文件内容为admin@123，即表示账户密码为admin@123。

```
root@node01:~# vim /root/cephpasswd
```

创建完账号密码文件之后即可依据该文件创建管理员用户。

```
root@node01:~# coryn dashboard ac-user-create admin -i /root/coahpasowd administrator
{"username": "admin", "password": "52b125I8v8HAgpTv5x3hVahBI0cGRdCHhF3m1n42xU4u0CaKMcgsbP32C", "roles": ["administrator"], "name": null, "email": null, "lastUpdate": 1685372994, "unabled": true, "pwdExpirationDate": null, "awdUpdateRequired": false}
root@node01:~#
```

配置完成后，使用`cephconfigdump`即可查看集群的配置情况，`dashboard`被部署在172.16.249.78:8443(https)以及172.16.249.78:8000(http).

```

root@ade01:~# ceph config dump
MHD      MASK      LEVEL      OPTION                                     VALUE                                     RO
mon              advanced  non_warn_on_insecure_global_id_reclaim_allowed  false
mgr              advanced  mgr/dashboard/server_addr                    172.16.249.78  *
mgr              advanced  mgr/dashboard/server_port                     8080           *
mgr              advanced  mgr/dashboard/sel_server_port                 8443           *
osd.0           basic     osd_max_object_size_opa_hdd                   20131.390007
osd.1           basic     osd_max_object_size_opa_hdd                   17753.057001
osd.2           basic     osd_max_object_size_opa_hdd                   17281.179272

```

设置完成后重新启动mrg服务。

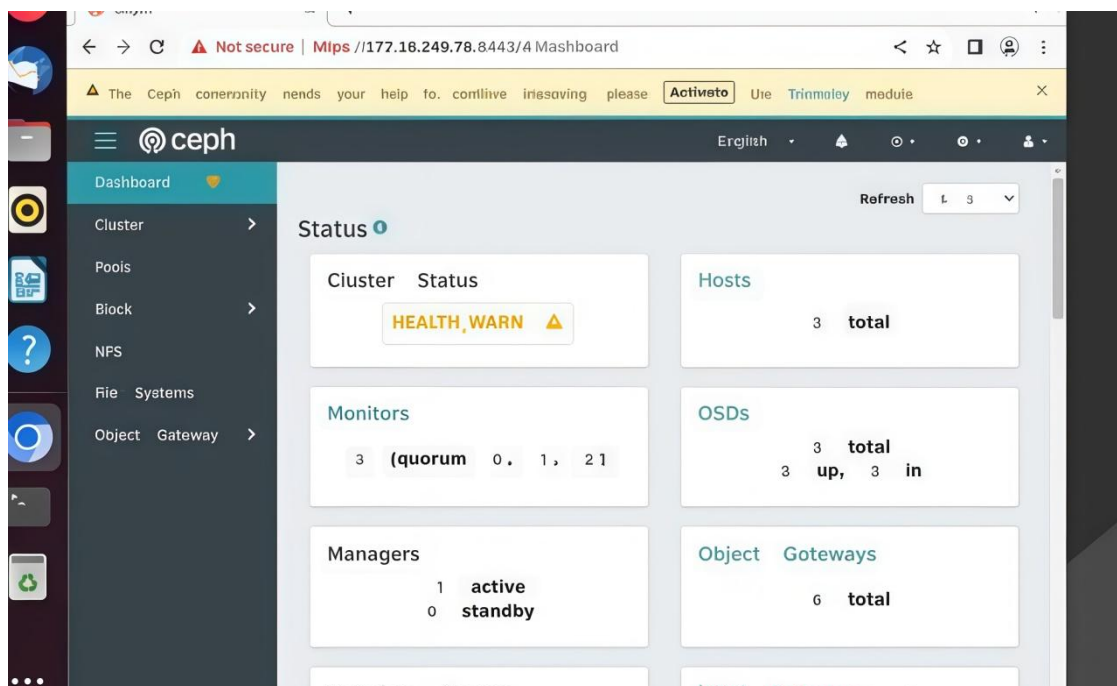
```
root@node01:~# systemctl restart ceph-mgr@node01
```

由于dashboard被部署在内网之中，如果希望能够公网访问则需要对公网ip和内网ip进行端口映射，较为麻烦，因此我们这里为服务器安装图形化用户界面，便于在机器中访问dashboard。

按照(参考资料-6)安装图形化界面，然后sudo snap install chromium 安装浏览器

```
root@node01:~# ceph mgr services
{
  "dashboard": "https://172.16.249.78:8443/"
}
```

查看服务部署的ip，浏览器访问https://172.16.249.78:8443即可，登录即可进入后台。



这里由于开启了debug模式，所以状态时warn

2.10 配置测试 ceph 块存储服务

2.10.1 创建一个rbd存储池

创建一个rbd存储池,并对存储池进行初始化。

```
root@node01:~# ceph osd lspools
1 .mgr
root@node01:~# ceph osd pool create rbd 8 8
pool 'rbd' created
root@node01:~# rbd pool init rbd
root@node01:~#
```

2.10.2 创建镜像

```

root@node01:~# rbd create --size 10248 rbd/image1
root@node01:~# rbd ls rbd
image1
root@node01:~# rbd info rbd/image
rbd: error opening image image: (2) No such file or directory
root@node01:~# rbd info rbd/image1
rbd image 'image1':
    size 10 018 in 2560 objects
    order 22 (4 MiB objects)
    snapshot_count: 0
    id: 148e9c613278f
    block_name_prefix: rbd_data.148e9c613278f
    format: 2
    features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
    op_features:
    flags:
    create_timestamp: Tue May 30 17:04:20 2023
    access_timestamp: Tue May 30 17:04:29 2023
    modify_timestamp: Tue May 30 17:04:20 2023
root@node01:~#

```

2.10.3创建快照

```

root@node01:~# rbd snap create rbd/image1@2023
Creating snap: 100% complete...done.
root@node01:~# rbd ls -l rbd
NAME          SIZE    PARENT  FMT  PROT  LOCK
image1        10 GiB             2
image1@2023   10 GiB             2

```

2.10.4删除镜像快照

```

root@node01:~# rbd snap rm rbd/image1@2023
Removing snap: 100% complete...done.

```

2.10.5创建两个存储池images和vms

```

root@node01:~# ceph osd pool create images 64 64
pool 'images' created
root@node01:~# ceph osd pool create vms 64 64
pool 'vms' created
root@node01:~#

```

2.10.6在images存储池中创建一个镜像

```

root@node01:~# rbd create --size 128 vms/vms01
root@node01:~# rbd info vms/vms01
rbd image 'vms01':
    size 128 MiB in 32 objects
    order 22 (4 MiB objects)
    snapshot_count: 0
    id: 149378e93d1f2
    block_name_prefix: rbd_data.149378e93d1f2
    format: 2
    features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
    op_features:
    flags:
    create_timestamp: Tue May 30 17:10:40 2023
    access_timestamp: Tue May 30 17:10:40 2023
    modify_timestamp: Tue May 30 17:10:40 2023

```

2.10.7给镜像创建快照

```

root@node01:~# rbd snap create vms/vms01@2023
Creating snap: 100% complete...done.
root@node01:~# rbd snap ls vms/vms01
SNAPID NAME SIZE PROTECTED TIMESTAMP
4 2023 128 MiB Tue May 30 17:11:48 2023
root@node01:~# rbd ls -l vms
NAME SIZE PARENT FMT PROT LOCK
vms01 128 MiB 2
vms01@2023 128 MiB 2

```

2.10.8对快照进行保护并克隆

快照在进行克隆之前，必须实现对被克隆的快照进行保护，避免数据被意外篡改。

快照进入保护状态之后才允许进行克隆操作。

```

root@node01:~# rbd anap protect vns/vns01@2023
root@node01:~# rbd clone vna/vna01@2023 vna/vns01_clone
root@node01:~# rbd info .nrg/vns01@clone
rbd: error opening image vna01: (2) No such file or directory
root@node01:~# rbd info .nrg/vna01_clone
rbd: error opening image vna01_clone: (2) No such file or directory
root@node01:~# rbd info --info vns/vns01_clone
rbd: unrecognized option '--info'
root@node01:~# rbd info --image vna/vns01_clone
rbd image 'vna01_clone':
    size 128 MiB in 32 objects
    order 22 (4 MiB objects)
    snapshot_count: 8
    id: 14a1ca6a073c9
    block_name_prefix: rbd_data.14a1ca6a073c9
    format: 2
    features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
    op_features:
    flags:
    create_timestamp: Tue May 30 17:34:19 2023
    access_timestamp: Tue May 30 17:34:18 2023
    modify_timestamp: Tue May 30 17:34:18 2023
    parent: vns/vns01@2023
    overlap: 128 MiB

```

2.10.9将父镜像信息合并到子镜像

```

root@node01:~# rbd flatten .nrg/vns01_clone
rbd: error opening image vns01_clone: (2) No such file or directory
root@node01:~# rbd flatten vns/vns01_clone
Image flatten: 100% complete...done.
root@node01:~# rbd info vms/vns01_clone
rbd image 'vna01_clone':
    size 128 MiB in 32 objects
    order 22 (4 MiB objects)
    snapshot_count: 8
    id: 14a3ca6a073c9
    block_name_prefix: rbd_data.14a3ca6a073c9
    format: 2
    features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
    op_features:
    flags:
    create_timestamp: Tue May 30 17:34:18 2023
    access_timestamp: Tue May 30 17:34:18 2023
    modify_timestamp: Tue May 30 17:34:16 2023
root@node01:~#

```

2.11 配置 ceph-fs 文件存储服务并测试

2.11.1创建节点目录并生成、导出密钥


```

root@node01:~# audo -u ceph nkdir -p /var/lib/ceph/nda/ceph-node01
root@node01:~# ceph auth get-or-create nds.nodo01 oed "allow rxx" nds "allow" mon "allow profile mds"
[nds.nodo01]
    key = AQ85xHVk70XXExAAfqtX7gi-L52MyikkLA607g--
root@node01:~# ceph auth get mds.nodo01 -o /var/lib/ceph/nda/ceph-node01/keyring
exported keyring for nds.nodo01
root@node01:~#

```

2.11.2修改ceph.conf

```

^ Egloball
fsid = 24956857-3602-4276-b7a6-616d4785bb54
mon initial members = node01
mon host = 172.16.249.76
public network = 172.16.249.8/24
#cluster network = 192.166.100.0/24
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd journal aite = 1924
osd pool default aite = 1
osd pool default nin size = 2
osd pool default pg num = 16
osd pool default pgp num = 16
osd crush choosoloaf typo = 1
[nde.nodo01]
host-noda01
-
-

```

2.11.3启动服务

```

root@node01:~# systemctl start ceph-nda@node01
root@node01:~# systemctl status ceph-ads@node01
* ceph-ads@node01.service - Ceph adadata server daemon
   Loaded: loaded (/lib/systemd/systemd/ceph-ads@node01.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-05-30 17:42:25 CST; 5s ago
     Main PID: 9070 (ceph-ads)
        Tasks: 16
       Memory: 12.7M
          CPU: 112ms
      CGroup: /system.slice/systemd-ceph\x2dads@node01.service
              └─9070 /usr/bin/ceph-ads -f --cluster ceph --id node01 --setuser ceph --aagroup ceph

May 30 17:42:25 node01 systemd[1]: Started Ceph putadata server daemon.
May 30 17:42:25 node01 ceph-ads(9070): starting nda.nodo01 at
root@node01:~# systemctl enable ceph-ads@node01
Created symlink /etc/systemd/systemd/ceph-ads.target.wants/ceph-ads@node01.service -- /lib/systemd/systemd/ceph-nda@node01.service.
root@node01:~#

```

2.11.4创建文件存储所需存储池

```

root@node01:~# ceph osd pool create cephfs_data 32
pool 'cephfs_data' created
root@node01:~# ceph osd pool create cephfs_metadata 32
pool 'cephfs_metadata' created
root@node01:~# ceph osd pool application enable cephfs_data cephfs
enabled application 'cephfs' on pool 'cephfs_data'
root@node01:~# ceph osd pool application enable cephfs_metadata cephfs
enabled application 'cephfs' on pool 'cephfs_metadata'
root@node01:~#

```

2.11.5创建文件系统

```

root@node01:~# ceph fs new cephfs cephfs_metadata cephfs_data
new fs with metadata pool 5 and data pool 5
root@node01:~# ceph fs ls
name: cephfs, metadata pool: cephfs_metadata, data pools: [cephfs_data ]
root@node01:~# ceph fs status cephfs
cephfs - 8 clients
=====
RANK STATE NDS ACTIVITY DNS INGS DIRS CAPS
  0 active node01 Reqs: 8 /s 18 13 12 9
    POOL TYPE USED AVAIL
cephfs_metadata metadata 96.8k 37.9G
cephfs_data data 0 37.9G
MDS version: ceph version 17.2.5 (98316ao89f1a893a5dod3a640405cddb33o88757) quincy (atablo)
root@node01:~# ceph nds stat
cephfs:1 (0=node01=up:active)
root@node01:~#

```

2.11.6创建cephfs用户，并允许该用户访问cephfs池

```
root@node01:~# ceph fs authorize cephfs client.cephfs / rw
[client.cephfs]
  key - AQDAXnVknZAFcBAAJfQh+Ns7IE6hPJlSU6U6Mg==
root@node01:~# ceph auth get-or-create client.cephfs -o /etc/ceph/client.cephfs.keyring
root@node01:~#
```

切记，令牌环导出命令中密钥文件应当是ceph.client.cephfs.keyring，并非client.cephfs.keyring，如果导入到后者那么后续mount挂载步骤将会出错，正确的导出命令如下所示。

```
root@node01:~# ceph auth get-or-create client.cephfs -o /etc/ceph/ceph.client.cephfs.keyring
root@node01:~#
```

2.11.7客户端安装ceph-common

```
root@node02:~# apt install ceph-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ceph-common is already the newest version (17.2.5-0ubuntu0.22.04.5).
ceph-common set to manually installed.
The following packages were automatically installed and are no longer required:
  libnss3 libnss3-tools libnss3-ldap libnss3-ldap-ldapd python3-babel python3-jinja2 python3-jinja2-jinja2 python3-jinja2-jinja2 python3-jinja2-jinja2
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 110 not upgraded.
root@node02:~# cp node01:/etc/ceph/client.cephfs.keyring /etc/ceph/keyring
client.cephfs.keyring
root@node02:~#
```

2.11.8将client.cephfs.keyring文件拷贝到客户端

```
root@node01:~# scp node02:/etc/ceph/client.cephfs.keyring /etc/ceph/keyring
ceph.client.cephfs.keyring
root@node01:~#
```

2.11.9挂载文件系统

```
root@node01:~# mount -t ceph 172.16.249.78:6789:/ /mnt/fs -o name=cephfs
root@node01:~#
```

在2.11.8步骤中将服务端(node01)的令牌环复制到客户端后(node02)，在客户端中重复服务端中所做的创建文件夹以及挂载操作。

```
root@node02:~# mkdir /mnt/fs
root@node02:~# mount -t ceph 172.16.249.78:6789:/ /mnt/fs/ -o name=cephfs
root@node02:~# df -hT
Filesystem                Type      Size  Used Avail Use% Mounted on
tmpfs                     tmpfs     179M  1.1M  178M   1% /run
/dev/vda1                 ext4      400G  4.2G   33G  12% /
tmpfs                     tmpfs     892M    0  892M   0% /dev/shm
tmpfs                     tmpfs     5.0M  4.0K   5.0M   1% /run/lock
tmpfs                     tmpfs     892M  28K   892M   1% /var/lib/ceph/osd/ceph-1
tmpfs                     tmpfs     179M  4.8K   179M   1% /run/user/8
172.16.249.78:6789:/ ceph 38G    0   38G   0% /mnt/fs
root@node02:~#
root@node02:~#
```

操作完成后，node02和node01将共享/mnt/fs，如下所示node02新建文件操作在node01总同样有效。

```
root@node02:~# touch /mnt/fs/demo.txt
root@node02:~#
```

```
root@node01:~# ls /mnt/fs
demo.txt
```

2.12 配置 ceph-rgw 对象存储网关提供对象存储并测试

2.12.1 安装radosgw后修改配置文件

在配置文件中添加如下内容:

```
[client.rgw.node01]
```

```
host=node01
```

```
rgwfrontends="beastport=7480"
```

```
rgwdnsname=node01
```

```
[global]
```

```
fsid = 24956837-3602-4276-b7a6-616d4785bb54
```

```
mon initial members = node01
```

```
mon host = 172.16.249.78
```

```
public network = 172.16.249.0/24
```

```
#cluster network = 192.168.100.0/24
```

```
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 = 32
```

```
osd pool default pgp num = 32
```

```
osd crush chooseleaf type = 1
```

```
[mds.node01]
```

```
host=node01
```

```
[client.rgw.node01]
```

```
host = node01
```

```
rgw frontends = "beast port=7480"
```

```
rgw dns name = node01
```

2.12.2 创建节点目录、client.rgw用户和授权并导出密钥环到节点目录下

```
root@node01:~# sudo -u ceph mkdir -p /var/lib/ceph/crdir/ceph-tgw.node01
root@node01:~# ceph auth get-or-create client.rgw.node01 osd 'allow rwx' mon 'allow rw' -o /var/lib/ceph/radosgw/ceph-tgw.node01/kmyring
-both: /var/lib/ceph/radosgw/ceph-tgw.node01/kmyring: No such file or directory
root@node01:~# ceph auth get-or-create client.rgw.node01 mon 'allow rwx' mon 'allow rw' -o /var/lib/ceph/radosgw/cdph-rgw.node01/kuyring
root@node01:~#
```

2.12.3 创建存储池

如果在创建存储池的过程中遇到如下情况,说明pgs设置数量较小,参考(参考资料-7),为配置文件中添加mon_max_pg_per_osd=2000。

```
root@node01:~# ceph osd pool create default.rgw.luchaili.data 10 16
Error: ERANGE: pg_num 16 size 3 would mean 771 total pgs, which exceeds max 780 (mon_max_pg_per_osd 280 * num_in_osds 3)
root@node01:~# ceph osd pool create default.rgw.buchats.ncn-ec 10 16
Error: ERANGE: pg_num 16 size 1 would mean 771 total pgs, which exceeds max 780 (mon_max_pg_per_osd 280 * num_in_osds 3)
root@node01:~# vim /etc/ceph/ceph.conf
root@node01:~# ceph osd pool create default.rgw.control 16 16
Error: ERANGE: pg_num 16 size 3 would mean 771 total pgs, which exceeds max 780 (mon_max_pg_per_osd 280 * num_in_osds 3)
root@node01:~# vim /etc/ceph/ceph.conf
root@node01:~# ceph osd pool create default.rgw.control 16 16
Error: ERANGE: pg_num 16 size 3 would mean 771 total pgs, which exceeds max 780 (mon_max_pg_per_osd 280 * num_in_osds 3)
```

```
[global]
faid - 24956837-3602-4276-b7a6-616d4785bb54
non initial members - node01
non host - 172.16.249.78
public network - 172.16.249.0/24
cluster network - 192.166.168.0/24
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 - 32
osd pool default pgp num - 32
osd crush chooseloaf type - 1
non_max_pg_per_osd - 2000
[mon.node01]
host=node01
[client.rgw.node01]
host - node01
rgw frontends - 'beast port=7480'
rgw dns name - node01
```

```
root@node01:~# vim /etc/ceph/ceph.conf
root@node01:~# systemctl restart ceph-daemon@node01
root@node01:~# ceph osd pool create default.rgw.control 16 16
pool 'default.rgw.control' created
root@node01:~# ceph osd pool create default.rgw.meta 16 16
pool 'default.rgw.meta' created
root@node01:~# ceph osd pool create default.rgw.log 16 16
pool 'default.rgw.log' created
root@node01:~# ceph osd pool create default.rgw.buckets.index 10 16
pool 'default.rgw.buckets.index' created
root@node01:~# ceph osd pool create default.rgw.buckets.data 16 16
pool 'default.rgw.buckets.data' created
root@node01:~# ceph osd pool create default.rgw.buckets.non-ec 16 16
pool 'default.rgw.buckets.non-ec' created
```

2.12.4 设置存储池类型为rgw

```
root@node01:~# ceph osd pool application enable .rgw.root rgw
enabled application 'rgw' on pool '.rgw.root'
root@node01:~# ceph osd pool application enable default.rgw.control rgw
enabled application 'rgw' on pool 'default.rgw.control'
root@node01:~# ceph osd pool application enable default.rgw.meta rgw
enabled application 'rgw' on pool 'default.rgw.meta'
root@node01:~# ceph osd pool application enable default.rgw.log rgw
enabled application 'rgw' on pool 'default.rgw.log'
root@node01:~# ceph osd pool application enable default.rgw.buckets.index rgw
enabled application 'rgw' on pool 'default.rgw.buckets.index'
root@node01:~# ceph osd pool application enable default.rgw.buckets.data rgw
enabled application 'rgw' on pool 'default.rgw.buckets.data'
root@node01:~# ceph osd pool application enable default.rgw.buckets.non-ec rgw
enabled application 'rgw' on pool 'default.rgw.buckets.non-ec'
root@node01:~#
```

2.12.5 启动服务

```
root@node01:~# systemctl start ceph-radosgw@rgw.node01
root@node01:~# systemctl status ceph-radosgw@rgw.node01
* ceph-radosgw@rgw.node01.service - Ceph rados gateway
   Loaded: loaded (/lib/systemd/system/ceph-radosgw@rgw.node01.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-05-30 20:20:35 CST; 1s ago
     Main PID: 3484 (radosgw)
        Tasks: 602
       Memory: 73.2M
          CPU: 310ms
     CGroup: /systemd/system/ceph-radosgw@rgw.node01.service
             └─5884 /usr/sbin/radosgw -f --cluster ceph --name client.rgw.node01 --autouser ceph --detgroup ceph

May 30 20:20:33 node01 systemd[1]: Started Ceph rados gateway.
root@node01:~# systemctl enable ceph-radosgw@rgw.node01
Created symlink /etc/systemd/system/ceph-radosgw.target.wants/ceph-radosgw@rgw.node01.service → /lib/systemd/system/ceph-radosgw@rgw.node01.service.
root@node01:~#
```

2.12.6 在RGW节点添加一个对象存储网关的管理员账户


```

root@node01:~# radosgw-admin user create --uid="adnin" --display-name="adnin user" --system
{
  "user_id": "adnin",
  "display_name": "adnin user",
  "email": "",
  "suspended": 0,
  "max_buckets": 1000,
  "subusers": [],
  "keys": [
    {
      "user": "adnin",
      "access_key": "I1MYRSH11890Y7H95NZH",
      "secret_key": "7EdixIObW0YCRzATSCV23NJKfjcdN+G6ShPKSXX"
    }
  ],
}

```

2.12.7保存access_key以及secret_key

```

"keys": [
  {
    "user": "adnin",
    "access_key": "I1MYRSH11890Y7H95NZH",
    "secret_key": "7EdixIObW0YCRzATSCV23NJKfjcdN+G6ShPKSXX"
  }
],

```

将箭头所指两个内容手动保存到/var/lib/ceph/radosgw/rgw_access_key, /var/lib/ceph/radosgw/rgw_secret_key。

2.12.8radosgw集成到dashboard，使用web界面管理

```

root@node01:~# ceph dashboard set-rgw-api-ssl-verify false
Option RGW_API_SSL_VERIFY updated

root@node01:~# ceph dashboard set-rgw-api-access-key -i /var/lib/ceph/radosgw/rgw_access_key
Option RGW_API_ACCESS_KEY updated
root@node01:~# ceph dashboard set-rgw-api-secret-key -i /var/lib/ceph/radosgw/rgw_secret_key
Option RGW_API_SECRET_KEY updated
root@node01:~#

```

2.13 使用 ceph 提供 iSCSI 服务，并测试

2.13.1修改ceph.conf

向ceph.conf中添加如下内容

```

[osd]
osd heartbeat grace = 20
osd heartbeat interval = 5

```

2.13.2让配置立即生效


```
root@node01:~# ceph dashboard iscsi-gateway-add -i iscsigw-node01
Success
```

```
root@node81:~# ceph dashboard iscsi-gateway-list
{"gateways": {"node81": {"service_url": "http://admin:admin@node81:5801 "}}}
```

2.13.6创建iSCSI网关，添加镜像

使用gwcli进入命令行，然后再进行如下操作。

```
root@node81:~# gwcli
Warning: Could not load preferences file /root/.gwcli/prefs.bin.
/> cd iqn.2022-01.com.test:testl/gateways
No such path /iqn.2022-01.com.test:testl
/> cd iscsi-targets
/iscsi-targets> cd iqn.2022-01.com.test:testl/gateways
No such path /iscsi-targets/iqn.2022-01.com.test:testl
/iscsi-targets> CO DISKS
Command not found CO
/iscsi-targets> ed disks
No such path /iscsi-targets/disks
/iscsi-targets> ed ..
/> cd disks
/disk> create pool=rbt image=disk_1 aize=10G
ok
```

```
root@node01:~# apt -y install epon-iscsi multipath-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
epon-iscsi is already the newest version (2.1.5-ubuntu1).
multipath-tools is already the newest version (0.5.8-ubuntu1.22.04.1).
The following packages were automatically installed and are no longer required:
  etatydets libeatydata1 libflashrom1 libftdl-2 python3-json-pointer python3-jsonpatch python3-jonschens python3-pyreisont
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 104 not upgraded.
```

2.13.6修改设置

```
## DO NOT EDIT OR REMOVE THIS FILE!  
## If you remove this file, the iSCSI daemon will not start.  
## If you change the InitiatorName, existing access control lists  
## may reject this initiator. The InitiatorName must be unique  
## for each iSCSI initiator. Do NOT duplicate iSCSI InitiatorNames.  
InitiatorName=iqn.2022-01.com.test:test1
```

```
node.session.auth.authmethod = CHAP
node.session.auth.username = student1
node.session.auth.password = hedjsunxlleid
#
# Opon-iSCSI default configuration.
# Could be located at /etc/iscsi/iscsid.conf or ~/.iscsid.conf
#
# Note: To set any of these values for a specific node/session run
# the iscsiadm --mode node --op command for the value. See the README
```

2.13.7发现服务

重启multipathd服务之后即可iscsiadm-mdiscovery-tst-p172.16.249.78发现目标节点，登录之后挂在即可。

```
^ defaults {  
    user_friendly_names yes  
}  
  
devices {  
    device {  
        vendor "LIO-ORG"  
        hardware_handler "1 alua"  
        path_grouping_policy "failover"  
        path_selector "queue-length 6"  
        failbeck 60  
        path_checker tur  
        prio alua  
        prio_args exclusive_pref_bit  
        fast_io_fail_tmo 25  
        no_path_retry queue  
    }  
}  
□
```