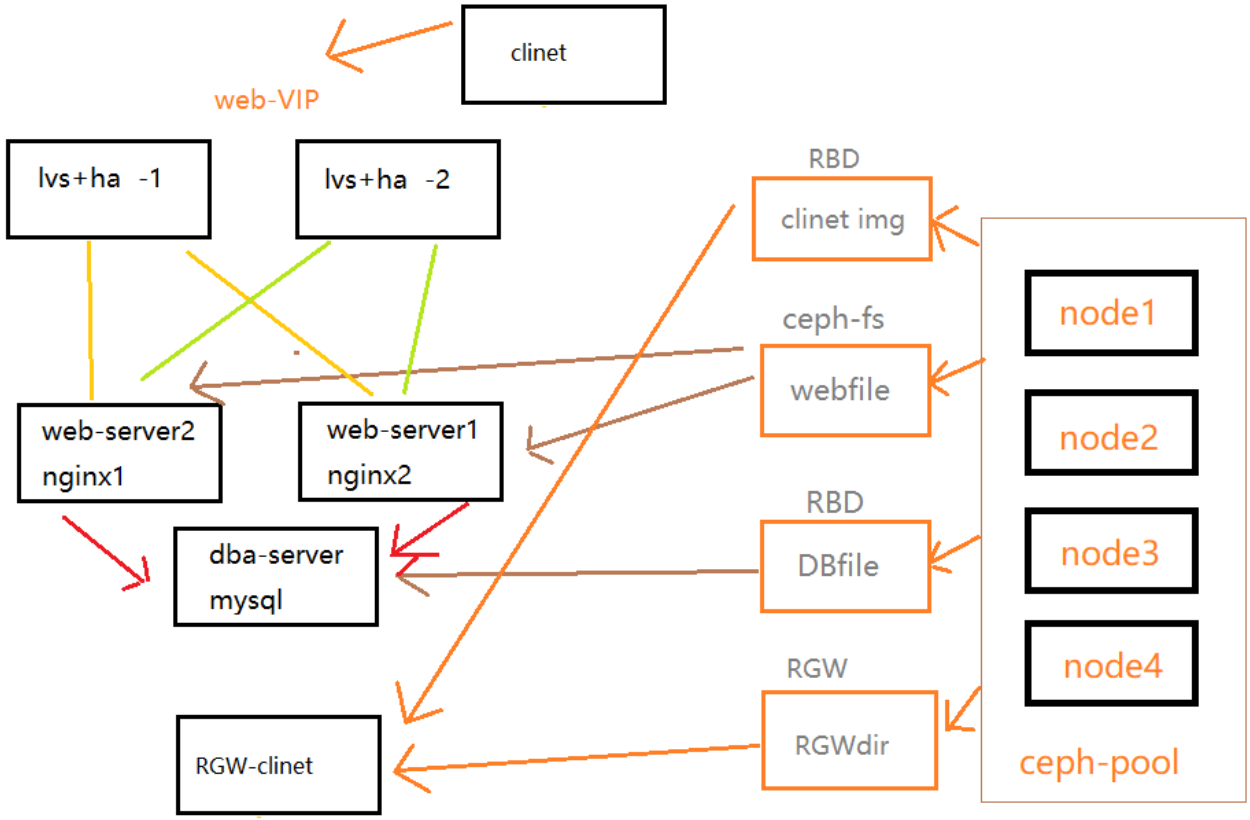
集群项目-lvs/ha/ceph

此项目主要是为锻炼综合技术为主，与实际生产环境项目有一定的区别；

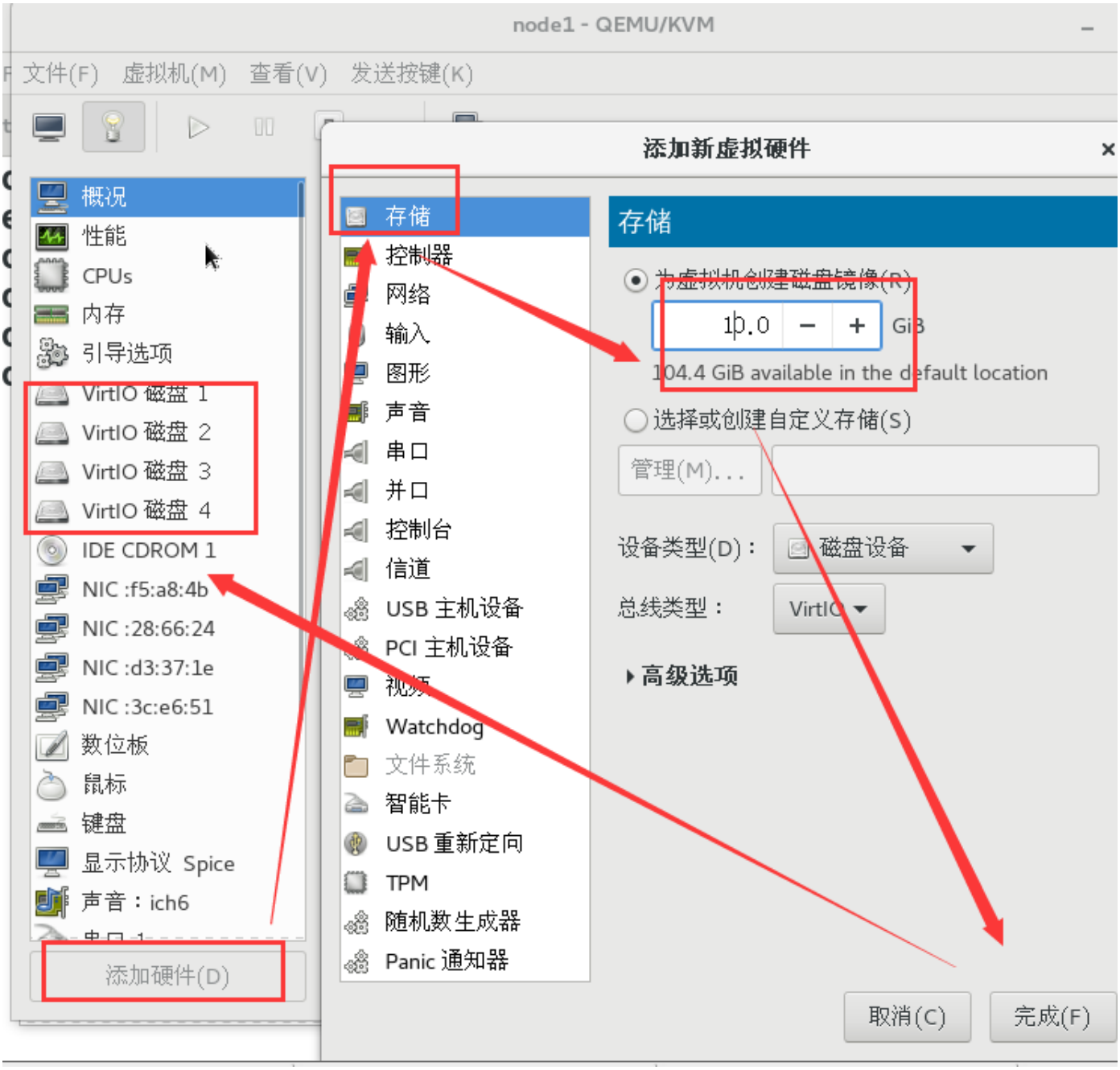
项目ip规划如下：
1、web-VIP:201.1.2.200 ， lvs_ha-1:192.168.4.254 ， lvs_ha-2:192.168.4.253 。
2、web-server1:192.168.4.100 ， web-server2:192.168.4.101 ,dba-server:192.168.4.50 。
3、clinet:201.1.2.100 ， RGW-clinet:192.168.4.201 。
4、node1-node5:192.168.4.1-5 。
node4为MDS,node5为RGW

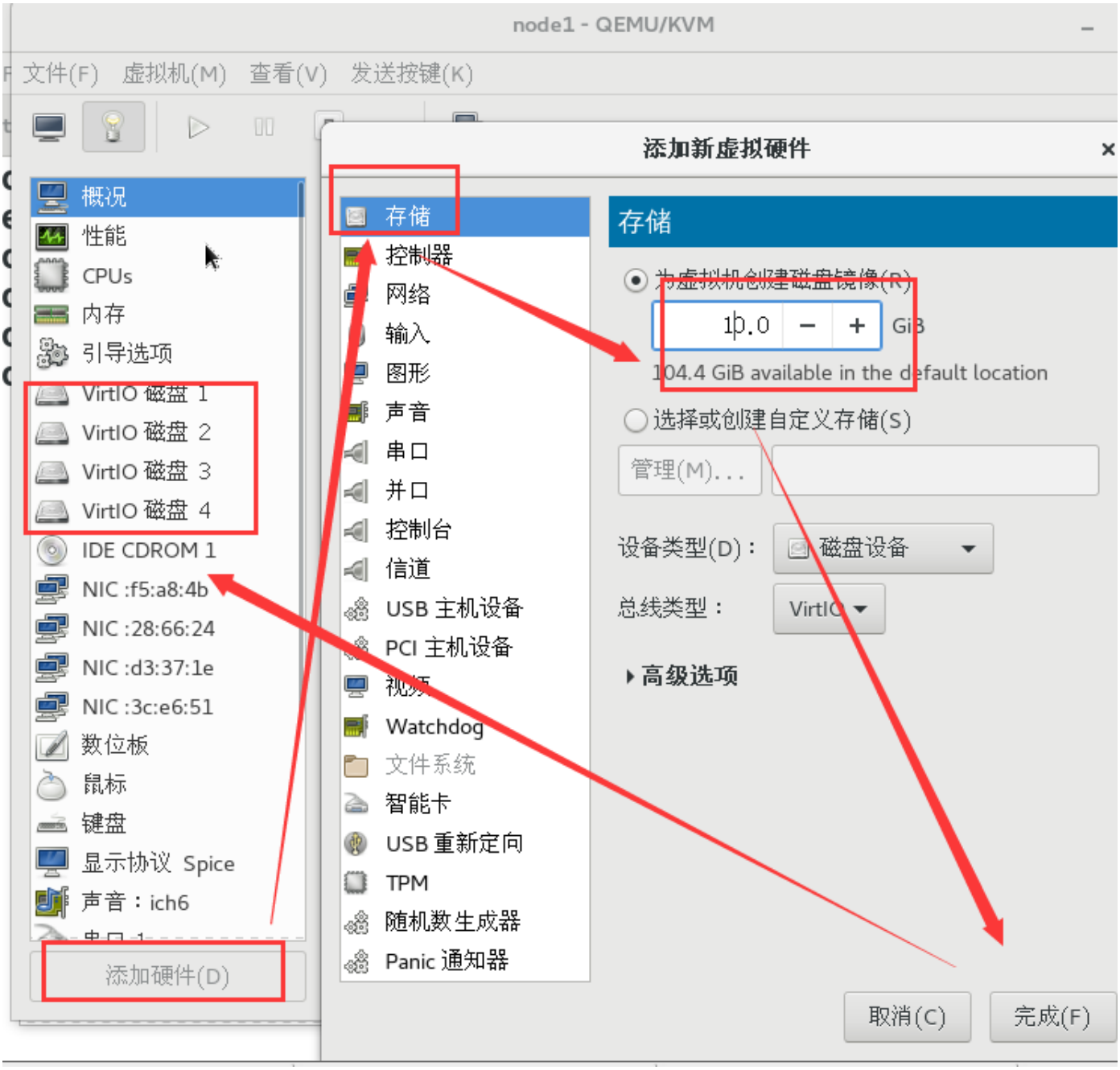


1.创建11台虚拟机并修改对应的ip跟主机名，机器直接ssh免密登录，配置主机名解析，新建一个RGW-clinet完成引导之后关闭机器。每一个node添加3个10G磁盘。

1.1 创建虚拟机并添加磁盘 。

[root@room00pc000 ~]# clone-auto7


node1 - QEMU/KVM
文件(F)
虚拟机(M)
查看(V)
发送按键(K)
概况
性能
CPUs
内存
引导选项
VirtIO 磁盘 1
VirtIO 磁盘 2
VirtIO 磁盘 3
VirtIO 磁盘 4
IDE CDROM 1
NIC :f5:a8:4b
NIC :28:66:24
NIC :d3:37:1e
NIC :3c:e6:51
数位板
鼠标
键盘
显示协议 Spice
声音：ich6
添加硬件(D)
添加新虚拟硬件
存储
控制器
网络
输入
图形
声音
串口
并口
控制台
信道
USB 主机设备
PCI 主机设备
视频
Watchdog
文件系统
智能卡
USB 重新定向
TPM
随机数生成器
Panic 通知器
存储
为虚拟机创建磁盘镜像(R)
10.0
GiB
104.4 GiB available in the default location
选择或创建自定义存储(S)
管理(M)...
设备类型(D)：
磁盘设备
总线类型：
VirtIO
高级选项
取消(C)
完成(F)



1.2 修改ip跟主机名，设置机器直接ssh 免密登录，配置主机名解析。

```
[root@room00pc000 ~]# for i in {1..5}
> do
>     echo -e "192.168.4.$i\tnode$i.tedu.cn\tnode$i" >> /etc/hosts
> done

### 编写成功后的hosts 文件
[root@room00pc000 ~]# tail -12 /etc/hosts

192.168.4.1 node1.tedu.cn    node1
192.168.4.2 node2.tedu.cn    node2
192.168.4.3 node3.tedu.cn    node3
192.168.4.4 node4.tedu.cn    node4
192.168.4.5 node5.tedu.cn    node5
192.168.4.252     lvs_ha-1.tedu.cn  lvs_ha-1
192.168.4.253     lvs_ha-2.tedu.cn  lvs_ha-2
192.168.4.100     web-server1.tedu.cn web-server1
192.168.4.101     web-server2.tedu.cn web-server2
192.168.4.50      dba-server.tedu.cn  dba-server
```

```
192.168.4.201    RGW-clinet.tedu.cn  RGW-clinet

### 将此机器制作成ftp服务器提供Yum 。
[root@room00pc000 ~]# cat /etc/yum.repos.d/ceph-server.repo
[rhel7.4]
name=rhel7.4
baseurl=ftp://192.168.4.254/rhel7
enabled=1
gpgcheck=0
[mon]
name=mon
baseurl=ftp://192.168.4.254/ceph/rhceph-2.0-rhel-7-x86_64/MON
enabled=1
gpgcheck=0
[osd]
name=osd
baseurl=ftp://192.168.4.254/ceph/rhceph-2.0-rhel-7-x86_64/OSD
enabled=1
gpgcheck=0
[tools]
name=tools
baseurl=ftp://192.168.4.254/ceph/rhceph-2.0-rhel-7-x86_64/Tools
enabled=1
gpgcheck=0


### 配置ip跟主机名
[root@localhost ~]# nmcli connection modify  eth0 ipv4.method  manual ipv4.addresses
192.168.4.5/24 connection.autoconnect yes
[root@localhost ~]# nmcli connection up eth0
[root@localhost ~]# hostnamectl  set-hostname  node5.tedu.cn

### 配置远程登录不需要询问yes或no
[root@room00pc000 ~]#  for i in `tail -11 /etc/hosts | head -10 > /etc/hosts.b ; awk -F' ' '{
print $3 }'  /etc/hosts.b`; do       ssh-keyscan $i >> /root/.ssh/known_hosts ; done

### 配置远程免密登录
[root@room00pc000 ~]# for i in `tail -11 /etc/hosts | head -10 > /etc/hosts.b ; awk -F' ' '{ print
$3 }'  /etc/hosts.b`; do       ssh-copy-id $i ; done

### 将本地的yum配置文件传输到几台服务器
[root@room00pc000 ~]# for i in `tail -11 /etc/hosts | head -10 > /etc/hosts.b ; awk -F' ' '{ print
$3 }'  /etc/hosts.b`; do       scp /etc/yum.repos.d/ceph-server.repo   $i:/etc/yum.repos.d/ ; done

### 将hosts文件传输到几台服务器
[root@room00pc000 ~]# for i in `tail -11 /etc/hosts | head -10 > /etc/hosts.b ; awk -F' ' '{ print
$3 }'  /etc/hosts.b`; do       scp  /etc/hosts.b   $i:/etc/hosts ; done

### 几台服务器之间也需要配置免密登录

[root@node1 ~]# ssh-keygen  -f /root/.ssh/id_rsa -N ''

[root@node1 ~]# for i in `awk -F' ' '{ print $3 }' /etc/hosts`; do     ssh-keyscan $i >>
/root/.ssh/known_hosts ; done

[root@node1 ~]# for i in `awk -F' ' '{ print $3 }' /etc/hosts`; do     ssh-copy-id $i ; done
```

2.配置ceph-pool， 并提供页面给web-server跟dba-server 以及

RGW-clinet 。

2.1 配置node1 为ntp服务器 。

[root@node1 ~]# yum install -y chrony [root@node1 ~]# vim /etc/chrony.conf server 0.centos.pool.ntp.org iburst allow 192.168.4.0/24 local stratum 10 [root@node1 ~]# systemctl enable chronyd [root@node1 ~]# systemctl restart chronyd

配置客户机的ntp， 指定ntp服务器为node1 [root@node2 ~]# vim /etc/chrony.conf server 192.168.4.1 iburst [root@node2 ~]# for i in awk -F’ ’ ’{ print $3 }’ /etc/hosts ; do scp /etc/chrony.conf *i* : */etc/*; *ssh*i systemctl restart chronyd; ntpdate 192.168.4.1; done

2.2 安装ceph 服务器 。

1、在node1上安装部署软件

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

2、创建ceph部署工具的工作目录

```
[root@node1 ~]# mkdir  ceph-clu
```

3、创建参与集群节点的配置文件

```
[root@node1 ceph-clu]# ceph-deploy  new  node{1..3}
[root@node1 ceph-clu]# ls
```

4、在3个节点上安装软件包

```
[root@node1 ceph-clu]# ceph-deploy  install  node{1..3}
```

5、初始化mon服务

```
[root@node1 ceph-clu]# ceph-deploy  mon  create-initial
```

如果出现以下错误：

```
[node1][ERROR ] admin_socket: exception getting command descriptions: [Errno 2] No such file or directory
```

解决方案：

```
[root@node1 ceph-clu]# vim ceph.conf  最下面加入行：
public_network = 192.168.4.0/24
```

再执行以下命令覆盖配置文件：

```
[root@host1 ceoh-clu]# ceph-deploy --overwrite-conf config push node1 node2 node3
```

然后执行 [root@node1 ceph-clu]# ceph-deploy mon create-initial

6、把node1-3的vdb作为日志盘。Ext / xfs都是日志文件系统，一个分区分成日志区和数据区。为了更好的性能，vdb专门作为vdc和vdd的日志盘。

```
[root@node1 ceph-clu]# for vm in node{1..3}
> do
> ssh $vm parted /dev/vdb mklabel gpt
> done
[root@node1 ceph-clu]# for vm in node{1..3}; do ssh $vm parted /dev/vdb mkpart primary 1M 50% ;
done
[root@node1 ceph-clu]# for vm in node{1..3}; do ssh $vm parted /dev/vdb mkpart primary 50% 100% ;
done
[root@node1 ceph-clu]# for vm in node{1..3}; do ssh ${vm} chown ceph.ceph /dev/vdb? ; done
```

编写UDEV规则，使得vdb1和vdb2重启后，属主属组仍然是ceph

```
[root@node1 ~]# vim /etc/udev/rules.d/90-cephdisk.rules
ACTION=="add", KERNEL=="vdb[12]", OWNER="ceph", GROUP="ceph"
```

7、创建OSD设备

```
[root@node1 ceph-clu]# for i in {1..3}
> do
> ceph-deploy disk zap node$i:vdc node$i:vdd
> done
[root@node1 ceph-clu]# for i in {1..3}
> do
> ceph-deploy osd create node$i:vdc:/dev/vdb1 node$i:vdd:/dev/vdb2
```

> done
8、验证
到第7步为止，ceph已经搭建完成。查看ceph状态
[root@node1 ceph-clu]# ceph -s 如果出现health HEALTH_OK表示正常

9、排错
https://www.zybuluo.com/dyj2017/note/920621

2.3 创建三种类型的存储。
2.3.1 创建RBD类型存储，创建clinet-img DBfile-img 镜像并挂载，制作快照跟克隆快照。

使用RBD(Rados块设备)
1、查看存储池
[root@node1 ~]# ceph osd lspools
可以查看到0号镜像池，名字为rbd
2、分别创建两个10GB镜像，名称分别为clinet-img、DBfile-img
[root@node1 ceph-clu]# rbd create clinet-img --image-feature layering --size 10G
[root@node1 ceph-clu]# rbd create DBfile-img --image-feature layering --size 10G
[root@node1 ~]# rbd list
[root@node1 ~]# rbd info clinet-img
[root@node1 ~]# rbd info DBfile-img

可以测试rbd池里面的镜像并测试扩容跟缩容操作。
创建第2个镜像，名为image，指定它位于rbd池中
[root@node1 ~]# rbd create rbd/image --image-feature layering --size 10G
将image镜像大小缩减为7G
[root@node1 ceph-clu]# rbd resize --size 7G image --allow-shrink
[root@node1 ceph-clu]# rbd info image
扩容image到15G
[root@node1 ceph-clu]# rbd resize --size 15G image
[root@node1 ceph-clu]# rbd info image

4、将dba-server作为客户端，使用ceph创建的镜像作为存储设备
(1)安装客户端软件
[root@dba-server ~]# yum install -y ceph-common
(2)拷贝相关文件
[root@node1 ceph-clu]# scp /etc/ceph/ceph.conf dba-server:/etc/ceph/
[root@node1 ceph-clu]# scp /etc/ceph/ceph.client.admin.keyring dba-server:/etc/ceph/

注：ceph.conf是配置文件，里面记录了ceph集群访问的方式和地址
ceph.client.admin.keyring是client.admin用户的密钥文件
(3)映射DBfile-img 镜像到本地
[root@dba-server ~]# rbd map DBfile-img
/dev/rbd1   ->rbd1就是映射出来的硬盘文件
[root@dba-server ~]# lsblk
[root@dba-server ~]# rbd showmapped
(4)格式化、挂载
[root@dba-server ~]# mkfs.ext4 /dev/rbd1
[root@dba-server ~]# mount /dev/rbd1 /var/lib/mysql/
[root@dba-server ~]# df -h /var/lib/mysql/
[root@dba-server ~]# echo 'hello world' > /var/lib/mysql/hello.txt

快照
1、查看image镜像的快照
[root@dba-server ~]# rbd snap ls DBfile-img
2、为image创建名为image-sn1的快照
[root@dba-server ~]# rbd snap create DBfile-img --snap DBfile-sn1
3、模拟误删除操作，恢复数据

（1）删除
[root@dba-server ~]# rm -f /var/lib/mysql/hello.txt
（2）卸载设备
[root@dba-server ~]# umount /var/lib/mysql/
（3）使用DBfile-sn1还原快照
[root@dba-server ~]# rbd snap rollback DBfile-img --snap DBfile-sn1
Rolling back to snapshot: 100% complete...done.
（4）挂载，查看是否已恢复
[root@dba-server ~]# mount /dev/rbd1 /var/lib/mysql/
[root@dba-server ~]# cat /var/lib/mysql/hello.txt
hello world

克隆快照
1、克隆快照，首先要把快照保护起来，防止误删除之类的操作
[root@dba-server ~]# rbd snap protect DBfile-img --snap DBfile-sn1
2、克隆image-sn1快照，克隆的名称是image-cl1
[root@dba-server ~]# rbd clone DBfile-img --snap DBfile-sn1 DBfile-cl1 --image-feature layering
3、查看状态
[root@dba-server ~]# rbd info DBfile-cl1
rbd image 'DBfile-cl1':
    size 10240 MB in 2560 objects
    order 22 (4096 kB objects)
    block_name_prefix: rbd_data.10483d1b58ba
    format: 2
    features: layering
    flags:
    parent: rbd/DBfile-img@DBfile-sn1
    overlap: 10240 MB
4、合并克隆文件
[root@dba-server ~]# rbd flatten DBfile-cl1
Image flatten: 100% complete...done.
[root@dba-server ~]# rbd info DBfile-cl1
rbd image 'DBfile-cl1':
    size 10240 MB in 2560 objects
    order 22 (4096 kB objects)
    block_name_prefix: rbd_data.10483d1b58ba
    format: 2
    features: layering
    flags:

    没有parent了
5、删除[需要的时候才用]
[root@dba-server ~]# umount /var/lib/mysql/
[root@dba-server ~]# rbd showmapped
id pool image   snap device
0 rbd clinet-img -   /dev/rbd0
1 rbd DBfile-img -   /dev/rbd1

[root@dba-server ~]# rbd unmap /dev/rbd/rbd/DBfile-img

无法删除快照，因为没有关闭快照保护
[root@dba-server ~]# rbd snap rm DBfile-img --snap DBfile-sn1
rbd: snapshot 'DBfile-sn1' is protected from removal.
2018-10-11 17:44:03.775649 7f6590f9cd80 -1 librbd::Operations: snapshot is protected
[root@dba-server ~]# rbd snap unprotect DBfile-img --snap DBfile-sn1
[root@dba-server ~]# rbd snap rm DBfile-img --snap DBfile-sn1

安装KVM虚拟机，使用ceph存储提供的镜像作为硬盘
1、将物理主机作为客户端，安装软件包，拷贝配置文件

```
[root@room00pc000 ~]# yum install ceph-common
[root@node1 ceph-clu]# scp /etc/ceph/ceph.c* 192.168.4.254:/etc/ceph/
```

2、正常创建一台KVM虚拟机，取名为myrhel7。向导结束之后，将其强制关机即可。

3、导出myrhel7虚拟的声明文件，将虚拟删掉。

```
[root@room00pc000 ~]# virsh dumpxml RGW-clinet > /tmp/myrhel7.xml
```

4、虚拟机使用CEPH存储，需要认证。方式是虚拟先生成secret，再将secret与CEPH账户映射

（1）编写账户信息文件

```
[root@room00pc000 ~]# cat /tmp/secret.xml
<secret ephemeral='no' private='no'>
      <usage type='ceph'>
            <name>client.admin secret</name>
      </usage>
</secret>
```

（2）生成secret

```
[root@room00pc000 ~]# virsh secret-define --file /tmp/secret.xml    #原因是已经定义过一次了。
错误：使用 /tmp/secret.xml 设定属性失败
错误：internal error: 已将 UUID 为0f7937cc-66d4-4d44-b1a2-962544bb88fd 的 secret 定义为与
client.admin secret 一同使用
l
[root@room00pc000 ~]# virsh secret-list       查看secret
UUID                                           用量

--------------------------------------------------------------------------------

0f7937cc-66d4-4d44-b1a2-962544bb88fd  ceph client.admin secret
```

6、将虚拟机软件的secret和ceph的管理员用户关联

（1）查看管理员的密钥

```
[root@room00pc000 ~]# cat /etc/ceph/ceph.client.admin.keyring
[client.admin]
    key = AQC4Bb9be4cYOBAAI1dHo/o61sCkW4Ht3cG33w==
```

（2）关联secret和ceph的管理员

```
[root@room00pc000 ~]# virsh secret-set-value --secret 0f7937cc-66d4-4d44-b1a2-962544bb88fd  --
base64  AQC4Bb9be4cYOBAAI1dHo/o61sCkW4Ht3cG33w==
secret 值设定
```

7、修改虚拟机的配置文件/tmp/myrhel7.xml，把管理员信息写到该文件中，并指定虚拟机磁盘使用ceph的镜像

```
[root@room00pc000 ~]# cat /tmp/myrhel7.xml  原来的配置文件
<domain type='kvm'>
   <name>RGW-clinet</name>
   <uuid>899f2482-c077-4d04-9ce6-6d743c762afa</uuid>
   <devices>
      <emulator>/usr/libexec/qemu-kvm</emulator>
      <disk type='file' device='disk'>
         <driver name='qemu' type='qcow2'/>
         <source file='/var/lib/libvirt/images/RGW-clinet.qcow2'/>
         <target dev='vda' bus='virtio'/>
         <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
      </disk>

[root@room00pc000 ~]# cat /tmp/myrhel7.xml  改过后的配置文件
<domain type='kvm'>
   <name>RGW-clinet</name>
   <uuid>899f2482-c077-4d04-9ce6-6d743c762afa</uuid>
   <devices>
      <disk type='network' device='disk'>
```

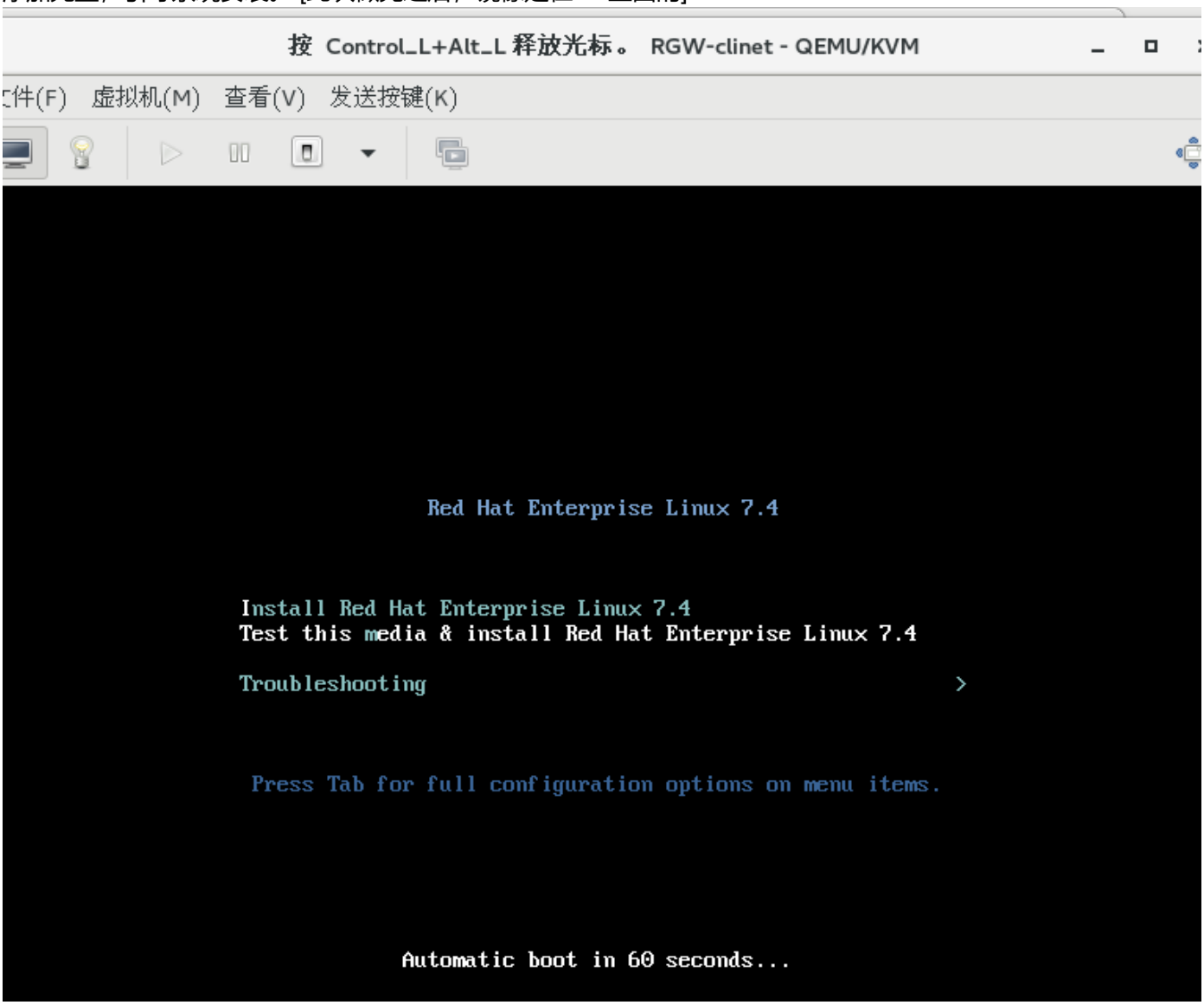
```
        <driver name='qemu' type='raw'/>
        <auth username='admin'>
           <secret type='ceph' uuid='0f7937cc-66d4-4d44-b1a2-962544bb88fd'/>
        </auth>
        <source protocol='rbd' name='rbd/clinet-img'>
           <host name='192.168.4.1' port='6789'/>
        </source>
        <target dev='vda' bus='virtio'/>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
```

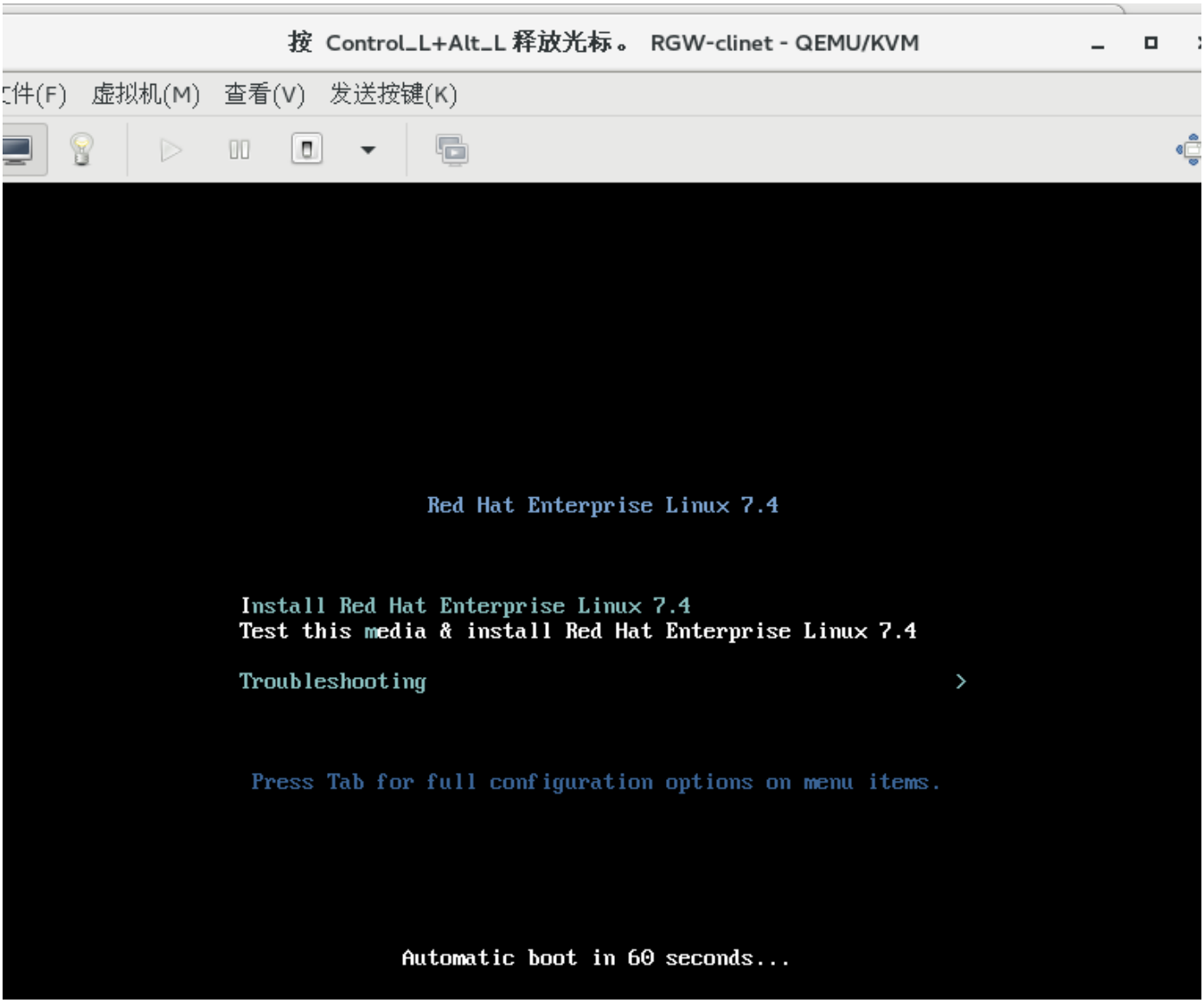
8、利用xml文件生成虚拟机

```
[root@room00pc000 ~]# virsh define /tmp/myrhel7.xml
定义域 RGW-clinet（从 /tmp/myrhel7.xml）
```

9、启动虚拟机时，不能直接安装，需要在虚拟机设置中连接光盘文件，并且设置启动选项，将光盘设置为第一启动介质。

添加光盘，引导系统安装。 [此次做完之后，镜像是在rbd上面的]





2.3.2 安装ceph-fs， 提供给到web服务器。

cephFS
它可以像NFS或SAMBA那样，提供共享文件夹，客户端通过挂载目录的方式使用CEPH的存储。
1、cephFS需要一台MDS元数据服务器node4

2、在node4上安装软件包
```
[root@node4 ~]# yum install -y ceph-mds
```

3、在node1上配置node4为mds服务器
```
[root@node1 ~]# cd ceph-clu/
[root@node1 ceph-clu]# ceph-deploy mds create node4
```

4、将管理密钥同步到mds服务器
```
[root@node1 ceph-clu]# ceph-deploy admin node4
```

5、创建两个池，一个名为cephfs_webfile用于存储数据，一个名为cephfs_metawebfile 用于存储元数据
```
[root@node1 ceph-clu]# ceph osd pool create cephfs_webfile 128
[root@node1 ceph-clu]# ceph osd pool create cephfs_metawebfile 128
```
128表示PG的数目是128。PG是规置组，文件存到PG中，PG存在池中。

6、创建名为webfile的文件系统
```
[root@node1 ceph-clu]# ceph fs new webfile cephfs_metawebfile cephfs_webfile
```

7、查看状态
```
[root@node1 ceph-clu]# ceph mds stat
[root@node1 ceph-clu]# ceph fs ls
```

8、客户端使用（web1 2 分别挂载）
```
[root@web-server1 ~]# mkdir /root/webfile-dir
[root@web-server2 ~]# mkdir /root/webfile-dir
[root@node1 ~]# cat /etc/ceph/ceph.client.admin.keyring
[root@web-server1 ~]# mount -t ceph 192.168.4.1:6789:/ /root/webfile-dir/  -o
name=admin,secret=AQC4Bb9be4cYOBAAI1dHo/o61sCkW4Ht3cG33w==
[root@web-server1 ~]#  df -h  /root/webfile-dir
文件系统                 容量  已用  可用 已用% 挂载点
192.168.4.1:6789:/    60G  4.5G    56G     8% /root/webfile-dir

[root@web-server2 ~]# tail -1  /etc/fstab       #可以直接挂也可以设置开机自动挂载
192.168.4.1:6789:/    /root/webfile-dir    ceph
name=admin,secret=AQC4Bb9be4cYOBAAI1dHo/o61sCkW4Ht3cG33w==,_netdev  0 0
[root@web-server2 ~]# mount -a
[root@web-server2 ~]# df -h /root/webfile-dir
文件系统                 容量  已用  可用 已用% 挂载点
192.168.4.1:6789:/    60G  4.5G    56G     8% /root/webfile-dir
```

2.3.3 安装RGW，给到RGW-clinet。

Ceph对象存储
1、使用ceph对象存储，需要RGW，即ceph网关。配置node5是RGW

2、可以在管理节点上为node5进行安装和配置
```
[root@node1 ceph-clu]# ceph-deploy install --rgw node5
```

3、将配置文件、密钥文件同步到node5
```
[root@node1 ceph-clu]# ceph-deploy admin node5
```

4、在node5上启动rgw服务
```
[root@node1 ceph-clu]# ceph-deploy rgw create node5
```

5、rgw内建了一个名为civetweb的web服务器，这个服务器与apache/nginx类似。既然是web服务，所以使用80端口更为方便。
为了应用简便，可以将默认的7480端口改为80端口
```
[root@node5 ~]# vim /etc/ceph/ceph.conf 追加以下内容
[client.rgw.node5]
host = node5
rgw_frontends = "civetweb port=80"
[root@node5 ~]# systemctl restart ceph-radosgw\*
```

6、验证端口是否已经修改成功
```
[root@RGW-clinet ~]# curl http://192.168.4.5
```

7、在RGW-clinet上安装客户端软件
```
[root@RGW-clinet ~]# rpm -ivh s3cmd-2.0.1-1.el7.noarch.rpm
```

8、创建ceph对象存储用户
```
[root@node5 ~]# radosgw-admin  user create --uid="guqiangjun" --display-name="haha guqiangjun"
{
    "user_id": "guqiangjun",
    "display_name": "haha guqiangjun",
    "email": "",
    "suspended": 0,
    "max_buckets": 1000,
    "auid": 0,
    "subusers": [],
    "keys": [
        {
```

```
                "user": "guqiangjun",
                "access_key": "DPZTI2AX3AR698GCJXJI",
                "secret_key": "iiX1Awl4E7hplksB7NBYzuHxd7GCyUNGxr6gTBxy"
            }
     ],
     "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access_key和secret_key

9、配置s3客户端

```
[root@RGW-clinet ~]# s3cmd --configure

Enter new values or accept defaults in brackets with Enter.
Refer to user manual for detailed description of all options.

Access key and Secret key are your identifiers for Amazon S3. Leave them empty for using the env
variables.
Access Key: DPZTI2AX3AR698GCJXJI
Secret Key: iiX1Awl4E7hplksB7NBYzuHxd7GCyUNGxr6gTBxy
Default Region [US]:

Use "s3.amazonaws.com" for S3 Endpoint and not modify it to the target Amazon S3.
S3 Endpoint [s3.amazonaws.com]: 192.168.4.5

Use "%(bucket)s.s3.amazonaws.com" to the target Amazon S3. "%(bucket)s" and "%(location)s" vars
can be used
if the target S3 system supports dns based buckets.
DNS-style bucket+hostname:port template for accessing a bucket [%(bucket)s.s3.amazonaws.com]: %
(bucket)s.192.168.4.5

Encryption password is used to protect your files from reading
by unauthorized persons while in transfer to S3
Encryption password:
Path to GPG program [/usr/bin/gpg]:

When using secure HTTPS protocol all communication with Amazon S3
servers is protected from 3rd party eavesdropping. This method is
slower than plain HTTP, and can only be proxied with Python 2.7 or newer
Use HTTPS protocol [Yes]: no

On some networks all internet access must go through a HTTP proxy.
Try setting it here if you can't connect to S3 directly
HTTP Proxy server name:

New settings:
```

```
  Access Key: DPZTI2AX3AR698GCJXJI
  Secret Key: iiX1Awl4E7hplksB7NBYzuHxd7GCyUNGxr6gTBxy
  Default Region: US
  S3 Endpoint: 192.168.4.5
  DNS-style bucket+hostname:port template for accessing a bucket: %(bucket)s.192.168.4.5
  Encryption password:
  Path to GPG program: /usr/bin/gpg
  Use HTTPS protocol: False
  HTTP Proxy server name:
  HTTP Proxy server port: 0

Test access with supplied credentials? [Y/n] y
Please wait, attempting to list all buckets...
Success. Your access key and secret key worked fine :-)

Now verifying that encryption works...
Not configured. Never mind.

Save settings? [y/N] y
Configuration saved to '/root/.s3cfg'


10、客户端测试
[root@RGW-clinet ~]# s3cmd ls       查看内容

创建一个bucket，相当于是文件夹，名称要求为xxx_yyy格式
[root@RGW-clinet ~]# s3cmd  mb s3://RGWfile_clinet
Bucket 's3://RGWfile_clinet/' created
[root@RGW-clinet ~]# s3cmd ls
2018-10-12 09:52  s3://RGWfile_clinet

上传文件
[root@RGW-clinet ~]# s3cmd put /etc/hosts s3://RGWfile_clinet
upload: '/etc/hosts' -> 's3://RGWfile_clinet/hosts'   [1 of 1]
 158 of 158    100% in     5s     29.84 B/s  done
[root@RGW-clinet ~]# s3cmd ls s3://RGWfile_clinet
2018-10-12 09:53          158    s3://RGWfile_clinet/hosts

下载文件到/tmp，下载后的文件改名为zhuji
[root@RGW-clinet ~]# s3cmd get s3://RGWfile_clinet/hosts  /tmp/hosts
download: 's3://RGWfile_clinet/hosts' -> '/tmp/hosts'   [1 of 1]
 158 of 158    100% in     0s     19.90 kB/s  done
[root@RGW-clinet ~]# cat /tmp/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1             localhost localhost.localdomain localhost6 localhost6.localdomain6

删除文件
[root@RGW-clinet ~]# s3cmd del s3://RGWfile_clinet/hosts
delete: 's3://RGWfile_clinet/hosts'
[root@RGW-clinet ~]# s3cmd ls s3://RGWfile_clinet
[root@RGW-clinet ~]#
```

3.配置lvs+ha 服务器，使其能够调用后端web服务器。

4.配置nginx， mysql， 使用ceph 存储挂载到两台web跟dba， 安装论坛 。

5.几种error 解决方法

此报错是因为node1-node3没有安装软件包 重新运行 ceph-deploy install node{1..3}

```
[root@node1 ceph-clu]# ceph-deploy  mon  create-initial
[ceph_deploy.conf][DEBUG ] found configuration file at: /root/.cephdeploy.co
nf
[ceph_deploy.cli][INFO  ] Invoked (1.5.33): /usr/bin/ceph-deploy mon create-
initial
[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  ]  subcommand                    : create-initial
[ceph_deploy.cli][INFO  ]  quiet                         : False
[ceph_deploy.cli][INFO  ]  cd_conf                       : <ceph_deploy.con
f.cephdeploy.Conf instance at 0x7f15c001d200>
[ceph_deploy.cli][INFO  ]  cluster                       : ceph
[ceph_deploy.cli][INFO  ]  func                          : <function mon at
 0x7f15c0013938>
[ceph_deploy.cli][INFO  ]  ceph_conf                     : None
[ceph_deploy.cli][INFO  ]  default_release               : False
[ceph_deploy.cli][INFO  ]  keyrings                      : None
[ceph_deploy.mon][DEBUG ] Deploying mon, cluster ceph hosts node1 node2 node
3
[ceph_deploy.mon][DEBUG ] detecting platform for host node1 ...
[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
[ceph_deploy.mon][ERROR ] ceph needs to be installed in remote host: node1
[ceph_deploy.mon][DEBUG ] detecting platform for host node2 ...
[node2][DEBUG ] connected to host: node2
[node2][DEBUG ] detect platform information from remote host
[node2][DEBUG ] detect machine type
[node2][DEBUG ] find the location of an executable
[ceph_deploy.mon][ERROR ] ceph needs to be installed in remote host: node2
[ceph_deploy.mon][DEBUG ] detecting platform for host node3 ...
[node3][DEBUG ] connected to host: node3
[node3][DEBUG ] detect platform information from remote host
[node3][DEBUG ] detect machine type
[node3][DEBUG ] find the location of an executable
[ceph_deploy.mon][ERROR ] ceph needs to be installed in remote host: node3
[ceph_deploy][ERROR ] GenericError: Failed to create 3 monitors
```

此错误可以vim ceph.conf 最下面加入行:

```
 public_network = 192.168.4.0/24
```

再执行以下命令：

```
ceph-deploy --overwrite-conf config push node1 node2 node3
```

```
 [root@node1 ceph-clu]# ceph-deploy  mon  create-initial
[ceph_deploy.conf][DEBUG ] found configuration file at: /root/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (1.5.33): /usr/bin/ceph-deploy mon create-initial
[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  ]  subcommand                    : create-initial
[ceph_deploy.cli][INFO  ]  quiet                         : False
[ceph_deploy.cli][INFO  ]  cd_conf                       :
<ceph_deploy.conf.cephdeploy.Conf instance at 0x28496c8>
[ceph_deploy.cli][INFO  ]  cluster                       : ceph
[ceph_deploy.cli][INFO  ]  func                          : <function mon at
0x283f938>
[ceph_deploy.cli][INFO  ]  ceph_conf                     : None
[ceph_deploy.cli][INFO  ]  default_release               : False
[ceph_deploy.cli][INFO  ]  keyrings                      : None
[ceph_deploy.mon][DEBUG ] Deploying mon, cluster ceph hosts node1 node2 node3
[ceph_deploy.mon][DEBUG ] detecting platform for host node1 ...
[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
[ceph_deploy.mon][INFO  ] distro info: Red Hat Enterprise Linux Server 7.4 Maipo
[node1][DEBUG ] determining if provided host has same hostname in remote
[node1][DEBUG ] get remote short hostname
[node1][DEBUG ] deploying mon to node1
[node1][DEBUG ] get remote short hostname
[node1][DEBUG ] remote hostname: node1
[node1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
[node1][DEBUG ] create the mon path if it does not exist
[node1][DEBUG ] checking for done path: /var/lib/ceph/mon/ceph-node1/done
[node1][DEBUG ] done path does not exist: /var/lib/ceph/mon/ceph-node1/done
[node1][INFO  ] creating keyring file: /var/lib/ceph/tmp/ceph-node1.mon.keyring
[node1][DEBUG ] create the monitor keyring file
[node1][INFO  ] Running command: ceph-mon --cluster ceph --mkfs -i node1 --keyring
/var/lib/ceph/tmp/ceph-node1.mon.keyring --setuser 167 --setgroup 167
[node1][DEBUG ] ceph-mon: mon.noname-a 192.168.4.1:6789/0 is local, renaming to
mon.node1
[node1][DEBUG ] ceph-mon: set fsid to ff979247-1a38-409a-905a-96f8031faf4e
[node1][DEBUG ] ceph-mon: created monfs at /var/lib/ceph/mon/ceph-node1 for
mon.node1
[node1][INFO  ] unlinking keyring file /var/lib/ceph/tmp/ceph-node1.mon.keyring
[node1][DEBUG ] create a done file to avoid re-doing the mon deployment
[node1][DEBUG ] create the init path if it does not exist
[node1][INFO  ] Running command: systemctl enable ceph.target
[node1][INFO  ] Running command: systemctl enable ceph-mon@node1
[node1][WARNIN] Created symlink from /etc/systemd/system/ceph-mon.target.wants/ceph-
mon@node1.service to /usr/lib/systemd/system/ceph-mon@.service.
[node1][INFO  ] Running command: systemctl start ceph-mon@node1
[node1][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
```

```
/var/run/ceph/ceph-mon.node1.asok mon_status
[node1][DEBUG ]
********************************************************************************
[node1][DEBUG ] status for monitor: mon.node1
[node1][DEBUG ] {
[node1][DEBUG ]   "election_epoch": 0,
[node1][DEBUG ]   "extra_probe_peers": [
[node1][DEBUG ]     "192.168.4.2:6789/0",
[node1][DEBUG ]     "192.168.4.3:6789/0"
[node1][DEBUG ]   ],
[node1][DEBUG ]   "monmap": {
[node1][DEBUG ]     "created": "2018-09-27 17:16:49.856465",
[node1][DEBUG ]     "epoch": 0,
[node1][DEBUG ]     "fsid": "ff979247-1a38-409a-905a-96f8031faf4e",
[node1][DEBUG ]     "modified": "2018-09-27 17:16:49.856465",
[node1][DEBUG ]     "mons": [
[node1][DEBUG ]       {
[node1][DEBUG ]         "addr": "192.168.4.1:6789/0",
[node1][DEBUG ]         "name": "node1",
[node1][DEBUG ]         "rank": 0
[node1][DEBUG ]       },
[node1][DEBUG ]       {
[node1][DEBUG ]         "addr": "0.0.0.0:0/1",
[node1][DEBUG ]         "name": "node2",
[node1][DEBUG ]         "rank": 1
[node1][DEBUG ]       },
[node1][DEBUG ]       {
[node1][DEBUG ]         "addr": "0.0.0.0:0/2",
[node1][DEBUG ]         "name": "node3",
[node1][DEBUG ]         "rank": 2
[node1][DEBUG ]       }
[node1][DEBUG ]     ]
[node1][DEBUG ]   },
[node1][DEBUG ]   "name": "node1",
[node1][DEBUG ]   "outside_quorum": [
[node1][DEBUG ]     "node1"
[node1][DEBUG ]   ],
[node1][DEBUG ]   "quorum": [],
[node1][DEBUG ]   "rank": 0,
[node1][DEBUG ]   "state": "probing",
[node1][DEBUG ]   "sync_provider": []
[node1][DEBUG ] }
[node1][DEBUG ]
********************************************************************************
[node1][INFO  ] monitor: mon.node1 is running
[node1][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node1.asok mon_status
[ceph_deploy.mon][DEBUG ] detecting platform for host node2 ...
[node2][DEBUG ] connected to host: node2
[node2][DEBUG ] detect platform information from remote host
[node2][DEBUG ] detect machine type
[node2][DEBUG ] find the location of an executable
```

```
[ceph_deploy.mon][INFO  ] distro info: Red Hat Enterprise Linux Server 7.4 Maipo
[node2][DEBUG ] determining if provided host has same hostname in remote
[node2][DEBUG ] get remote short hostname
[node2][WARNIN]
********************************************************************************
[node2][WARNIN] provided hostname must match remote hostname
[node2][WARNIN] provided hostname: node2
[node2][WARNIN] remote hostname: localhost
[node2][WARNIN] monitors may not reach quorum and create-keys will not complete
[node2][WARNIN]
********************************************************************************
[node2][DEBUG ] deploying mon to node2
[node2][DEBUG ] get remote short hostname
[node2][DEBUG ] remote hostname: localhost
[node2][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
[node2][DEBUG ] create the mon path if it does not exist
[node2][DEBUG ] checking for done path: /var/lib/ceph/mon/ceph-localhost/done
[node2][DEBUG ] done path does not exist: /var/lib/ceph/mon/ceph-localhost/done
[node2][INFO  ] creating keyring file: /var/lib/ceph/tmp/ceph-localhost.mon.keyring
[node2][DEBUG ] create the monitor keyring file
[node2][INFO  ] Running command: ceph-mon --cluster ceph --mkfs -i localhost --
keyring /var/lib/ceph/tmp/ceph-localhost.mon.keyring --setuser 167 --setgroup 167
[node2][DEBUG ] ceph-mon: mon.noname-b 192.168.4.2:6789/0 is local, renaming to
mon.localhost
[node2][DEBUG ] ceph-mon: set fsid to ff979247-1a38-409a-905a-96f8031faf4e
[node2][DEBUG ] ceph-mon: created monfs at /var/lib/ceph/mon/ceph-localhost for
mon.localhost
[node2][INFO  ] unlinking keyring file /var/lib/ceph/tmp/ceph-localhost.mon.keyring
[node2][DEBUG ] create a done file to avoid re-doing the mon deployment
[node2][DEBUG ] create the init path if it does not exist
[node2][INFO  ] Running command: systemctl enable ceph.target
[node2][INFO  ] Running command: systemctl enable ceph-mon@localhost
[node2][WARNIN] Created symlink from /etc/systemd/system/ceph-mon.target.wants/ceph-
mon@localhost.service to /usr/lib/systemd/system/ceph-mon@.service.
[node2][INFO  ] Running command: systemctl start ceph-mon@localhost
[node2][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node2.asok mon_status
[node2][ERROR ] admin_socket: exception getting command descriptions: [Errno 2] No
such file or directory
[node2][WARNIN] monitor: mon.node2, might not be running yet
[node2][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node2.asok mon_status
[node2][ERROR ] admin_socket: exception getting command descriptions: [Errno 2] No
such file or directory
[node2][WARNIN] monitor node2 does not exist in monmap
[node2][WARNIN] neither `public_addr` nor `public_network` keys are defined for
monitors
[node2][WARNIN] monitors may not be able to form quorum
[ceph_deploy.mon][DEBUG ] detecting platform for host node3 ...
[node3][DEBUG ] connected to host: node3
[node3][DEBUG ] detect platform information from remote host
[node3][DEBUG ] detect machine type
```

```
[node3][DEBUG ] find the location of an executable
[ceph_deploy.mon][INFO  ] distro info: Red Hat Enterprise Linux Server 7.4 Maipo
[node3][DEBUG ] determining if provided host has same hostname in remote
[node3][DEBUG ] get remote short hostname
[node3][DEBUG ] deploying mon to node3
[node3][DEBUG ] get remote short hostname
[node3][DEBUG ] remote hostname: node3
[node3][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
[node3][DEBUG ] create the mon path if it does not exist
[node3][DEBUG ] checking for done path: /var/lib/ceph/mon/ceph-node3/done
[node3][DEBUG ] done path does not exist: /var/lib/ceph/mon/ceph-node3/done
[node3][INFO  ] creating keyring file: /var/lib/ceph/tmp/ceph-node3.mon.keyring
[node3][DEBUG ] create the monitor keyring file
[node3][INFO  ] Running command: ceph-mon --cluster ceph --mkfs -i node3 --keyring
/var/lib/ceph/tmp/ceph-node3.mon.keyring --setuser 167 --setgroup 167
[node3][DEBUG ] ceph-mon: mon.noname-c 192.168.4.3:6789/0 is local, renaming to
mon.node3
[node3][DEBUG ] ceph-mon: set fsid to ff979247-1a38-409a-905a-96f8031faf4e
[node3][DEBUG ] ceph-mon: created monfs at /var/lib/ceph/mon/ceph-node3 for
mon.node3
[node3][INFO  ] unlinking keyring file /var/lib/ceph/tmp/ceph-node3.mon.keyring
[node3][DEBUG ] create a done file to avoid re-doing the mon deployment
[node3][DEBUG ] create the init path if it does not exist
[node3][INFO  ] Running command: systemctl enable ceph.target
[node3][INFO  ] Running command: systemctl enable ceph-mon@node3
[node3][WARNIN] Created symlink from /etc/systemd/system/ceph-mon.target.wants/ceph-
mon@node3.service to /usr/lib/systemd/system/ceph-mon@.service.
[node3][INFO  ] Running command: systemctl start ceph-mon@node3
[node3][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node3.asok mon_status
[node3][DEBUG ]
********************************************************************************
[node3][DEBUG ] status for monitor: mon.node3
[node3][DEBUG ] {
[node3][DEBUG ]   "election_epoch": 1,
[node3][DEBUG ]   "extra_probe_peers": [
[node3][DEBUG ]     "192.168.4.1:6789/0",
[node3][DEBUG ]     "192.168.4.2:6789/0"
[node3][DEBUG ]   ],
[node3][DEBUG ]   "monmap": {
[node3][DEBUG ]     "created": "2018-09-27 17:16:56.677241",
[node3][DEBUG ]     "epoch": 0,
[node3][DEBUG ]     "fsid": "ff979247-1a38-409a-905a-96f8031faf4e",
[node3][DEBUG ]     "modified": "2018-09-27 17:16:56.677241",
[node3][DEBUG ]     "mons": [
[node3][DEBUG ]       {
[node3][DEBUG ]         "addr": "192.168.4.1:6789/0",
[node3][DEBUG ]         "name": "node1",
[node3][DEBUG ]         "rank": 0
[node3][DEBUG ]       },
[node3][DEBUG ]       {
[node3][DEBUG ]         "addr": "192.168.4.3:6789/0",
```

```
[node3][DEBUG ]         "name": "node3",
[node3][DEBUG ]         "rank": 1
[node3][DEBUG ]       },
[node3][DEBUG ]       {
[node3][DEBUG ]         "addr": "0.0.0.0:0/2",
[node3][DEBUG ]         "name": "node2",
[node3][DEBUG ]         "rank": 2
[node3][DEBUG ]       }
[node3][DEBUG ]     ]
[node3][DEBUG ]   },
[node3][DEBUG ]   "name": "node3",
[node3][DEBUG ]   "outside_quorum": [],
[node3][DEBUG ]   "quorum": [],
[node3][DEBUG ]   "rank": 1,
[node3][DEBUG ]   "state": "electing",
[node3][DEBUG ]   "sync_provider": []
[node3][DEBUG ] }
[node3][DEBUG ]
********************************************************************************
[node3][INFO  ] monitor: mon.node3 is running
[node3][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node3.asok mon_status
[ceph_deploy.mon][INFO  ] processing monitor mon.node1
[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: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node1.asok mon_status
[ceph_deploy.mon][WARNIN] mon.node1 monitor is not yet in quorum, tries left: 5
[ceph_deploy.mon][WARNIN] waiting 5 seconds before retrying
[node1][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node1.asok mon_status
[ceph_deploy.mon][WARNIN] mon.node1 monitor is not yet in quorum, tries left: 4
[ceph_deploy.mon][WARNIN] waiting 10 seconds before retrying
[node1][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node1.asok mon_status
[ceph_deploy.mon][WARNIN] mon.node1 monitor is not yet in quorum, tries left: 3
[ceph_deploy.mon][WARNIN] waiting 10 seconds before retrying
[node1][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node1.asok mon_status
[ceph_deploy.mon][WARNIN] mon.node1 monitor is not yet in quorum, tries left: 2
[ceph_deploy.mon][WARNIN] waiting 15 seconds before retrying
[node1][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node1.asok mon_status
[ceph_deploy.mon][WARNIN] mon.node1 monitor is not yet in quorum, tries left: 1
[ceph_deploy.mon][WARNIN] waiting 20 seconds before retrying
[ceph_deploy.mon][INFO  ] processing monitor mon.node2
[node2][DEBUG ] connected to host: node2
[node2][DEBUG ] detect platform information from remote host
[node2][DEBUG ] detect machine type
[node2][DEBUG ] find the location of an executable
```

```
[node2][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node2.asok mon_status
[node2][ERROR ] admin_socket: exception getting command descriptions: [Errno 2] No
such file or directory
[ceph_deploy.mon][WARNIN] mon.node2 monitor is not yet in quorum, tries left: 5
[ceph_deploy.mon][WARNIN] waiting 5 seconds before retrying
[node2][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node2.asok mon_status
[node2][ERROR ] admin_socket: exception getting command descriptions: [Errno 2] No
such file or directory
[ceph_deploy.mon][WARNIN] mon.node2 monitor is not yet in quorum, tries left: 4
[ceph_deploy.mon][WARNIN] waiting 10 seconds before retrying
[node2][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node2.asok mon_status
[node2][ERROR ] admin_socket: exception getting command descriptions: [Errno 2] No
such file or directory
[ceph_deploy.mon][WARNIN] mon.node2 monitor is not yet in quorum, tries left: 3
[ceph_deploy.mon][WARNIN] waiting 10 seconds before retrying
[node2][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node2.asok mon_status
[node2][ERROR ] admin_socket: exception getting command descriptions: [Errno 2] No
such file or directory
[ceph_deploy.mon][WARNIN] mon.node2 monitor is not yet in quorum, tries left: 2
[ceph_deploy.mon][WARNIN] waiting 15 seconds before retrying
[node2][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node2.asok mon_status
[node2][ERROR ] admin_socket: exception getting command descriptions: [Errno 2] No
such file or directory
[ceph_deploy.mon][WARNIN] mon.node2 monitor is not yet in quorum, tries left: 1
[ceph_deploy.mon][WARNIN] waiting 20 seconds before retrying
[ceph_deploy.mon][INFO  ] processing monitor mon.node3
[node3][DEBUG ] connected to host: node3
[node3][DEBUG ] detect platform information from remote host
[node3][DEBUG ] detect machine type
[node3][DEBUG ] find the location of an executable
[node3][INFO  ] Running command: ceph --cluster=ceph --admin-daemon
/var/run/ceph/ceph-mon.node3.asok mon_status
[ceph_deploy.mon][INFO  ] mon.node3 monitor has reached quorum!
[ceph_deploy.mon][ERROR ] Some monitors have still not reached quorum:
[ceph_deploy.mon][ERROR ] node1
[ceph_deploy.mon][ERROR ] node2
```

ceph-deploy mon create-initial 之后还是提示文件不一致需要覆盖，可以删掉有问题的机器的 /var/run/ceph/* 然后重启之后再执行一次，如果还不行的话，使用 ceph-deploy --overwrite-conf mon create node2 node3 node1 代替 ceph-deploy mon create-initial 也行

```
 [root@node1 ceph-clu]# ceph-deploy  mon  create-initial
[ceph_deploy.conf][DEBUG ] found configuration file at: /root/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (1.5.33): /usr/bin/ceph-deploy mon create-initial
[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  ]  subcommand                    : create-initial
[ceph_deploy.cli][INFO  ]  quiet                         : False
[ceph_deploy.cli][INFO  ]  cd_conf                       :
<ceph_deploy.conf.cephdeploy.Conf instance at 0x1bc06c8>
[ceph_deploy.cli][INFO  ]  cluster                       : ceph
[ceph_deploy.cli][INFO  ]  func                          : <function mon at
0x1bb6938>
[ceph_deploy.cli][INFO  ]  ceph_conf                     : None
[ceph_deploy.cli][INFO  ]  default_release               : False
[ceph_deploy.cli][INFO  ]  keyrings                      : None
[ceph_deploy.mon][DEBUG ] Deploying mon, cluster ceph hosts node1 node2 node3
[ceph_deploy.mon][DEBUG ] detecting platform for host node1 ...
[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
[ceph_deploy.mon][INFO  ] distro info: Red Hat Enterprise Linux Server 7.4 Maipo
[node1][DEBUG ] determining if provided host has same hostname in remote
[node1][DEBUG ] get remote short hostname
[node1][DEBUG ] deploying mon to node1
[node1][DEBUG ] get remote short hostname
[node1][DEBUG ] remote hostname: node1
[node1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
[ceph_deploy.mon][ERROR ] RuntimeError: config file /etc/ceph/ceph.conf exists with
different content; use --overwrite-conf to overwrite
[ceph_deploy.mon][DEBUG ] detecting platform for host node2 ...
[node2][DEBUG ] connected to host: node2
[node2][DEBUG ] detect platform information from remote host
[node2][DEBUG ] detect machine type
[node2][DEBUG ] find the location of an executable
[ceph_deploy.mon][INFO  ] distro info: Red Hat Enterprise Linux Server 7.4 Maipo
[node2][DEBUG ] determining if provided host has same hostname in remote
[node2][DEBUG ] get remote short hostname
[node2][WARNIN]
********************************************************************************
[node2][WARNIN] provided hostname must match remote hostname
[node2][WARNIN] provided hostname: node2
[node2][WARNIN] remote hostname: localhost
[node2][WARNIN] monitors may not reach quorum and create-keys will not complete
[node2][WARNIN]
********************************************************************************
[node2][DEBUG ] deploying mon to node2
[node2][DEBUG ] get remote short hostname
[node2][DEBUG ] remote hostname: localhost
```

```
[node2][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
[ceph_deploy.mon][ERROR ] RuntimeError: config file /etc/ceph/ceph.conf exists with
different content; use --overwrite-conf to overwrite
[ceph_deploy.mon][DEBUG ] detecting platform for host node3 ...
[node3][DEBUG ] connected to host: node3
[node3][DEBUG ] detect platform information from remote host
[node3][DEBUG ] detect machine type
[node3][DEBUG ] find the location of an executable
[ceph_deploy.mon][INFO  ] distro info: Red Hat Enterprise Linux Server 7.4 Maipo
[node3][DEBUG ] determining if provided host has same hostname in remote
[node3][DEBUG ] get remote short hostname
[node3][DEBUG ] deploying mon to node3
[node3][DEBUG ] get remote short hostname
[node3][DEBUG ] remote hostname: node3
[node3][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
[ceph_deploy.mon][ERROR ] RuntimeError: config file /etc/ceph/ceph.conf exists with
different content; use --overwrite-conf to overwrite
[ceph_deploy][ERROR ] GenericError: Failed to create 3 monitors
```