手工方式创建ceph集群

乔建峰 麻烦大体看下，现在这个过程就是能够实现基本功能的ceph集群

系统版本
系统版本：centos6.6
内核版本：2.6.32-504.el6.x86_64

ceph集群拓扑说明

IP地址	节点名称	功能角色
192.168.5.11	node1	mon&osd
192.168.5.12	node2	mon&osd
192.168.5.13	node3	mon&osd

---

ceph集群部署前提，在所有节点上进行如下配置

1.关闭各个节点主机的iptables和SELinux，在本地配置启动ntp服务来保证ceph集群主机之间的时间同步

2.添加hosts文件实现集群内主机与主机之间可以通过名称相互能够解析

在各个node主机上编辑host配置文件，添加如下内容
```
# vim /etc/hosts
>>
192.168.5.11 node1

192.168.5.12 node2

192.168.5.13 node3
<<
## 此处应该保证hostname应该与此处的设置的节点名称应该相同
```

3.将node1的公钥复制到其他各个节点，实现node1无密码登录其他各个节点

```
# yum install -y openssh-server    ## 安装openssh-server

# ssh-keygen -t rsa    ## 在本地生成ssh密钥

# for no in {1..3}; do ssh-copy-id -i /root/.ssh/id_rsa.pub root@node${no};done    ## 将node1的公钥复制到其他各个节点

# for no in {1..3}; do ssh node${no} hostname; done    ## 使用for循环显示远程主机名称验证无密码登录是否配置成功
```

4. 获取ceph的域名源的key文件
```
#rpm --import 'https://download.ceph.com/keys/release.asc'
#rpm --import 'https://download.ceph.com/keys/autobuild.asc'
```

5. 添加yum的repo文件

5-1. base源的repo文件

#vim/etc/yum.repos.d/CentOS-Base.repo

```
[base]
name=CentOS-$releasever - Base - mirrors.aliyun.com
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/$releasever/os/$basearch/
http://mirrors.aliyuncs.com/centos/$releasever/os/$basearch/
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
gpgcheck=1
gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-6

#released updates
[updates]
name=CentOS-$releasever - Updates - mirrors.aliyun.com
```

```
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/$releasever/updates/$basearch/
http://mirrors.aliyuncs.com/centos/$releasever/updates/$basearch/
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates
gpgcheck=1
gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-6

#additional packages that may be useful
[extras]
name=CentOS-$releasever - Extras - mirrors.aliyun.com
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/$releasever/extras/$basearch/
http://mirrors.aliyuncs.com/centos/$releasever/extras/$basearch/
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras
gpgcheck=1
gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-6

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus - mirrors.aliyun.com
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/$releasever/centosplus/$basearch/
http://mirrors.aliyuncs.com/centos/$releasever/centosplus/$basearch/
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus
gpgcheck=1
enabled=0
gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-6

#contrib - packages by Centos Users
[contrib]
name=CentOS-$releasever - Contrib - mirrors.aliyun.com
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/$releasever/contrib/$basearch/
http://mirrors.aliyuncs.com/centos/$releasever/contrib/$basearch/
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=contrib
gpgcheck=1
enabled=0
gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-6
```

5-2. epel源的repo文件

```
#/etc/yum.repos.d/epel.repo

[epel]
name=Extra Packages for Enterprise Linux 6 - $basearch
baseurl=http://mirrors.aliyun.com/epel/6/$basearch
http://mirrors.aliyuncs.com/epel/6/$basearch
#mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-6&arch=$basearch
failovermethod=priority
enabled=1
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6

[epel-debuginfo]
name=Extra Packages for Enterprise Linux 6 - $basearch - Debug
baseurl=http://mirrors.aliyun.com/epel/6/$basearch/debug
http://mirrors.aliyuncs.com/epel/6/$basearch/debug
#mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-6&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
gpgcheck=0

[epel-source]
name=Extra Packages for Enterprise Linux 6 - $basearch - Source
baseurl=http://mirrors.aliyun.com/epel/6/SRPMS
http://mirrors.aliyuncs.com/epel/6/SRPMS
#mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-source-6&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
gpgcheck=0
```

5-3. ceph源的repo文件

#vim/etc/yum.repos.d/ceph.repo ## repo文件中指定的ceph版本为hammer版本, 系统版本为rhel6版本,
如使用其他版本请对应修改baseurl字段中的对应字段
[ceph]
name=Ceph packages for $basearch
baseurl=http://download.ceph.com/rpm-hammer/rhel6/$basearch
enabled=1
priority=2
gpgcheck=1
type=rpm-md
gpgkey=https://download.ceph.com/keys/release.asc

[ceph-noarch]
name=Ceph noarch packages
baseurl=http://download.ceph.com/rpm-hammer/rhel6/noarch
enabled=1
priority=2
gpgcheck=1
type=rpm-md
gpgkey=https://download.ceph.com/keys/release.asc

[ceph-source]
name=Ceph source packages
baseurl=http://download.ceph.com/rpm-hammer/rhel6/SRPMS
enabled=0
priority=2
gpgcheck=1
type=rpm-md
gpgkey=https://download.ceph.com/keys/release.asc

6. 安装ceph存储集群所用程序包
6-1. 安装ceph依赖程序包
#yum install yum-plugin-priorities
#yum install -y snappy leveldb gdisk python-argparse gperftools-libs

6-2. 安装ceph软件
#yum install -y ceph

部署第一个mon节点

1. 登录监控节点node1节点
[root@node1 ~]# ls /etc/ceph #查看ceph配置文件目录是否有东西

2. 创建ceph配置文件并配置ceph配置文件内的内容
[root@node1 ~]# touch /etc/ceph/ceph.conf #创建一个ceph配置文件
[root@node1 ~]# uuidgen #执行此命令可以得到一个唯一的标识. 作为ceph集群ID
f11240d4-86b1-49ba-aacc-6d3d37b24cc4

按下面的内容编辑ceph配置文件
[root@node1 ~]# vi /etc/ceph/ceph.conf
[global]
fsid = f11240d4-86b1-49ba-aacc-6d3d37b24cc4
mon initial members = node1,node2,node3
mon host = 192.168.5.11,192.168.5.12,192.168.5.13
public network = 192.168.5.0/24
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd journal size = 1024
filestore xattr use omap = true
osd pool default size = 3
osd pool default min size = 1
osd crush chooseleaf type = 1
osd_mkfs_type = xfs
max mds = 5

```
mds max file size = 100000000000000
mds cache size = 1000000
mon osd down out interval = 900 ## 设置osd节点down后900s. 把此osd节点逐出ceph集群. 把之前映射到此节点的数据映射到其他节点.
cluster_network = 192.168.5.0/24
[mon]
mon clock drift allowed = .50 ## 把时钟偏移设置成0.5s. 默认是0.05s.由于ceph集群中存在异构PC. 导致时钟偏移总是大于0.05s.
为了方便同步直接把时钟偏移设置成0.5s
```

3. 在node1创建各种密钥

```
[root@node1 ~]# ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --cap mon 'allow *' ##
为监控节点创建管理密钥
[root@node1 ~]# ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n client.admin --set-uid=0 --cap
mon 'allow *' --cap osd 'allow *' --cap mds 'allow' ## 为ceph amin用户创建管理集群的密钥并赋予访问权限
[root@node1 ~]# ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring ## 添加client.admin
key到 ceph.mon.keyring
```

4. 在node1监控节点创建一个mon数据目录

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

5. 在node1创建一个boot引导启动osd的key

```
[root@node1 ~]# mkdir -p /var/lib/ceph/bootstrap-osd/
[root@node1 ~]# ceph-authtool -C /var/lib/ceph/bootstrap-osd/ceph.keyring
```

6. 在node1节点上初始化mon节点. 执行下面的命令

```
[root@node1 ~]# ceph-mon --mkfs -i node1 --keyring /tmp/ceph.mon.keyring
```

7. 为了防止重新被安装创建一个空的done文件

```
[root@node1 ~]# touch /var/lib/ceph/mon/ceph-node1/done
```

8. 创建一个空的初始化文件

```
[root@node1 ~]# touch /var/lib/ceph/mon/ceph-node1/sysvinit
```

9. 启动ceph进程

```
[root@node1 ~]# /sbin/service ceph -c /etc/ceph/ceph.conf start mon.node1
其他方式? /etc/init.d/ceph start mon.node1
```

10. 查看asok mon状态

```
[root@node1 ~]# ceph --cluster=ceph --admin-daemon /var/run/ceph/ceph-mon.node1.asok mon_status
```

==========

部署第二个mon节点

1. 复制node1节点的/etc/ceph目录到node2

```
[root@node1 ~]# scp /etc/ceph/* node2:/etc/ceph/
```

2. 在node2节点上新建一个/var/lib/ceph/bootstrap-osd/目录

```
[root@node2 ~]# mkdir /var/lib/ceph/bootstrap-osd/
```

3. 复制node1节点上的/var/lib/ceph/bootstrap-osd/ceph.keyring文件到node2

```
[root@node1 ~]# scp /var/lib/ceph/bootstrap-osd/ceph.keyring node2:/var/lib/ceph/bootstrap-osd/
```

4. 复制node1节点上的/tmp/ceph.mon.keyring

```
[root@node1 ~]# scp /tmp/ceph.mon.keyring node2:/tmp/
```

5. 在node2节点上建立一个/var/lib/ceph/mon/ceph-node2目录

```
[root@node2 ~]# mkdir -p /var/lib/ceph/mon/ceph-node2
```

6. 在node2节点上初始化mon节点，执行下面的命令

```
[root@node2 ~]# ceph-mon --mkfs -i node2 --keyring /tmp/ceph.mon.keyring
```

7. 为了防止重新被安装创建一个空的done文件

```
[root@node2 ~]# touch /var/lib/ceph/mon/ceph-node2/done
```

8. 创建一个空的初始化文件

```
[root@node2 ~]# touch /var/lib/ceph/mon/ceph-node2/sysvinit
```

9. 启动ceph进程

```
[root@node2 ~]# /sbin/service ceph -c /etc/ceph/ceph.conf start mon.node2
```

==========

部署第三个mon节点

1. 复制node1节点的/etc/ceph目录到node3
[root@node1 ~]# scp /etc/ceph/* node3:/etc/ceph/

2. 在node3节点上新建一个/var/lib/ceph/bootstrap-osd/目录
[root@node3 ~]# mkdir /var/lib/ceph/bootstrap-osd/

3. 复制node1节点上的/var/lib/ceph/bootstrap-osd/ceph.keyring文件到node3
[root@node1 ~]# scp /var/lib/ceph/bootstrap-osd/ceph.keyring node3:/var/lib/ceph/bootstrap-osd/

4. 复制node1节点上的/tmp/ceph.mon.keyring
[root@node1 ~]# scp /tmp/ceph.mon.keyring node3:/tmp/

5. 在node3节点上建立一个/var/lib/ceph/mon/ceph-node3目录
[root@node3 ~]# mkdir -p /var/lib/ceph/mon/ceph-node3

6. 在node3节点上初始化mon节点，执行下面的命令
[root@node3 ~]# ceph-mon --mkfs -i node3 --keyring /tmp/ceph.mon.keyring

7. 为了防止重新被安装创建一个空的done文件
[root@node3 ~]# touch /var/lib/ceph/mon/ceph-node3/done

8. 创建一个空的初始化文件
[root@node3 ~]# touch /var/lib/ceph/mon/ceph-node3/sysvinit

9. 启动ceph进程
[root@node3 ~]# /sbin/service ceph -c /etc/ceph/ceph.conf start mon.node3

配置第一个OSD
1. 创建一个OSD，生成一个osd number
[root@node1 ~]# ceph osd create
0

2. 为osd节点创建一个osd目录
[root@node1 ~]# mkdir -p /var/lib/ceph/osd/ceph-0

3. 格式化已准备好的osd硬盘（格式化为xfs格式）
[root@node1 ~]# mkfs.xfs -f /dev/vdb
meta-data=/dev/vdb isize=256 agcount=4, agsize=1310720 blks
= sectsz=512 attr=2, projid32bit=0
data = bsize=4096 blocks=5242880, imaxpct=25
= sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0
log =internal log bsize=4096 blocks=2560, version=2
= sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0

4. 挂载目录，并查看挂载的情况
[root@node1 ~]# mount /dev/vdb /var/lib/ceph/osd/ceph-0
[root@node1 ~]# mount -o remount,user_xattr /var/lib/ceph/osd/ceph-0

[root@node1 ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
vmware-vmblock on /var/run/vmblock-fuse type fuse.vmware-vmblock (rw,nosuid,nodev,default_permissions,allow_other)
/dev/vdb on /var/lib/ceph/osd/ceph-1 type xfs (rw,user_xattr)

把上面的挂载信息写入分区表
[root@node1 ~]# vi /etc/fstab
>>
/dev/vdb /var/lib/ceph/osd/ceph-0 xfs defaults 0 0
/dev/vdb /var/lib/ceph/osd/ceph-0 xfs remount,user_xattr 0 0
<<

5. 初始化osd数据目录
[root@node1 ~]# ceph-osd -i 0 --mkfs --mkkey
2016-01-22 00:29:25.152226 7faada50d800 -1 journal FileJournal::_open: disabling aio for non-block journal. Use journal_force_aio

```
to force use of aio anyway
2016-01-22 00:29:25.241871 7faada50d800 -1 journal FileJournal::_open: disabling aio for non-block journal. Use journal_force_aio
to force use of aio anyway
2016-01-22 00:29:25.244493 7faada50d800 -1 filestore(/var/lib/ceph/osd/ceph-0) could not find 23c2fcde/osd_superblock/0//-1 in
index: (2) No such file or directory
2016-01-22 00:29:25.395252 7faada50d800 -1 created object store /var/lib/ceph/osd/ceph-0 journal /var/lib/ceph/osd/ceph-0/journal
for osd.0 fsid 9a2bc392-bd9d-42a2-a428-718b5eb52c6d
2016-01-22 00:29:25.395348 7faada50d800 -1 auth: error reading file: /var/lib/ceph/osd/ceph-0/keyring: can't open
/var/lib/ceph/osd/ceph-0/keyring: (2) No such file or directory
2016-01-22 00:29:25.395472 7faada50d800 -1 created new key in keyring /var/lib/ceph/osd/ceph-0/keyring
```

6. 注册osd的认证密钥

```
[root@node1 ~]# ceph auth add osd.0 osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-0/keyring
```

7. 为此osd节点创建一个crush map

```
[root@node1 ~]# ceph osd crush add-bucket node1 host
added bucket node1 type host to crush map
```

8. 将该osd节点作为默认节点

```
[root@node1 ~]# ceph osd crush move node1 root=default
moved item id -2 name 'node1' to location {root=default} in crush map
```

9. 使用crush算法来添加osd节点

```
[root@node1 ~]# ceph osd crush add osd.0 1.0 host=node1
add item id 0 name 'osd.0' weight 1 at location {host=node1} to crush map
```

10. 创建一个初始化目录

```
[root@node1 ~]# touch /var/lib/ceph/osd/ceph-0/sysvinit
```

11. 启动osd进程

```
/etc/init.d/ceph start osd.0
```

12. 查看osd目录树

```
[root@node1 ~]# ceph osd tree
# id weight type name up/down reweight
-1 1 root default
-2 1 host node1
0 1 osd.0 up 1
```

==========

创建第二个osd节点

1. 创建一个OSD，生成一个osd number

```
[root@node2 ~]# ceph osd create
1
```

2. 为osd节点创建一个osd目录

```
[root@node2 ~]# mkdir -p /var/lib/ceph/osd/ceph-1
```

3. 格式化已准备好的osd硬盘，并挂在到上一步创建的osd目录（格式化为xfs格式）

```
[root@node2 ~]# mkfs.xfs -f /dev/vdb
meta-data=/dev/vdb isize=256 agcount=4, agsize=1310720 blks
= sectsz=512 attr=2, projid32bit=0
data = bsize=4096 blocks=5242880, imaxpct=25
= sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0
log =internal log bsize=4096 blocks=2560, version=2
= sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
```

4. 挂在目录

```
[root@node2 ~]# mount /dev/vdb /var/lib/ceph/osd/ceph-1
[root@node2 ~]# mount -o remount,user_xattr /var/lib/ceph/osd/ceph-1
```

查看挂载的情况

```
[root@node2 ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
```

```
vmware-vmblock on /var/run/vmblock-fuse type fuse.vmware-vmblock (rw,nosuid,nodev,default_permissions,allow_other)
/dev/vdb on /var/lib/ceph/osd/ceph-1 type xfs (rw,user_xattr)
```

把上面的挂载信息写入分区表

```
[root@node2 ~]# vi /etc/fstab
>>
/dev/vdb /var/lib/ceph/osd/ceph-1 xfs defaults 0 0
/dev/vdb /var/lib/ceph/osd/ceph-1 xfs remount,user_xattr 0 0
<<
```

5. 初始化osd数据目录

```
[root@node2 ~]# ceph-osd -i 1 --mkfs --mkkey
2014-06-25 23:17:37.633040 7fa8fd06b7a0 -1 journal FileJournal::_open: disabling aio for non-block journal. Use journal_force_aio to force use of aio anyway
2014-06-25 23:17:37.740713 7fa8fd06b7a0 -1 journal FileJournal::_open: disabling aio for non-block journal. Use journal_force_aio to force use of aio anyway
2014-06-25 23:17:37.744937 7fa8fd06b7a0 -1 filestore(/var/lib/ceph/osd/ceph-1) could not find 23c2fcde/osd_superblock/0//-1 in index: (2) No such file or directory
2014-06-25 23:17:37.812999 7fa8fd06b7a0 -1 created object store /var/lib/ceph/osd/ceph-1 journal /var/lib/ceph/osd/ceph-1/journal for osd.1 fsid f11240d4-86b1-49ba-aacc-6d3d37b24cc4
2014-06-25 23:17:37.813192 7fa8fd06b7a0 -1 auth: error reading file: /var/lib/ceph/osd/ceph-1/keyring: can't open /var/lib/ceph/osd/ceph-1/keyring: (2) No such file or directory
2014-06-25 23:17:37.814050 7fa8fd06b7a0 -1 created new key in keyring /var/lib/ceph/osd/ceph-1/keyring
```

6. 注册osd的认证密钥

```
[root@node2 ~]# ceph auth add osd.1 osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-1/keyring
added key for osd.1
```

7. 为此osd节点创建一个crush map

```
[root@node2 ~]# ceph osd crush add-bucket node2 host
added bucket node2 type host to crush map
```

8. 将该osd节点作为默认节点

```
[root@node2 ~]# ceph osd crush move node2 root=default
moved item id -3 name 'node2' to location {root=default} in crush map
```

9. 使用crush算法来添加osd节点

```
[root@node2 ~]# ceph osd crush add osd.1 1.0 host=node2
add item id 1 name 'osd.1' weight 1 at location {host=node2} to crush map
```

10. 创建一个初始化目录

```
[root@node2 ~]# touch /var/lib/ceph/osd/ceph-1/sysvinit
```

11. 启动osd进程

```
[root@node2 ~]# /etc/init.d/ceph start osd.1
=== osd.1 ===
create-or-move updated item name 'osd.1' weight 0.02 at location {host=node2,root=default} to crush map
Starting Ceph osd.1 on node2...
starting osd.1 at :/0 osd_data /var/lib/ceph/osd/ceph-1 /var/lib/ceph/osd/ceph-1/journal
```

12. 查看osd目录树

```
[root@node2 ~]# ceph osd tree
# id weight type name up/down reweight
-1 2 root default
-2 1 host node1
0 1 osd.0 up 1
-3 1 host node2
1 1 osd.1 up 1
```

==========

创建第三个osd节点

1. 创建一个OSD，生成一个osd number

```
[root@node3 ~]# ceph osd create
```

2. 为osd节点创建一个osd目录

```
[root@node3 ~]# mkdir -p /var/lib/ceph/osd/ceph-2
```

3. 格式化已准备好的osd硬盘（格式化为xfs格式）

```
[root@node3 ~]# mkfs.xfs -f /dev/vdb
meta-data=/dev/vdb isize=256 agcount=4, agsize=1310720 blks
= sectsz=512 attr=2, projid32bit=0
```

```
data = bsize=4096 blocks=5242880, imaxpct=25
= sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0
log =internal log bsize=4096 blocks=2560, version=2
= sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
```

4. 挂在目录

```
[root@node3 ~]# mount /dev/vdb /var/lib/ceph/osd/ceph-2
[root@node3 ~]# mount -o remount,user_xattr /var/lib/ceph/osd/ceph-2
```

查看挂载的情况

```
[root@node2 ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
vmware-vmblock on /var/run/vmblock-fuse type fuse.vmware-vmblock (rw,nosuid,nodev,default_permissions,allow_other)
/dev/vdb on /var/lib/ceph/osd/ceph-1 type xfs (rw,user_xattr)
```

把上面的挂载信息写入分区表

```
[root@node3 ~]# vi /etc/fstab
/dev/vdb /var/lib/ceph/osd/ceph-2 xfs defaults 0 0
/dev/vdb /var/lib/ceph/osd/ceph-2 xfs remount,user_xattr 0 0
```

5. 初始化osd数据目录

```
[root@node3 ~]# ceph-osd -i 2 --mkfs --mkkey
2014-06-25 23:29:01.734251 7f52915927a0 -1 journal FileJournal::_open: disabling aio for non-block journal. Use journal_force_aio
to force use of aio anyway
2014-06-25 23:29:01.849158 7f52915927a0 -1 journal FileJournal::_open: disabling aio for non-block journal. Use journal_force_aio
to force use of aio anyway
2014-06-25 23:29:01.852189 7f52915927a0 -1 filestore(/var/lib/ceph/osd/ceph-2) could not find 23c2fcde/osd_superblock/0//-1 in
index: (2) No such file or directory
2014-06-25 23:29:01.904476 7f52915927a0 -1 created object store /var/lib/ceph/osd/ceph-2 journal /var/lib/ceph/osd/ceph-2/journal
for osd.2 fsid f11240d4-86b1-49ba-aacc-6d3d37b24cc4
2014-06-25 23:29:01.904712 7f52915927a0 -1 auth: error reading file: /var/lib/ceph/osd/ceph-2/keyring: can't open
/var/lib/ceph/osd/ceph-2/keyring: (2) No such file or directory
2014-06-25 23:29:01.905376 7f52915927a0 -1 created new key in keyring /var/lib/ceph/osd/ceph-2/keyring
[root@node3 ~]#
```

6. 注册osd的认证密钥

```
[root@node3 ~]# ceph auth add osd.2 osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-2/keyring
added key for osd.2
```

7. 为此osd节点创建一个crush map

```
[root@node3 ~]# ceph osd crush add-bucket node3 host
added bucket node3 type host to crush map
```

8. 将该osd节点作为默认节点

```
[root@node3 ~]# ceph osd crush move node3 root=default
moved item id -4 name 'node3' to location {root=default} in crush map
```

9. 使用crush算法来添加osd节点

```
[root@node3 ~]# ceph osd crush add osd.2 1.0 host=node3
add item id 2 name 'osd.2' weight 1 at location {host=node3} to crush map
```

10. 创建一个初始化目录

```
[root@node3 ~]# touch /var/lib/ceph/osd/ceph-2/sysvinit
```

11. 启动osd进程

```
[root@node3 ~]# /etc/init.d/ceph start osd.2
=== osd.2 ===
create-or-move updated item name 'osd.2' weight 0.02 at location {host=node3,root=default} to crush map
Starting Ceph osd.2 on node3...
starting osd.2 at :/0 osd_data /var/lib/ceph/osd/ceph-2 /var/lib/ceph/osd/ceph-2/journal
```

12. 查看osd目录树

```
[root@node3 ~]# ceph osd tree
# id weight type name up/down reweight
-1 3 root default
-2 1 host node1
```

```
0 1 osd.0 up 1
-3 1 host node2
1 1 osd.1 up 1
-4 1 host node3
2 1 osd.2 up 1
```

至此一个三节点的ceph基本功能集群已经部署完成，这三个节点在ceph集群中同时扮演monitor和osd的角色