1 RBD（Reliable Block Device）介绍

构建在RADOS集群之上为客户端提供块设备存储接口接口的中间层，提供的块存储服务可以形成一个裸磁盘，提供格式化、映射的功能，挂载到服务器中（这里服务器就是ceph的客用户端）。

RBD组件支持存储空间的动态扩容，也可以借助RADOS实现快照、副本和一致性。

客户端访问RBD的方式：

- 通过**内核模块rbd.ko**将块存储映射成本地的一块磁盘，例如/dev/vdbx等等，可以进行格式化和分区。
- 通过**librbd接口**，KVM虚拟化就是使用这种接口。

创建完RBD块存储类型的存储后，从块存储中映射出来的裸磁盘，如果想要被集群之外的其他服务器连接使用，该服务器需要具备以下三个条件：

- 该服务器需要安装RBD组件。
- 该服务器的网络需要与Ceph集群的Public网络互相通信。
- Ceph集群的keyring文件要拷贝到该服务器中。

Ceph集群的RBD块存储通常情况下是给虚拟化集群做底层存储使用，有专门的接口连接，如果想单独为某个服务器挂载块存储虚拟出来的磁盘，就需要先具备以上的三个条件。

单独服务器挂载块存储的磁盘一般都是通过内核级别进行挂载。

*RBD*存储是建立在*Pool*资源池中的，一个*Pool*资源池中可以创建多个*RBD*块存储设备。

1.1 RBD块存储数据写入流程

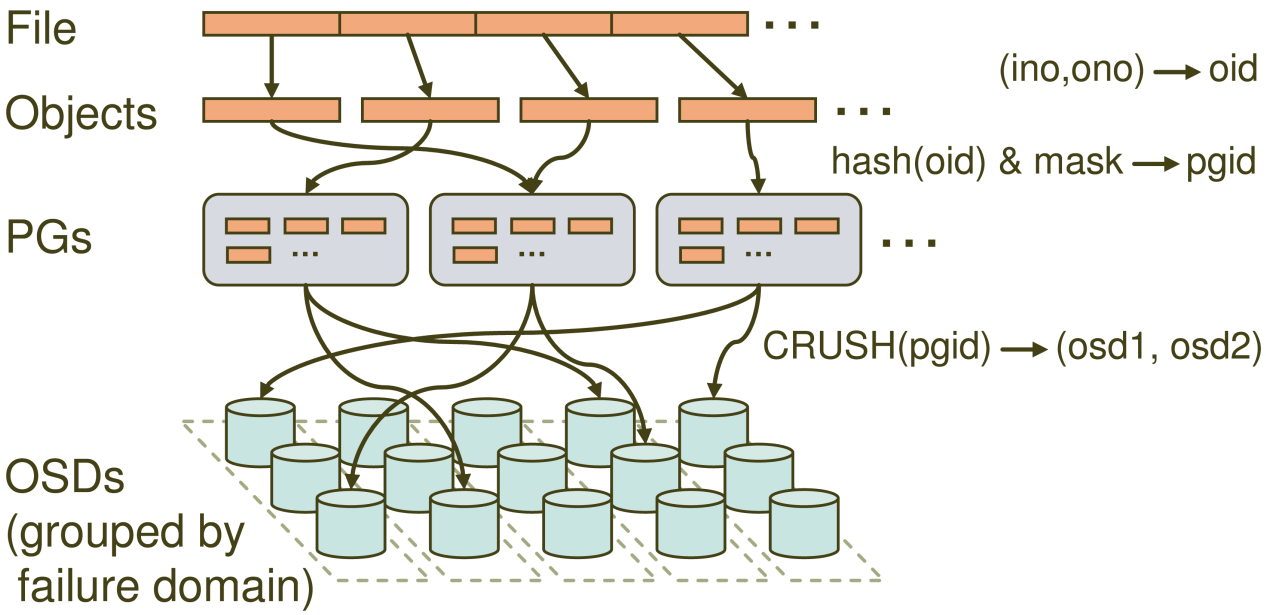
一个数据文件首先会被拆分成多个**Object对象**，

每一个Object对象文件都有一个**Oid**，

通过**Hash算法**将Oid进行计算，在通过Mask掩码计算出在Pool资源池中某个**PG的ID**，

拿到这个PGID后将**对象文件存储到PG中**，

最后在通过**GRUSH算法**将PG中的**Object对象写入到不同的OSD中**。



块存储本质就是将裸磁盘或类似裸磁盘(lvm)设备映射给主机使用，主机可以对其进行格式化并存储和读取数据，块设备读取速度快但是不支持共享。

2 在Ceph集群中创建RBD块存储并挂载到服务器

1）为RBD块存储创建一个Pool资源池。

2）基于Pool资源池创建出一个块存储。

3）如果是Centos7系统，使用内核挂载块设备存储时，对于默认创建的RBD块存储，有一些特性不支持，需要先将RBD中不支持特性禁用。

4）将RBD块存储映射成内核级别能挂载使用的裸磁盘。

5）格式化并使用。

RBD块存储映射成的磁盘是瘦分配模式，也就是说虽然映射的空间是那么大，比如说是10G，但是这10G不会立刻分配给客户端，而是客户端使用多少就给分配多少，直到达到限制。

2.1 RBD常用命令

命令	功能
rbd create	创建块设备映像
rbd ls	列出 rbd 存储池中的块设备
rbd info	查看块设备信息
rbd diff	可以统计 rbd 使用量
rbd map	映射块设备
rbd showmapped	查看已映射块设备
rbd remove	删除块设备
rbd resize	更改块设备的大小

2.2 为RBD块存储社保创建一个Pool资源池

创建pool资源池

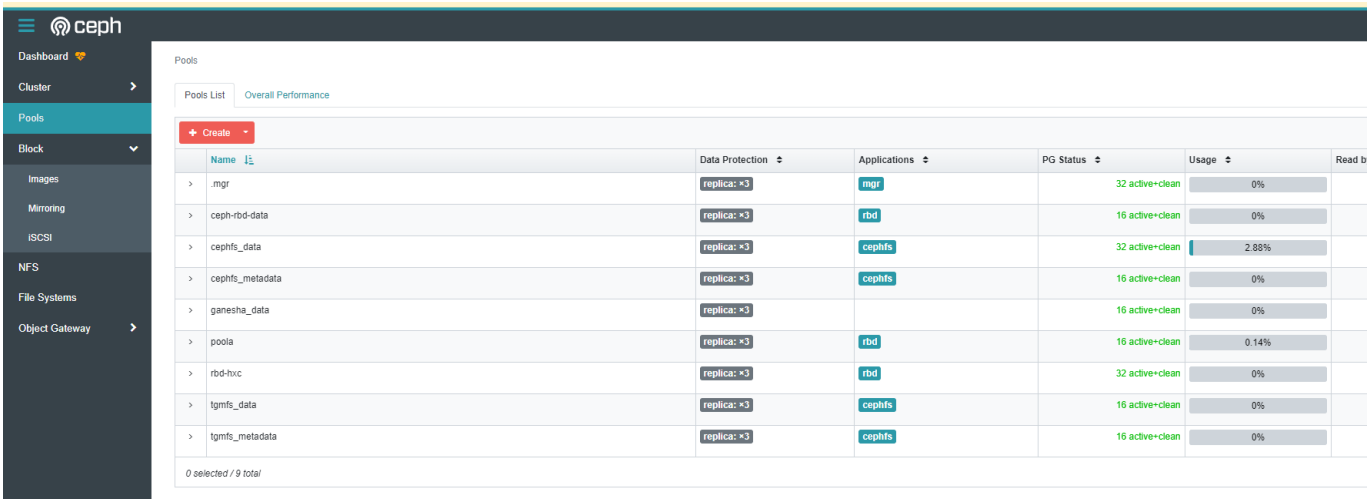
```
# 集群空间不够用了
[root@node1 ~]# ceph osd pool create rbd-hxc 128 128
Error ERANGE: pg_num 128 size 3 would mean 864 total pgs, which exceeds max 750
(mon_max_pg_per_osd 250 * num_in_osds 3)

# 32 32: PG数和PGP数(PG内包含的对象,待确定？？);PG会随着容量的增加也会动态的扩容;
# 生产上需要提前规划好数量
[root@node1 ~]# ceph osd pool create rbd-hxc 32 32
pool 'rbd-hxc' created

# 给rbd使用的pool标识成rbd，设置应用模式
[root@node1 ~]# ceph osd pool application enable rbd-hxc rbd
enabled application 'rbd' on pool 'rbd-hxc'

# 查看rbd的poll
[root@node1 ~]# ceph osd pool ls detail
pool 1 '.mgr' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins pg_num 32
pgp_num 32 autoscale_mode off last_change 153 lfor 0/0/151 flags hashpspool
stripe_width 0 pg_num_max 32 pg_num_min 1 application mgr
pool 5 'poola' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins pg_num
16 pgp_num 16 autoscale_mode off last_change 385 flags hashpspool stripe_width 0
application rbd
pool 7 'cephfs_data' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins
pg_num 32 pgp_num 32 autoscale_mode on last_change 982 lfor 0/0/980 flags hashpspool
stripe_width 0 application cephfs
pool 8 'cephfs_metadata' replicated size 3 min_size 2 crush_rule 0 object_hash
rjenkins pg_num 16 pgp_num 16 autoscale_mode on last_change 999 lfor 0/0/980 flags
hashpspool stripe_width 0 pg_autoscale_bias 4 pg_num_min 16 recovery_priority 5
application cephfs
pool 16 'tgmfs_data' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins
pg_num 16 pgp_num 16 autoscale_mode on last_change 1485 flags hashpspool stripe_width
0 application cephfs
pool 17 'tgmfs_metadata' replicated size 3 min_size 2 crush_rule 0 object_hash
rjenkins pg_num 16 pgp_num 16 autoscale_mode on last_change 1485 flags hashpspool
stripe_width 0 pg_autoscale_bias 4 pg_num_min 16 recovery_priority 5 application
cephfs
pool 18 'ganesha_data' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins
pg_num 16 pgp_num 16 autoscale_mode on last_change 1514 flags hashpspool stripe_width
0
pool 20 'ceph-rbd-data' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins
pg_num 16 pgp_num 16 autoscale_mode on last_change 1556 flags
hashpspool,selfmanaged_snaps stripe_width 0 application rbd
pool 21 'rbd-hxc' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins
pg_num 32 pgp_num 32 autoscale_mode on last_change 1562 flags hashpspool stripe_width
0 application rbd
# pool 21 是刚刚创建的，或者使用 ceph osd lspools 命令查看简略信息。
```

在dashboard中也可以进行查看，创建的pool信息



2.3 创建一个RBD块存储设备

命令格式：`rbd create -p {pool_name} --image {rbd_name} --size ${size}`

`-p`：指定pool资源池的名称。

`--image`：指定rbd块存储设备的名称。

`--size`：块存储的大小。

```
# 1.创建一个10G的rbd块设备
[root@node1 ~]# rbd create -p rbd-hxc --image rbd-hxc.img --size 10G

# 2.查看pool资源池中创建的rbd块设备
[root@node1 ~]# rbd -p rbd-hxc ls
rbd-hxc.img
# 同样此时在dashboard中，Block->Images中也可以看到rbd-hxc.img信息。

3.查看rbd块设备的信息
[root@node1 ~]# rbd info rbd-hxc/rbd-hxc.img
rbd image 'rbd-hxc.img': #块设备的名称
        size 10 GiB in 2560 objects #块设备的大小为10G，可以存放2560个对象文件，一个对象文件为4MB
        order 22 (4 MiB objects) #当前已经产生的对象文件数量
        snapshot_count: 0 #快照数量
        id: fc9b5bdd30776 #rbd块存储的id号
        block_name_prefix: rbd_data.fc9b5bdd30776 #块设备名称的前缀，所有的object文件命名中都会带有这个前缀
        format: 2
        features: layering, exclusive-lock, object-map, fast-diff, deep-flatten #rbd设备的特性，有部分不支持，需要禁用
        op_features:
        flags:
        create_timestamp: Tue Aug 29 13:52:18 2023
        access_timestamp: Tue Aug 29 13:52:18 2023
        modify_timestamp: Tue Aug 29 13:52:18 2023
```

2.4 通过内核级别挂载使用RBD块存储设备(存储服务器上面操作)

通过Linux内核将块设备映射成可挂载使用的磁盘设备。

```
[root@node1 ~]# rbd map rbd-hxc/rbd-hxc.img
/dev/rbd0
# centos7 这里可能会出现错误，需要禁用ceph rbd的一些特性，成功情况下，以下操作忽略

# 1、查看RBD块设备中有哪些特性
[root@node1 ~]# rbd -p rbd-hxc --image rbd-hxc.img info
rbd image 'rbd-hxc.img':
        size 10 GiB in 2560 objects
        order 22 (4 MiB objects)
        snapshot_count: 0
        id: fc9b5bdd30776
        block_name_prefix: rbd_data.fc9b5bdd30776
        format: 2
        features: layering, exclusive-lock, object-map, fast-diff, deep-flatten # 将
layering以后的特性全部禁用
        op_features:
        flags:
        create_timestamp: Tue Aug 29 13:52:18 2023
        access_timestamp: Tue Aug 29 13:52:18 2023
        modify_timestamp: Tue Aug 29 13:52:18 2023

2、禁用操作系统不支持的特性
[root@ceph-node-1 ~]# rbd feature disable rbd-hxc/rbd-hxc.img deep-flatten
[root@ceph-node-1 ~]# rbd feature disable rbd-hxc/rbd-hxc.img fast-diff
[root@ceph-node-1 ~]# rbd feature disable rbd-hxc/rbd-hxc.img object-map
[root@ceph-node-1 ~]# rbd feature disable rbd-hxc/rbd-hxc.img exclusive-lock
3、再次查看RBD块设备的特性
[root@node1 ~]# rbd -p rbd-hxc --image rbd-hxc.img info
rbd image 'rbd-hxc.img':
        size 10 GiB in 2560 objects
        order 22 (4 MiB objects)
        snapshot_count: 0
        id: fc9b5bdd30776
        block_name_prefix: rbd_data.fc9b5bdd30776
        format: 2
        features: layering # 仅剩下了layering
        op_features:
        flags:
        create_timestamp: Tue Aug 29 13:52:18 2023
        access_timestamp: Tue Aug 29 13:52:18 2023
        modify_timestamp: Tue Aug 29 13:52:18 2023
```

2.4.1 查看块设备映射的磁盘（存储服务器上面操作）

```
[root@node1 ~]# rbd device list
id  pool     namespace  image        snap  device
0   rbd-hxc             rbd-hxc.img  -     /dev/rbd0
# 可以看到映射了一块/dev/rbd0名称的磁盘

[root@node1 ~]# lsblk /dev/rbd0
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
rbd0 251:0    0  10G  0 disk


#取消映射，我们需要在客户电脑上面挂载
[root@node1 ~]# rbd unmap /dev/rbd0
[root@node1 ~]# lsblk /dev/rbd0
lsblk: /dev/rbd0: not a block device
```

2.5 创建访问用户

创建一个普通用户，linux使用普通用户访问pool存储池。

```
用户名称为：client.linux_mount  //用户命名遵循 <TYPE.ID> 的命名规则，其中 Type 有
mon,osd,client 三者，L 版本以后添加了 mgr 类型。
授权pool为rbd-hxc
```

```
#
[root@node1 ~]# ceph auth get-or-create client.linux_mount mon 'profile rbd' osd
'profile rbd pool=rbd-hxc' mgr 'profile rbd pool=rbd-hxc'
[client.linux_mount]
        key = AQCAme1kAZZIIhAAPMluaeAXiHXM/MWhVw3QMw==

# 查看用户信息
[root@node1 ~]# ceph auth ls
...
client.linux_mount
        key: AQCAme1kAZZIIhAAPMluaeAXiHXM/MWhVw3QMw==
        caps: [mgr] profile rbd pool=rbd-hxc
        caps: [mon] profile rbd
        caps: [osd] profile rbd pool=rbd-hxc
...
```

删除用户信息：ceph auth del {TYPE}.{ID}

导出用户信息文件。

```
[root@node1 ~]# ceph auth get-or-create client.linux_mount -o
ceph.client.linux_mount.keyring
[root@node1 ~]# ls
admin.secret  anaconda-ks.cfg  ceph.client.linux_mount.keyring  ceph.conf  ceph-env-
rpms  ceph.log  ceph-rpms  daemon.txt  mds.yaml
```

将用户的`ceph.client.linux_mount.keyring`文件和`ceph.conf`文件复制到客户端的`/etc/ceph`目录，**这个步骤可以等客户端安装完`ceph-common`**

2.6 客户端安装ceph工具

```
# 安装client端ceph工具
hxc@ubuntu:~$ sudo apt install -y python-setuptools ceph-common


#拷贝用户授权文件到client端
到ceph节点中拷贝ceph.client.linux_mount.keyring文件和ceph.conf，存储位置/etc/ceph/
或使用命令：
scp ceph.client.linux_mount.keyring root@192.168.31.73:/etc/ceph/
scp /etc/ceph/ceph.conf root@192.168.31.73:/etc/ceph/
```

拷贝完成后，我们就可以去客户端执行命令，检查是否可以获取到rbd信息。

```
hxc@ubuntu:~$ rbd list --id  linux_mount -p rbd-hxc
rbd-hxc.img # 和ceph服务器上面显示一致
```

2.7 客户端挂载

将远端的RBD文件映射到本地。

```
# root权限运行命令
hxc@ubuntu:~$ sudo rbd map rbd-hxc/rbd-hxc.img --id linux_mount
/dev/rbd0

# 查看rbd映射信息
hxc@ubuntu:~$ rbd showmapped
id  pool     namespace  image         snap  device
0   rbd-hxc             rbd-hxc.img   -     /dev/rbd0

# lsblk 命令查看
hxc@ubuntu:~$ lsblk /dev/rbd0
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
rbd0 252:0    0  10G  0 disk
```

格式化并挂载

```
hxc@ubuntu:~$ sudo mkfs.ext4 /dev/rbd0
mke2fs 1.45.5 (07-Jan-2020)
Discarding device blocks: done
Creating filesystem with 2621440 4k blocks and 655360 inodes
Filesystem UUID: 92ef024c-4d93-438d-910b-a0dd45230d1b
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inodetables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

mount到本地盘

```
hxc@ubuntu:~$ sudo mkdir /mnt/ceph-rbd
hxc@ubuntu:~$ sudo mount /dev/rbd0 /mnt/ceph-rbd/
```

查看挂载结果

```
hxc@ubuntu:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           388M   18M  371M   5% /run
/dev/sda5        98G   56G   37G  61% /
tmpfs           1.9G     0  1.9G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           1.9G     0  1.9G   0% /sys/fs/cgroup
...
/dev/rbd0       9.8G   24K  9.3G   1% /mnt/ceph-rbd
```

2.8 开机自动启动挂载

编辑/etc/ceph/rbdmap文件，设置自动map

```
echo "rbd-hxc/rbd-hxc.img
id=linux_mount,keyring=/etc/ceph/ceph.client.linux_mount.keyring " >>/etc/ceph/rbdmap
#rbd-hxc 为pool名称
#rbd-hxc.img 为rbd文件
#keyring 为之前同步的client key
#id 为rbd用户
```

设置rbdmap为开机启动

```
systemctl enable rbdmap
```

修改fstab，设置开机挂载

```
echo "/dev/rbd0  /mnt/ceph-rbd             ext4     defaults,noatime,_netdev 0 0"
>>/etc/fstab
```

重启客户端系统，验证开机挂载是否成功。

2.9 扩容

```
hxc@ubuntu:~$ sudo rbd resize rbd-hxc/rbd-hxc.img --id  linux_mount  --size 15G
[sudo] password for hxc:
Resizing image: 100% complete...done.

#resize2fs这个命令必须要执行，不然df -h显示的大小还是扩容前的容量，但是使用lsblk /dev/rbd0命令查看是扩容后的容量
hxc@ubuntu:~$ sudo resize2fs /dev/rbd0
resize2fs 1.45.5 (07-Jan-2020)
Filesystem at /dev/rbd0 is mounted on /mnt/ceph-rbd; on-line resizing required
old_desc_blocks = 2, new_desc_blocks = 2
The filesystem on /dev/rbd0 is now 3932160 (4k) blocks long.

hxc@ubuntu:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           388M   18M  371M   5% /run
/dev/sda5        98G   56G   37G  61% /
tmpfs           1.9G     0  1.9G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           1.9G     0  1.9G   0% /sys/fs/cgroup
...
/dev/rbd0        15G   24K   14G   1% /mnt/ceph-rbd # 15G 扩容后的容量了，使用dashboard查看也变成15G
```

2.10 删除创建的设备

```
hxc@ubuntu:~$ sudo umount /dev/rbd0
hxc@ubuntu:~$ sudo rbd unmap /dev/rbd0
hxc@ubuntu:~$ sudo rbd -p rbd-hxc rm --image rbd-hxc.img
Removing image: 100% complete...done.
```