参考

http://docs.ceph.org.cn/rbd/rbd-openstack/

安装rados工具包

```
# 在cinder-volume,nova-compute安装,python-rados,ceph-common

yum -y install python-rados,ceph-common
```

准备ceph环境

创建用户

```
ceph auth get-or-create client.cinder mon 'allow r' osd 'allow class-read
ceph auth get-or-create client.glance mon 'allow r' osd 'allow class-read
ceph auth get-or-create client.cinder-backup mon 'allow r' osd 'allow clas
```

分发keyring到指定节点

```
ceph auth get-or-create client.glance | ssh {your-glance-api-server} sudo
ceph auth get-or-create client.cinder | ssh {your-volume-server} sudo tee
ceph auth get-or-create client.cinder-backup | ssh {your-cinder-backup-ser
```

Cinder 对接

```
[ceph]
volume_driver = cinder.volume.drivers.rbd.RBDDriver
```

```
rbd_pool = volumes
rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_flatten_volume_from_snapshot = false
rbd_max_clone_depth = 5
rbd_store_chunk_size = 4
rados_connect_timeout = -1
glance_api_version = 2
rbd_user = cinder
rbd_secret_uuid = 457eb676-33da-42ec-9a8c-9293d545c337 # 参考后面为nova-com
```

Glance 对接

这里指的是Glance直接对接Ceph，而不是通过Cinder来对接Ceph。

```
# 配置glance-api.conf
[glance_store]
filesystem_store_datadir = /opt/stack/data/glance/images/
stores = rbd,file
rbd_store_pool = images
rbd_store_user = glance
rbd_store_ceph_conf = /etc/ceph/ceph.conf
rbd_store_chunk_size = 8
default_store = rbd
# 提供相应pool需要的keyring
```

存放在Ceph中的Image文件

```
[root@vmlqing-pike2 ~]# rbd ls images
3d6ef5aa-5410-4ce3-b24b-1dea48f061d4
```

镜像在Ceph中被切分成众多的Object来存储

```
[root@vmlqing-pike2 ~]# rados -p images ls
rbd_data.111f630e12923.0000000000000000
rbd_header.111f630e12923
rbd_directory
rbd_data.111f630e12923.0000000000000001
rbd_id.3d6ef5aa-5410-4ce3-b24b-1dea48f061d4
```

其中rbd_id.记录了镜像文件的指纹，上述例子即111f630e12923。

rbd_header.111f630e12923记录镜像的元数据信息，rbd_data.111f630e12923.0~N记录镜像的实际数据。

每个rbd_data的最大大小通过rbd_store_chunk_size指定。

创建镜像成功之后，ceph会自动为该镜像创建一个快照（快照是保护状态的）。创建这个快照的原因是因为，当Cinder从Image创建卷时会直接对该块照进行Clone（当然Cinder的后端也是同一个ceph集群）。
可以观察Image的location信息，实际上是指向快照的：
rbd://*fsid*/images/*images-id*/snap

```
[root@vmlqing-pike2 ~]# rbd info images/3d6ef5aa-5410-4ce3-b24b-1dea48f061d4@snap
rbd image '3d6ef5aa-5410-4ce3-b24b-1dea48f061d4':
        size 12957 kB in 2 objects
        order 23 (8192 kB objects)
        block_name_prefix: rbd_data.111f630e12923
        format: 2
        features: layering
        flags:
        protected: True
```

分析Cinder的rbd驱动会发现，实际上该驱动实现了cinder.volume.drivers.rbd.RBDDriver#clone_image接口，netapp、huawei之流都没有实现这个接口。
该接口会判断image能否直接Clone（cinder.volume.drivers.rbd.RBDDriver#_is_cloneable 如果：1.Cinder和glance对接同一个fsid；2.image是raw格式的；3.Cinder能够读ceph的image文件）。
注意这里创建出的卷虽然是Clone，但是似乎会被自动的执行_flatten操作，但是该操作没有体现在rbd的驱动代码里。（这里找了半天，发现rbd接口调用的rbd_clone3接口，这里发现有三个类似的接口rbd_clone、rbd_clone2、rbd_clone3，有什么区别？？？）

Nova 对接

为nova-compute节点上libvirt添加秘钥

libvirt需要为ceph配置秘钥，才能读写ceph提供的卷
参考：http://libvirt.org/formatsecret.html#CephUsageType

```
ceph auth get-key client.cinder | ssh {your-compute-node} tee client.cinde
```

```
uuidgen
457eb676-33da-42ec-9a8c-9293d545c337
```

```
cat > secret.xml <<EOF
<secret ephemeral='no' private='no'>
<uuid>457eb676-33da-42ec-9a8c-9293d545c337</uuid>
<usage type='ceph'>
<name>client.cinder secret</name>
</usage>
</secret>
EOF
sudo virsh secret-define --file secret.xml
Secret 457eb676-33da-42ec-9a8c-9293d545c337 created
sudo virsh secret-set-value --secret 457eb676-33da-42ec-9a8c-9293d545c337
```

Nova直接启动ceph中的镜像

这里指的是qemu和Ceph中Image直接交互，不通过Cinder。
启动虚机时如果不创建卷，那么Nova的系统盘不再存放在nova/instances目录中，转而存放在Ceph的RBD池中。

在nova的nova-compute.conf中配置以下参数（有的版本中n-cpu有独立的配置文件）

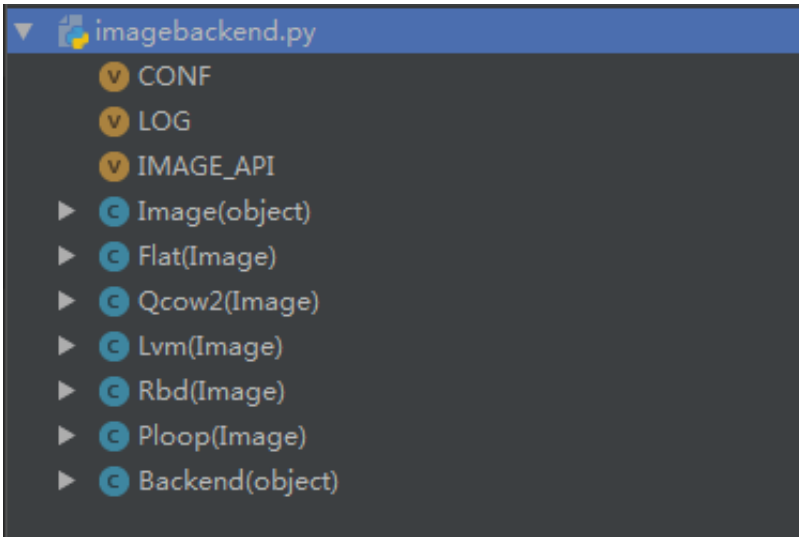
```
[libvirt]
images_type = rbd
images_rbd_pool = vms
images_rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_user = cinder
rbd_secret_uuid = 457eb676-33da-42ec-9a8c-9293d545c337
disk_cachemodes="network=writeback"
live_migration_flag="VIR_MIGRATE_UNDEFINE_SOURCE,VIR_MIGRATE_PEER2PEER,VIR
```

分析：
当Nova创建虚机时，如果不创建卷那么Nova会自己管理Image文件。
根据libvirt中配置的images_type值，nova会选择不同的Image后端：

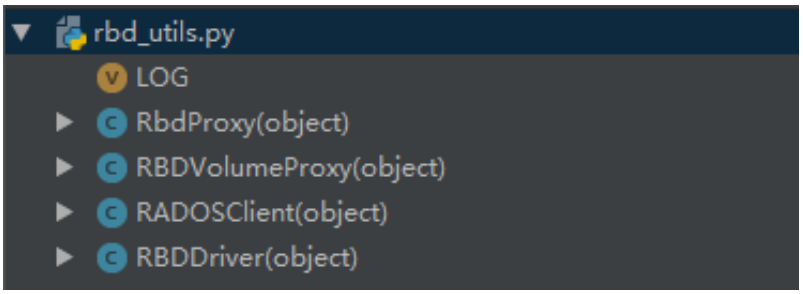
```
cfg.StrOpt('images_type',default='default', choices=('raw', 'flat', 'qcow2
```

当images_type的值是default时，nova根据use_cow的配置选择flat模式（不支持写时复制）或者qcow2模式（支持写时复制），此时文件是放在nova/instances目录中的。raw类型等价于flat。

Nova创建虚机镜像文件的过程如下：
virt.libvirt.driver.LibvirtDriver#spawn ---> virt.libvirt.driver.LibvirtDriver#_create_image ---> virt.libvirt.driver.LibvirtDriver#_create_and_inject_local_root ---> virt.libvirt.imagebackend.Backend#by_name（创建镜像对应的Backend Class）



Nova创建 Rbd 对应的 backend 类之后调用clone方法，直接克隆glance存在Ceph中的镜像（要求镜像格式时raw格式或者iso格式）。
Nova此时调用的rbd工具是nova/virt/libvirt/storage/rbd_utils.py



配置Ceph的管理套接字

```
# 调整这些路径的权限
mkdir -p /var/run/ceph/guests/ /var/log/qemu/
chown qemu:libvirtd /var/run/ceph/guests /var/log/qemu/
# 编辑所有计算节点上的 Ceph 配置文件
[client]
admin socket = /var/run/ceph/guests/$cluster-$type.$id.$pid.$cctid.asok
```

```
7 log file = /var/log/qemu/qemu-guest-$pid.log
```