**1.  snapshot overview**

对openstack而言，虚拟机的快照即是镜像，快照做完后以镜像形式存于glance。虽然openstack的快照是基于libvirt(qemu-kvm)，但是二者在实现上有很大区别：

* libvirt 主流快照实现： 采用virDomainSnapshotCreateXML()函数(CLI为virsh snapshot-create)。 新建的快照与虚拟机有关联：若为内置快照，快照信息和虚拟机存在同一个qcow2镜像中；若为外置快照，新建一个qcow2文件，原虚拟机的disk将变 为一个read only的模板镜像，新qcow2镜像仅记录与2.模板镜像的差异数据。这种快照包含快照链信息，可保留disk和ram信息，可回滚至快照点。
* openstack 快照实现：openstack并未采用virDomainSnapshotCreateXML()来实现快照，而是单纯的对虚拟机镜像做转换和拷贝，生成 一个与虚拟机无关联的镜像，最后上传至glance中。这种快照不包含快照链信息，只保留disk信息，无法回滚至快照点，只能采用该快照镜像创建一个新 的虚拟机。

**2. cold snapshot and live snapshot**

* cold snapshot:  创建snapshot时，需暂停虚拟机。
* live snapshot:   创建snapshot时，无需暂停虚拟机。

**3. cold snapshot 流程:**

# Save the state and stop a running guest, then detach pci devices

**$ virsh   managedsave   vm**

# Create a qemu internal snapshot

**$ qemu-img    snapshot   -c   snap1   vm\_snapshot**

# Extract the internal snapshot, convert it to qcow2 and export it a file, then upload to glance

**$ qemu-img    convert   -f   qcow2   vm   -O   qcow2   vm\_snapshot**

# Start the guest again

**$ virsh   start   vm**

**4. live snapshot 流程**

# Abort any failed/finished block operations:

**$ virsh blockjob vm vda --abort**

# Undefine a running domain. (Note: Undefining a running domain does not \_kill\_ the domain, it just converts it from persistent to transient.)

**$ virsh undefine vm**

# create a destination image with the original backing file and matching size of the instance root disk.

**$ qemu-img create -f qcow2 vm\_copy --backing\_file=backing\_file --size=root\_disk\_size**

#Invoke 'virsh blockcopy' (This will take time, depending on the size of disk image vm1):

**$ virsh blockcopy --domain vm vda vm\_copy --wait --verbose**

#Abort any failed/finished block operations:

**$ virsh blockjob vm vda --abort**

#Define the guest again (to make it persistent):

**$ virsh define vm**

#From the obtained new copy, convert the QCOW2 with a backing file to a qcow2 image with no backing file, then upload to glance:

**$ qemu-img convert -f qcow2 -O raw vm\_copy vm\_convert**

**5. virsh snapshot-create-as/snapshot-create 快照简析**

    默认为内置快照，支持快照链，支持快照回滚，支持内存信息。

    快照过程中，虚拟机短暂卡顿。

**不知 openstack 为什么不通过该方法实现主流快照 ？！**

**6. conclusion**

支持功能

* 虚拟机快照（快照存放于glance）
* 卷快照（LVM快照存放于本地）

限制与缺点

* 没有快照链信息，不支持revert恢复虚拟机到某一个快照点
* 只对系统盘进行快照，不支持内存快照，不支持同时对虚拟机和磁盘做快照
* 需要用户进行一致性操作
* 不支持含元数据导出，不支持含元数据导入
* 只支持虚拟机全量数据快照（与快照的实现方式有关，因为是通过image进行保存的），过程较长（需要先通过存储快照，然后抽取并上传至glance)，快照以Image方式保存，而非以cinder卷方式保存，无法充分利用存储本身能力加快快照的创建和使用