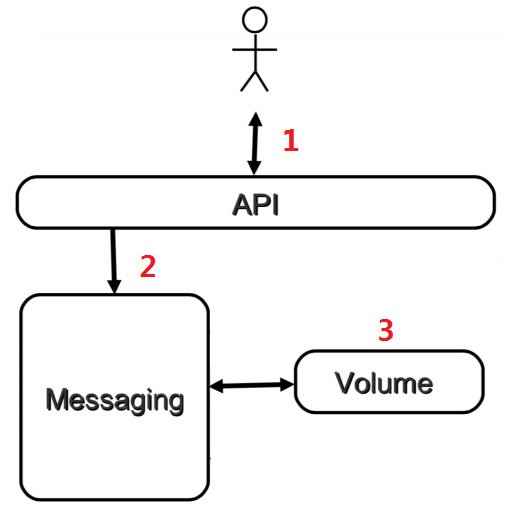


Snapshot 可以为 volume 创建快照，快照中保存了 volume 当前的状态，以后可以通过 snapshot 回溯。snapshot 操作实现比较简单，流程图如下：



1. 向 cinder-api 发送 snapshot 请求
2. cinder-api 发送消息
3. cinder-volume 执行 snapshot 操作

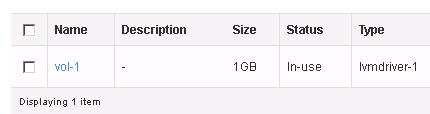
下面我们详细讨论每一个步骤。

**向 cinder-api 发送 snapshot 请求**

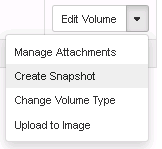
客户（可以是 OpenStack 最终用户，也可以是其他程序）向 cinder-api 发送请求：“请 snapshot 指定的 volume。

这里我们将 snapshot volume “vol-1”。

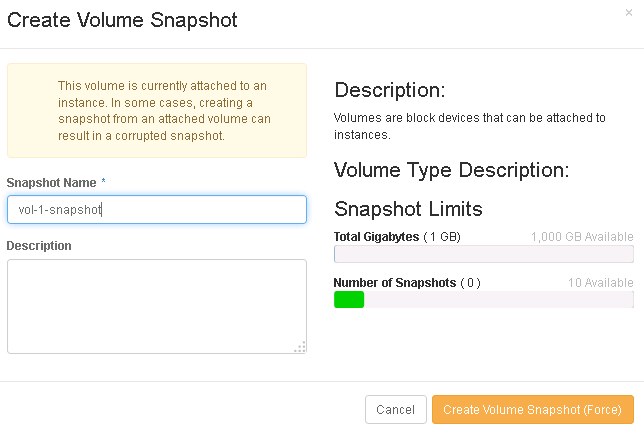
进入 GUI 操作菜单 Project -> Compute -> Volumes。



选择 volume “vol-1”，点击 “Create Snapshot”。

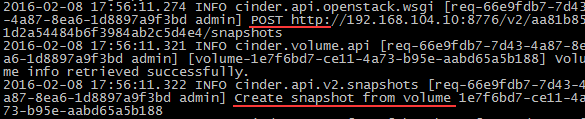


为 snapshot 命名。



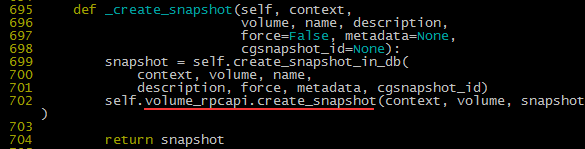
这里我们看到界面提示当前 volume 已经 attach 到某个 instance，创建 snapshot 可能导致数据不一致。我们可以先 pause instance，或者确认当前 instance 没有大量的磁盘 IO，处于相对稳定的状态，则可以创建 snapshot，否则还是建议先 detach volume 在做 sanpshot。

cinder-api 将接收到 snapshot volume 的请求。日志文件在 /opt/stack/logs/c-api.log。



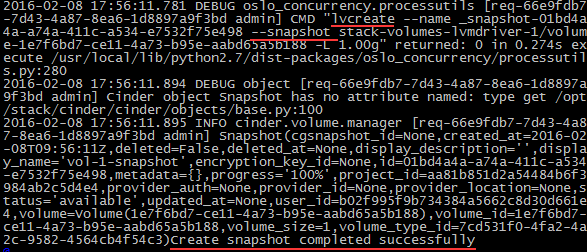
**cinder-api 发送消息**

cinder-api 发送消息 snapshot 消息。cinder-api 没有打印发送消息的日志，只能通过源代码查看 /opt/stack/cinder/cinder/volume/api.py，方法为 \_create\_snapshot。

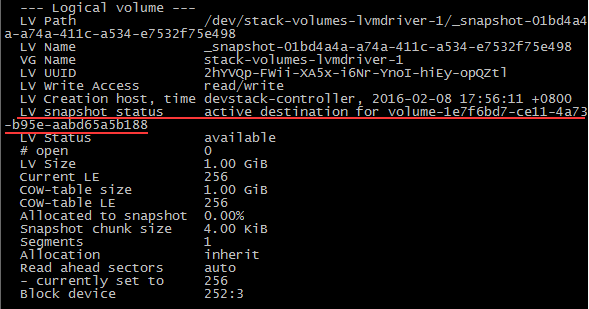


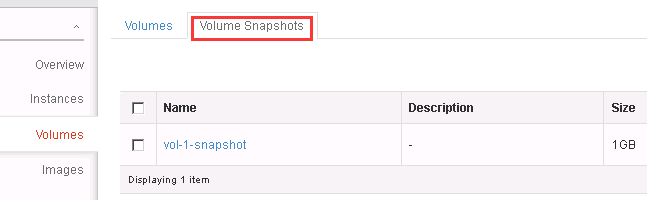
**cinder-volume 执行 snapshot 操作**

cinder-volume 执行 lvcreate 命令创建 snapshot。 日志为 /opt/stack/logs/c-vol.log。

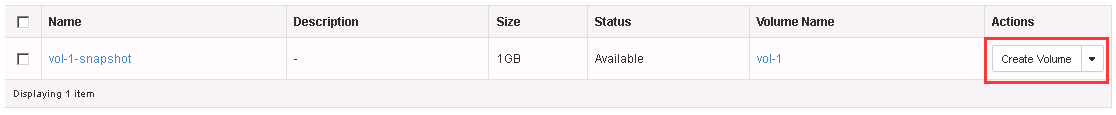


对于 LVM volume provider，snapshot 实际上也是一个 LV，同时记录了与源 LV 的 snapshot 关系，可以通过 lvdisplay 查看。

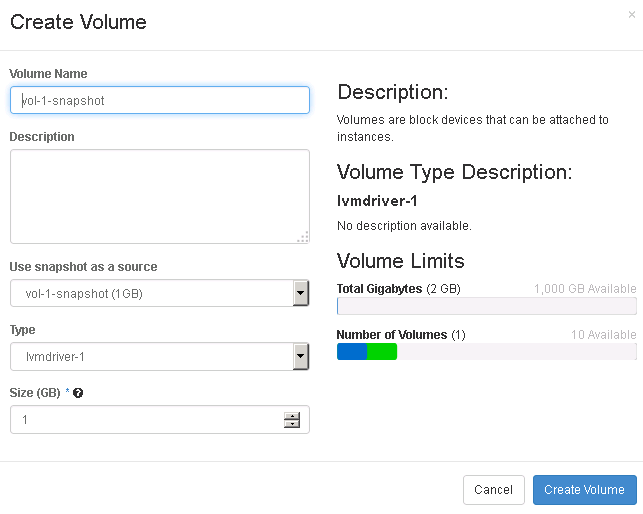
  
GUI 的 Volume Snapshots 标签中可以看到新创建的 “vol-1-snapshot”。



有了 snapshot，我们就可以将 volume 回溯到创建 snapshot 时的状态。方法是通过 snapshot 创建新的 volume。



新创建的 volume 容量必须大于或等于 snapshot 的容量



其过程与 Create Volume 类似，不同之处在于 LV 创建之后会通过 dd 将 snapshot 的数据 copy 到新的 volume。  
日志分析不再赘述，留个大家练习。

如果一个 volume 存在 snapshot，则这个 volume 是无法删除的。  
这是因为 snapshot 依赖于 volume，snapshot 无法独立存在。

在 LVM 作为 volume provider 的环境中，snapshot 是从源 volume 完全 copy 而来，所以这种依赖关系不强。  
但在其他 volume provider（比如商业存储设备或者分布式文件系统），snapshot 通常是源 volume 创建快照时数据状态的一个引用（指针），占用空间非常小，在这种实现方式里 snapshot 对源 volume 的依赖就非常明显了。

以上就是 volume snapshot 的分析，下一节我们讨论一个与 snapshot 容易混淆的操作：backup。