这是 OpenStack 实施经验分享系列的第 13 篇。

instance snapshot 操作可用于备份或者将 instance 保存为新的 image。如果在生产系统中执行 snapshot 操作，必须确保此操作快速且安全。这里有两个关键点：

1. **快速。**   
   为保证数据的一致性，snapshot 时需要 pause instance，操作完后再 resume。在这个过程中 instance 是无法对外服务的，为了减少对业务的影响，我们希望 snapshot 越快越好。
2. **安全。**   
   即数据一致性，snapshot 出来的 image 不能有没落盘的数据，能够正常启动。所以通常在执行 snapshot 前要 pause instance，暂停所有的 IO 操作。

**默认的 snapshot**

默认配置下的 snapshot 操作是否能满足快速和安全这两个条件呢？

snapshot 是对 instance 的镜像文件（系统盘）做快照，镜像文件位于计算节点 /var/lib/nova/instances/<instance id>/disk。在第036篇中我们详细讨论了 snapshot 的执行步骤：

1. pause instance
2. 执行 qemu-img convert 命令复制 disk 文件，生成快照文件
3. resume instance
4. 将快照文件上传到 Glance

其中第 1 步保证了 安全，而是否 快速 取决于第 2 步要花多长时间。qemu-img convert 的执行时间取决于 disk 及其 backing 文件的大小，通常 instance 系统盘都以 G 为单位，所以第 2 步花费的时间是分钟级别的。

让生产系统暂停几分钟通常是不能被接受的，所以默认的 snapshot 操作没法做到 快速。

解决方案是什么呢？

**不靠谱的 live snapshot**

Nova 很早就提出了 live snapshot 的替代方案，具体见官网 <http://docs.openstack.org/ops-guide/ops-user-facing-operations.html#live-snapshots>

live snapshot 的原理是：做快照时不 pause instance，直接执行 qemu-img convert。也就是去掉第 1 和 第 3 步。

这样虽然 快速 条件满足了，业务不会受到影响。但由于没有 pause instance，有可能出现快照过程中不断有新数据写进 disk 文件的情况，很难保证数据的一致性，结果 安全 又成了问题。

官网文档建议：如果要做 live snapshot，用户必须自己保证数据的一致性，比如做快照前确保所有数据已经落盘，并且不会有新数据写进来。个人觉得，live snapshot 基本上没法在生产中使用。

那到底有没有既 安全 又 快速 的方案呢？

**真正的解决方案**

默认 snapshot 的问题在于 qemu-img convert 耗时太长，而耗时太长的原因是 instance 的系统盘是文件，拷贝文件本身就是一个耗时的操作。真正的解决方案是：

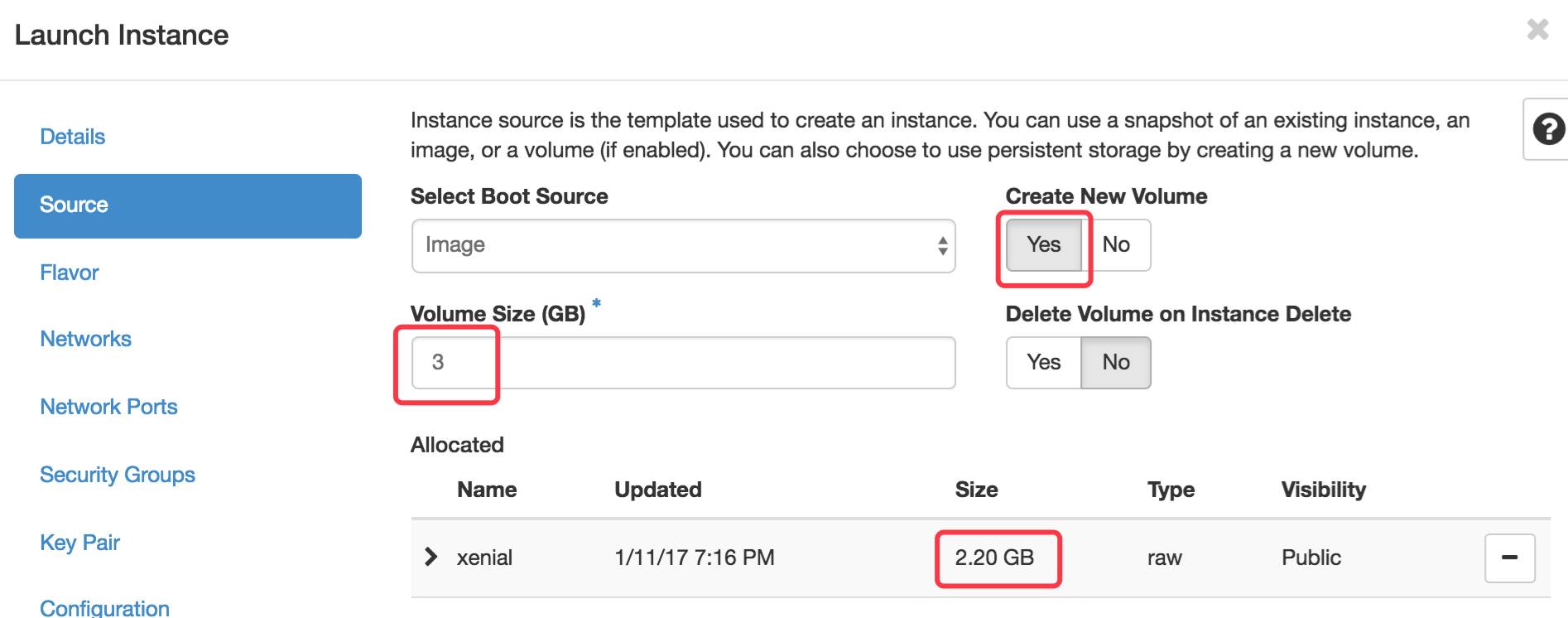
**让 instance 从 cinder volume 启动，利用 storage provider 自己的 snapshot 技术实现快速复制。**

现代存储系统（无论开源还是商业存储）基本上都提供了 volume 的 snapshot 功能，而这个 snapshot 是基于指针的，不会真正拷贝数据，所以非常快，通常一瞬间就完成，对业务几乎没有影响。所以如果 instance 是从 cinder volume 启动的，那么做快照的时候 OpenStack 就会使用 storage provider 的 snapshot 完成操作。这就实现了既 安全 又 快速。

下面我们使用流行的分布式存储系统 ceph 来演示这个过程。这里省略了 ceph 作为 cinder backend 的配置方法，如果有兴趣可以参考官网 <http://docs.ceph.com/docs/master/rbd/rbd-openstack/>

**boot from volume**

部署 instance 时指定创建 volume。



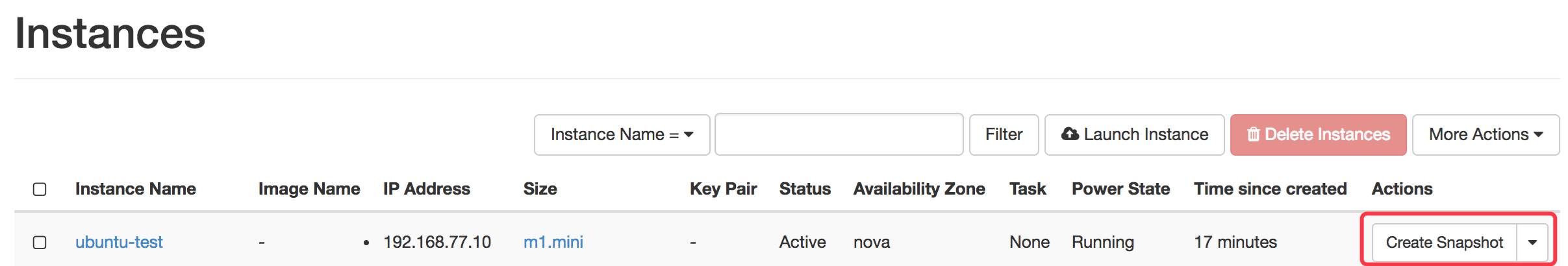
这里我们选择的 image 是 xenial（Ubuntu 16.04），大小为 2.20 GB。确保 Create New Volume 选项为 Yes。Volume Size 为 3（大于 image 2.20 GB）。执行部署后，OpenStack 会完成如下工作：

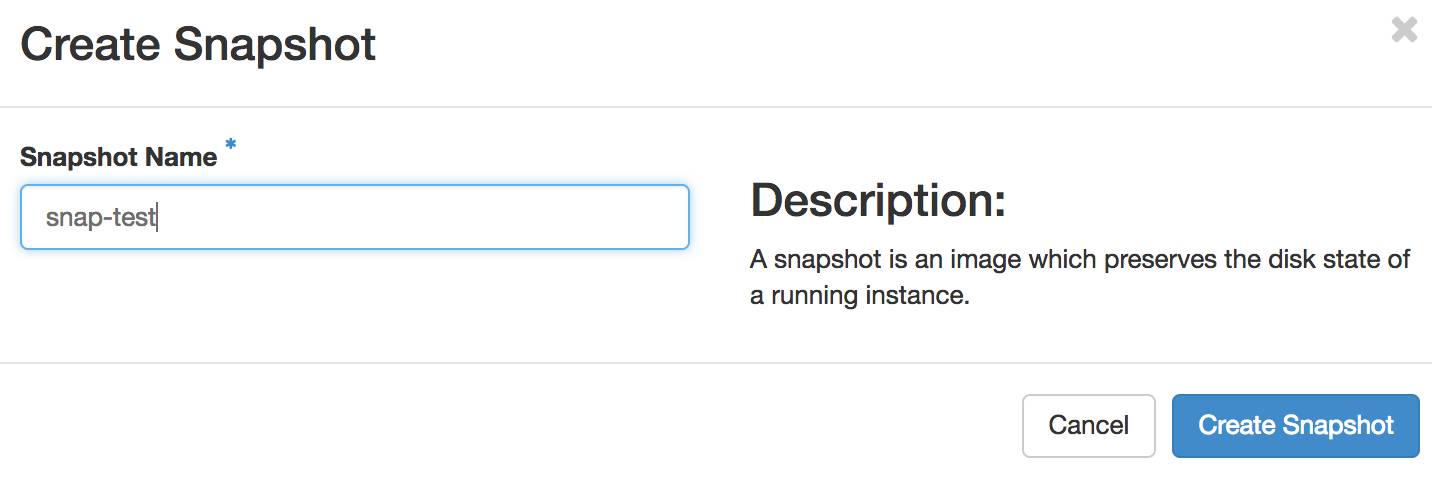
1. 在 ceph 中创建一个 3 GB 的 volume。
2. 将 image 数据拷贝到该 volume。
3. instance 从该 volume 启动。

在 volume 管理界面可以看到这个新创建的 volume。

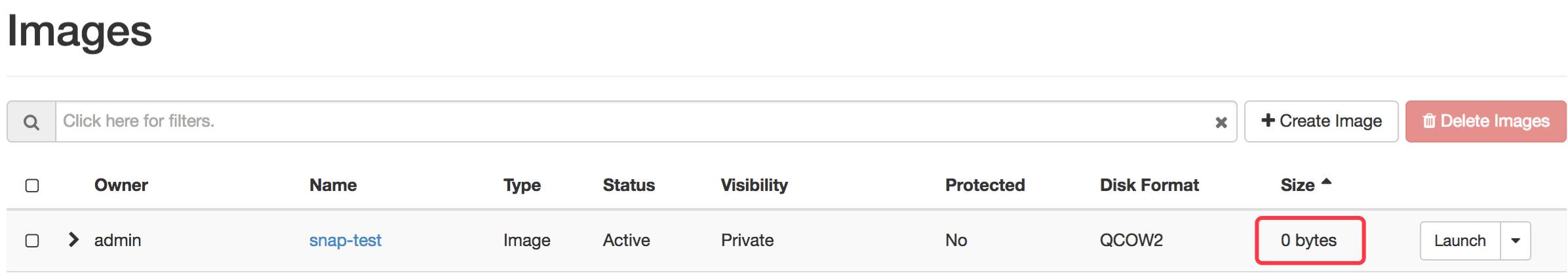


ubuntu-test 就是我们部署的 instance。接下来对 ubuntu-test 执行 snapshot 操作。





操作瞬间完成！



注意到快照 snap-test 的 Size 是 0 字节，这是因为它真正的存放位置在 cehp。通过 snap-test 部署出来的 instance 直接就是 boot from volume 的。

boot from volume 其实是 OpenStack 部署 instance 的最佳实践，instance 的启动盘和数据盘都由 cinder 管理，而且做快照和做备份都很方便。

**下期预告**

到这里，实施经验分享部分就先告一段落。按照之前的计划，接下来是容器技术的内容。不过最近收到很多同学的留言，希望讲一讲 cloud-init 的工作原理和相关应用。

instance 定制化其实是个非常重要的内容，在生产环境中的需求很大。目前最主流的方案就是 cloud-init，当然仅仅 cloud-init 是不够的，还得需要 OpenStack 服务的支持。前面之所以没有讨论，主要是因为这个主题会同时涉及 nova 和 neutron 两大模块，要求的知识和技能比较综合，不过现在则是个非常好的时机。

接下来 CloudMan 会系统讲解 instance 定制化这个主题，从原理到实践力求把它讲透。只是容器部分不得不再多等一下了，见谅见谅。