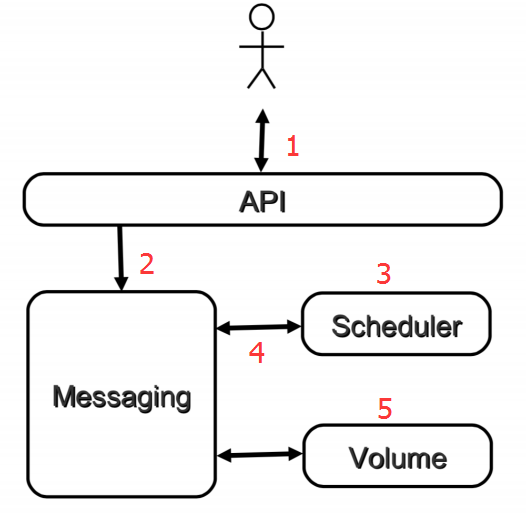


前面已经学习了 Cinder 的架构和相关组件，从本节我们开始详细分析 Cinder 的各种操作，首先讨论 Cinder 如何创建 volume。

Create 操作流程如下：



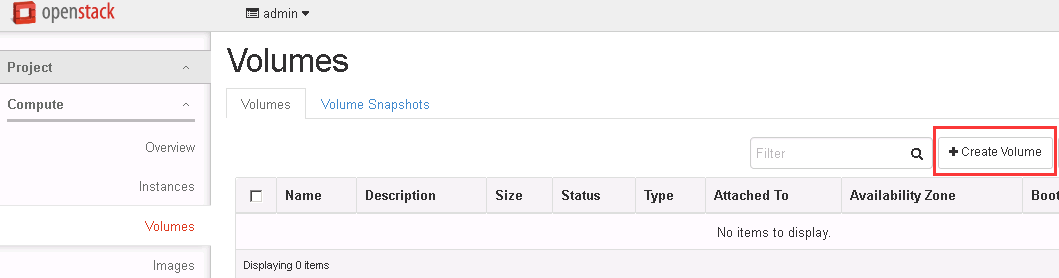
1. 客户（可以是 OpenStack 最终用户，也可以是其他程序）向 API（cinder-api）发送请求：“帮我创建一个 volume”。
2. API 对请求做一些必要处理后，向 Messaging（RabbitMQ）发送了一条消息：“让 Scheduler 创建一个 volume”。
3. Scheduler（cinder-scheduler）从 Messaging 获取到 API 发给它的消息，然后执行调度算法，从若干计存储点中选出节点 A。
4. Scheduler 向 Messaging 发送了一条消息：“让存储节点 A 创建这个 volume”。
5. 存储节点 A 的 Volume（cinder-volume）从 Messaging 中获取到 Scheduler 发给它的消息，然后通过 driver 在 volume provider 上创建 volume。

因为 Create Volume 操作比较复杂，我们将分三次讨论：  
今天是第一部分，讨论 cinder-api 的处理过程；  
第二部分讨论 cinder-scheduler；  
第三部分讨论 cinder-volume 的操作。

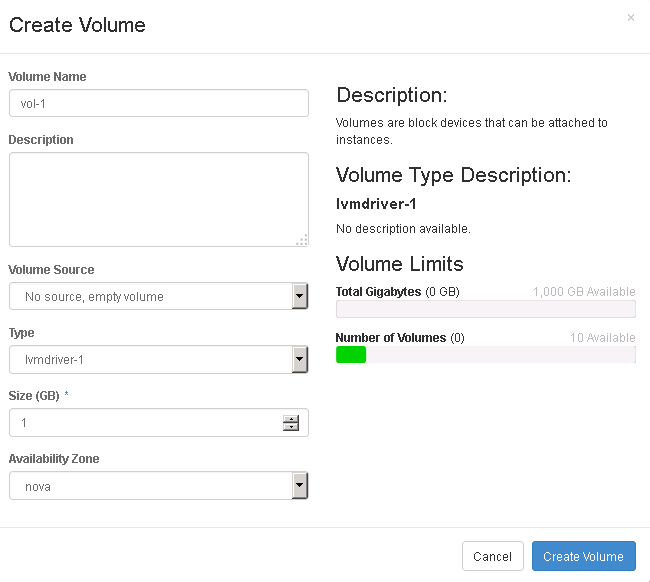
**向 cinder-api 发送请求**

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：“帮我创建一个 volume。

GUI 上操作的菜单为 Project -> Compute -> Volumes -> Create Volume



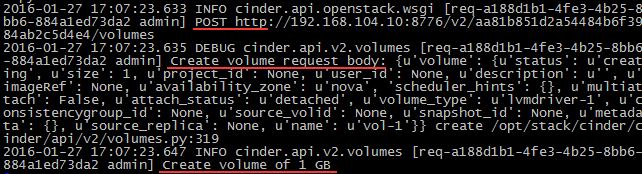
设置 volume 的名称，volume type，大小，Availability Zone 等基本信息。



这里我们没有设置 Volume Source，这样会创建一个空白的 volume。

点击“Create Volume” 按钮，cinder-api 将接收到创建 volume 的请求。

查看 cinder-api 日志 /opt/stack/logs/c-api.log。



日志显示 cinder-api 接收到一个 POST 类型的 REST API，经过对 HTTP body 的分析，该请求是：创建一个 1GB 的 volume。

紧接着，cinder-api 启动了一个 Flow（工作流）volume\_create\_api。

Flow 的执行状态依次为 PENDING, RUNNING 和 SUCCESS。volume\_create\_api 当前的状态由 PENDING 变为 RUNNING。

http://7xo6kd.com1.z0.glb.clouddn.com/upload-ueditor-image-20160621-1466485443157049363.jpg

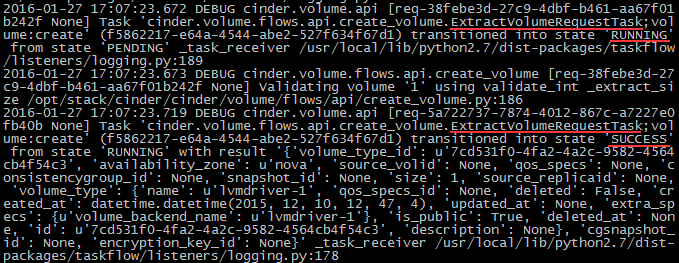
volume\_create\_api 工作流包含若干 Task，每个 Task 完成特定的任务。

这些任务依次为 ExtractVolumeRequestTask, QuotaReserveTask, EntryCreateTask, QuotaCommitTask, VolumeCastTask。

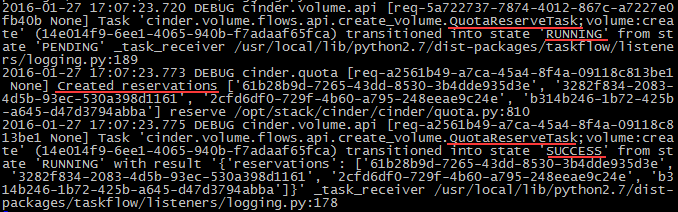
Task 的执行状态也会经历 PENDING, RUNNING 和 SUCCESS 三个阶段。

Task 的名称基本上说明了任务的工作内容，前面几个 Task 主要是做一些创建 volume 的准备工作，比如：

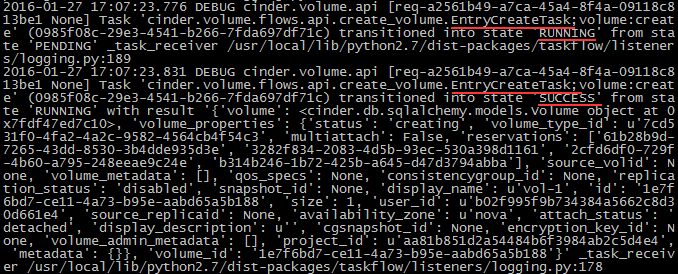
**ExtractVolumeRequestTask** 获取 request 信息



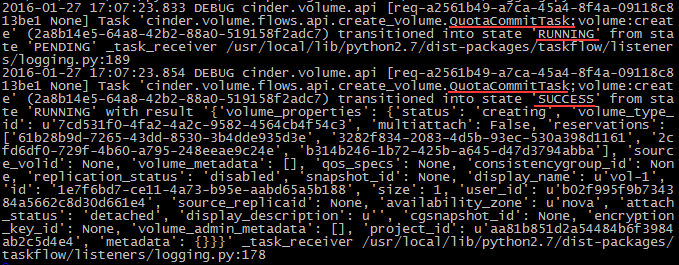
**QuotaReserveTask** 预留配额



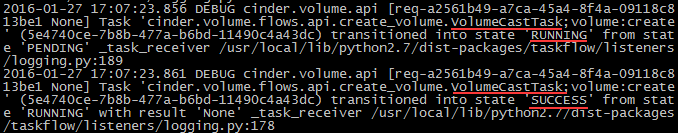
**EntryCreateTask** 在数据库中创建 volume 条目



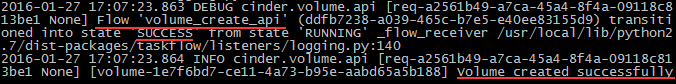
**QuotaCommitTask** 确认配额



最后 **VolumeCastTask** 是向 cinder-sheduler 发送消息，开始调度工作



至此，Flow volume\_create\_api 已经完成，状态由 RUNNING 变为 SUCCESS，volume 创建成功。日志如下：



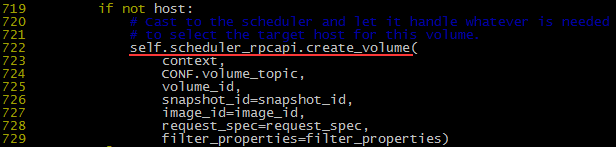
需要特别注意的是，“volume 创建成功”只是指 cinder-api 已经成功处理了 volume create 请求，将消息发给了 cinder-scheduler，但并不意味 volume 在存储节点上已经成功创建，这一点是容易引起误解的。我们可以通过 cinder-volume 创建 volume 日志的时间戳验证。

**cinder-api 发送消息**

cinder-api 向 RabbitMQ 发送了一条消息：“让cinder-scheduler 创建一个 volume”

前面我们提到消息是由 VolumeCastTask 发出的，因为 VolumeCastTask 没有打印相关日志，我们只能通过源代码查看

/opt/stack/cinder/cinder/volume/flows/api/create\_volume.py ，方法为 create\_volume。



下一节我们讨论 Create Volume 的第二部分： cinder-scheduler 的处理过程。