# 一．Cinder

## （一）Block Storage块存储

操作系统有两种获取存储空间的方式

### 1.硬盘存储（块存储）

通过某种协议（SAS，SCSI，SAN，ISCSI等）挂接裸硬盘，然后进行分区，格式化，创建文件系统，或者直接使用裸硬盘存储数据（数据库）。每个裸硬盘称为Volume卷。

### 2.远程文件系统

通过NFS，CIFS等协议，mount远程的文件系统。如NAS，NFS服务器以及各种分布式文件系统。

## （二）Cinder服务

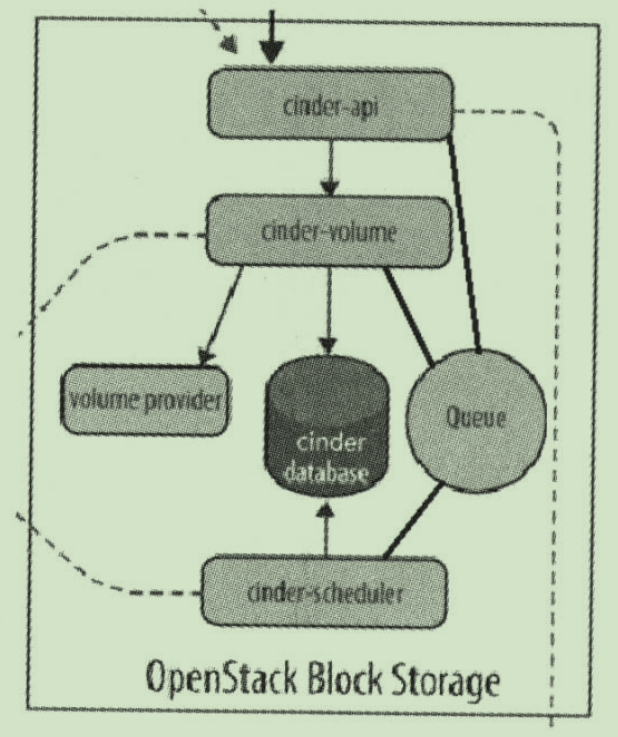
### 1.Cinder的功能

（1）提供REST API使用户能够查询和管理volume，volume snapshot 以及 volume type；

（2）提供scheduler调度volume创建请求，合理优化存储资源的分配；

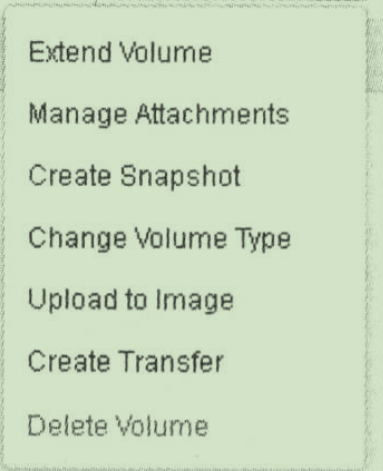
（3）通过driver架构支持多种back-end（后端）存储方式，包括LVM，NFS，Ceph和其他存储方案；

### 2.Cinder的架构



#### （1）cinder-api

接受API请求，调用cinder-volume执行操作；



#### （2）cinder-volume

管理volume的服务，与volume provider协调工作，管理volume的生命周期，运行cinder-volume服务的节点被称为存储节点，OpenStack对Volume的操作，最后都是交给cinder-volume来完成的。cinder-volume自身并不管理真正的存储设备，它是由volume provider管理的，cinder-volume 和 volume provider一起实现volume生命周期的管理。

#### （3）cinder-scheduler

通过调度算法选择最合适的存储节点创建volume。创建Volume时，cinder-scheduler会基于容量，volume Type等条件选择出最合适的存储节点，然后让其创建Volume。和nova-scheduler一样，都是通过多个过滤器实现的条件过滤，然后计算权重选择权重最大的存储节点。

在/etc/cinder/cinder.conf中，cinder通过schedulerdriver, schedulerdefault\_filters 和 schedulerdefaultweighters三个参数来配置cinder-scheduler。

在cinder配置文件中，还可以通过设置“volume\_backend\_name”参数，为存储节点的Volume Provider命令，不同的存储节点可以在各自的cinder.conf中配置相同的volumebackendname，因为它们可能使用的是一种Volume Provider。

默认的计算权重Weigter，使用的是存储节点的空闲容量计算权重值。

#### （4）volume provider

数据的存储设备，为volume提供物理存储空间，cinder-volume支持多种volume provider，每种volume provider通过自己的driver与cinder-volume协调工作。

#### （5）Message Queue

Cinder各个子服务之间通过消息队列实现进程之间的通信和相互协作，实现子服务之间的解耦。

#### （6）Database

Cinder有一些数据需要存放到数据库中，一般使用的是Mysql。

### 3.Cinder子服务的物理部署方案

Cinder的各种子服务会部署在两类节点上，控制节点和存储节点。

#### （1）控制节点

cinder-api；

cinder-scheduler；

cinder-volume;

RabbitMQ；

Mysql；

#### （2）存储节点

cinder-volume；

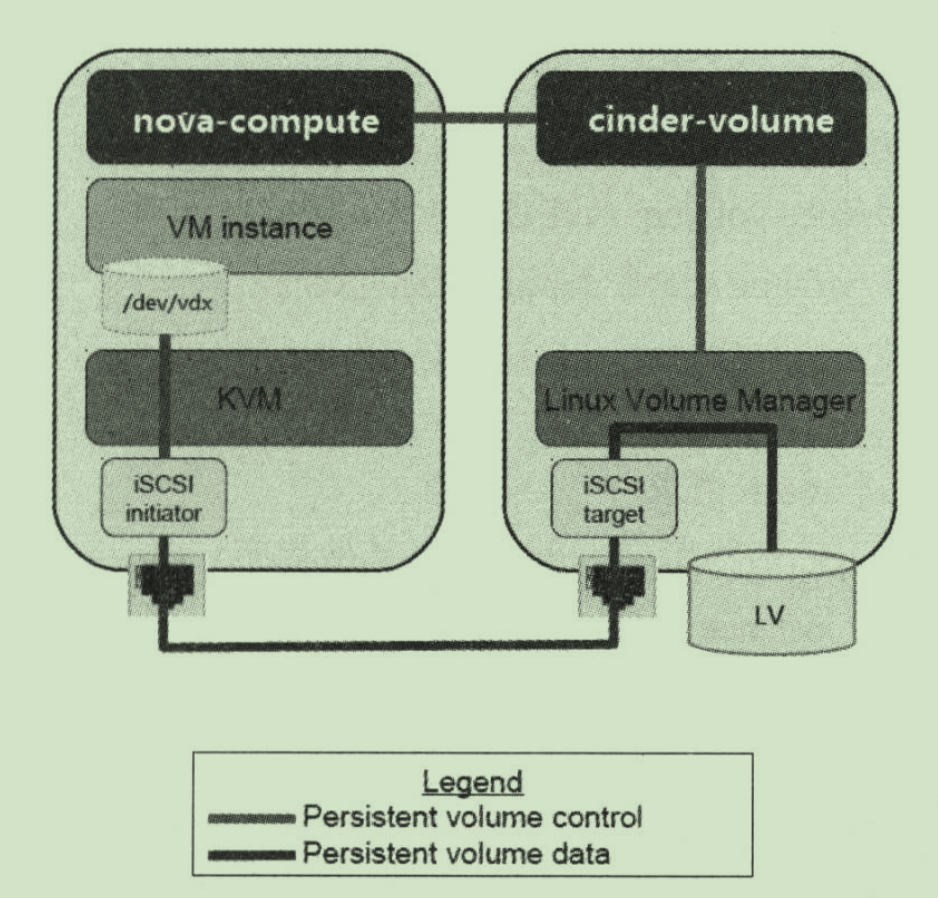
注意，volume provider是独立的，cinder-volume使用driver与volume provider通信并协调工作，并且只需要将driver与cinder-volume放在一起就可以了。

### 4.Cinder操作详解

#### （1）Create Volume

1. 向cinder-api发送请求，设置vvolume的名称，volume type, 大小， Availability Zone，接着cinder-api启动一个Flow工作流 volumecreateapi，其中包含了若干个Task，每个Task完成特定的任务，这些任务依次为ExtractVolumeRequestTask，QuotaReserveTask，EntryCreateTask，QuotaCommitTask，VolumeCastTask，这些任务的名称就已经说明了任务的工作内容，前面的Task是做创建volume的准备工作，如获取request工作，预留配额，在数据库中创建volume条目，最后向cinder-scheduler发送消息，开始调度工作 ；
2. cinder-api接收请求，并向RabbitMQ发送消息；
3. cinder-scheduler从RabbitMQ中获取到cinder-api发给它的消息，然后执行调度算法，从若干个存储节点中选出节点；
4. cinder-scheduler向RabbitMQ发送消息，让存储节点创建volume；
5. 存储节点的cinder-volume从RabbitMQ中获取到cinder-scheduler发送给它的消息后，通过driver在volume provider上创建volume，cinder-volume也启动了一个Flow来完成volume创建工作，Flow的名称为volume\_create\_manager；
6. 当前的volume provider为LVM，因此工作流通过lvcreate命令在VG（虚拟组）创建了一个1G的LV，cinder-volume 将这个LV作为volume，完成volume的创建工作。

#### （2）Attach



Volume最主要的用途是作为虚拟硬盘提供给instance使用，通过Attach操作挂载到instance上的。每个volume实际上是存储节点上VG中的一个LV，那么存储节点上本地的LV怎样挂载到计算节点上的instance上呢？

由于存储节点和计算节点是不同的物理节点，通常是使用ISCSI，ISCSI是CS架构，有target和initiator两个术语，Target是提供ISCSI存储资源的设备，即ISCSI服务器；Initiator是使用ISCSI存储资源的设备，也就是ISCSI客户端。

Initiator需要与Target建立ISCSI连接，执行login操作，然后就可以使用target上面的块存储设备了。Target提供的块存储设备支持多种实现方式，Cinder的存储节点ccinder-volume默认使用tgt软件来管理和监控ISCSI Target，在计算节点nova-compute使用ISCSIADM执行Initiator相关操作。

1. 向cinder-api发送attach请求，指定将某个volume attach到指定的instance上，这个请求包含两个步骤：
2. 初始化volume的连接

由于volume创建后，只是在volume provider中创建了相应的存储对象（LV），cinder-volume 需要将volume export出来，计算节点才能够访问到volume，这个export的过程就是初始化volume的连接，这个工作主要由cinder-volume完成；

1. Attach volume

初始化volume连接之后，计算节点将volume挂载到指定的instance上，完成attach操作。attach的操作主要由nova-compute完成

1. cinder-api发送消息

cinder-api会向RabbitMQ发送两条消息，一条是初始化volume的连接，另一条是attach volume。

1. cinder-volume初始化volume的连接

cinder-volume接受到初始化连接消息之后，会通过tgt创建target，并将volume所对应的LV通过target export出来。

1. nova-compute将volume attach到instance

计算节点作为ISCSI Initiator访问存储节点ISCSI Target 上的volume，并将其attach到instance。

nova-compute依次执行了iscsiadm 的new，update，login，rescan操作访问target上的volume。计算节点会将ISCSI Target上的volume识别为一个磁盘文件，然后通过更新instance的XML配置文件将volume映射给instance。

#### （3）Detach

与volume attach相对的操作是Detach，就是将volume从instance卸载下来。Detach的操作由nova-compute和cinder-volume共同完成，首先nova-compute将volume从instance上detach，然后断开与ISCSI target的连接，最后cinder-volume删除volume相关的ISCSI target。

#### （4）Extend

Extend操作扩大了volume的容量，注意，只有状态为Available的volume才能够被extend。如果当前的volume已经被attach给了Instance，需要先detach之后才能够extend。

是由cinder-volume执行的lvextend命令extend volume。

#### （5）Delete

只有装填为Available的volume才能够被delete，如果volume当前已经被attach到了Instance上，需要先detach之后才能够delete。

是由cinder-volume执行lvremove命令delete volume的，注意，它执行的是安全地删除，所谓的安全指的是将volume中的数据抹掉，LVM driver使用的是dd操作符将LV的数据清零，然后删除LV。

#### （6）Snapshot

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。当volume已经attach到了某个instance上时，创建snapshot可能导致数据的不一致，我们可以先pause instance，或者确认当前的instance没有大量的磁盘IO，处于相对稳定的状态，则可以创建snapshot，否则还是先detach volume 再做snapshot。

cinder-volume会执行LVcreate命令创建snapshot。对于LVM 去咯么 provider来说，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命令查看。

有了snapshot，就可以将volume回溯到创建snapshot时的状态，方法时通过snapshot创建新的volume，新创建的volume的容量必须大于或等于snapshot的容量。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无法独立存在的。在LVM作为volume provider的环境中，snapshot是从源volume完全copy起来的，所以这种依赖关系不强。但是在其他的volume provider中，snapshot通常是源volume创建快照时的一个引用，占用空间很小。

#### （7）Backup

Backup操作是将volume备份到其他的地方，将来可以通过restore操作恢复。backup是存放在独立的备份设备中的，有自己的备份方案和实现，与volume provider没有关系，因此backup具有容灾功能。

Cinder的backup功能是由cinder-backup服务提供的，cinder-backup也是通过driver架构支持多种备份backend。

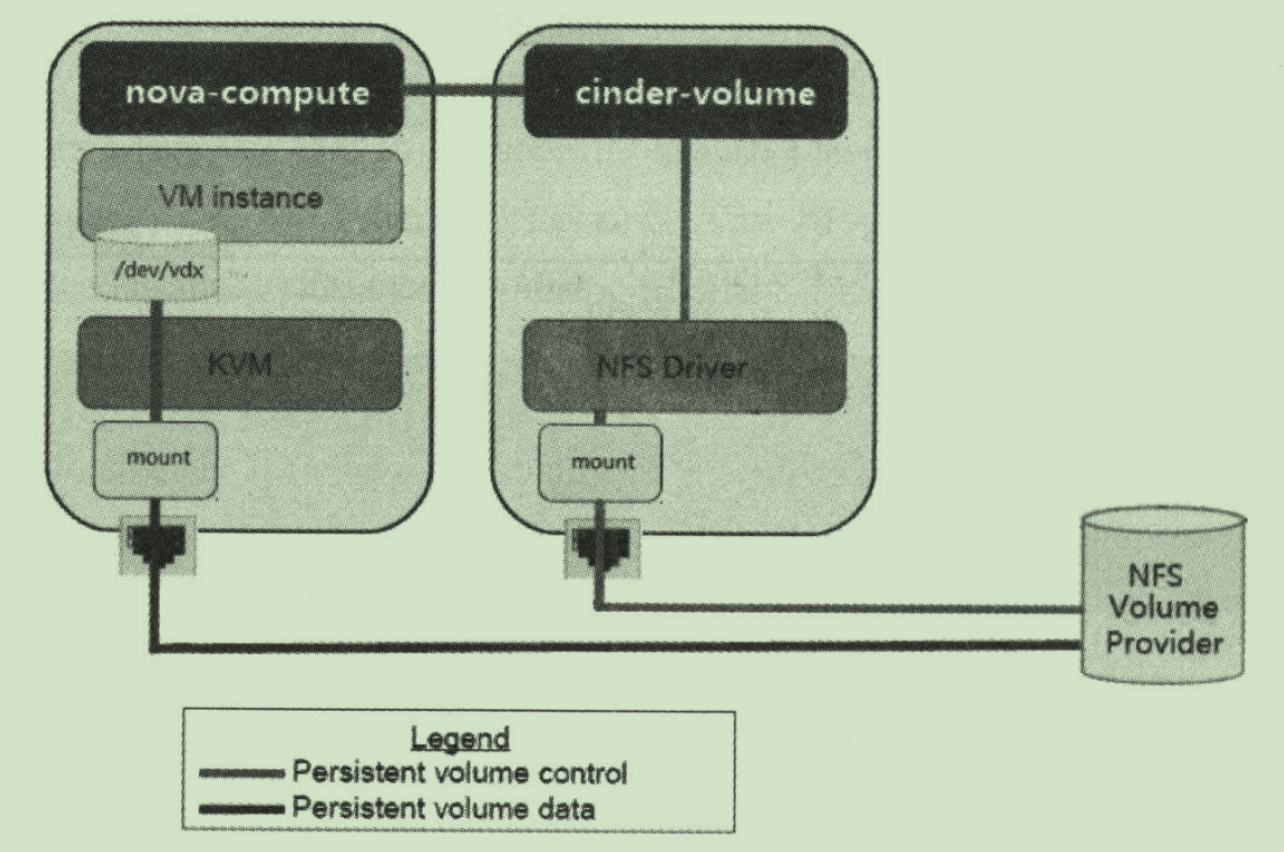
#### （8）Restore

通过cinder-scheduler和cinder-volume在存储节点上创建一个空白volume；

cinder-backup启动restore 操作，然后将backup的数据copy到空白的volume上。

#### （9）NFS Volume Provider

由于cinder-volume支持多种volume provider，这里我们考虑使用NFS volume provider。



#### （10）NFS Volume Provider

就是通常所说的NFS Server，提供远程的NFS目录，NFS client可以mount这些远程目录到本地，然后像使用本地目录一样创建，读写文件以及子目录。

1. cinder-volume

存储节点通过NFS driver管理NFS volume provider中的volume，这些volume在NFS中其实是一个个文件。

1. nova-compute

计算节点将NFS volume provider 存放volume的目录mount到本地，然后将volume文件作为虚拟硬盘映射为instance。

注意，Instance读写volume时，数据流不需要经过存储节点，而是直接对volume provider中的volume进行读写。存储节点与NFS volume provider之间进行连接完成volume的管理和控制，而数据连接，是通过计算节点和NFS Volume Provider之间的连接完成的，这种控制流和数据流分离的设计保证了读写的效率。