

```
[stack@localhost logs]$ cinder help migrate
usage: cinder migrate [--force-host-copy [<True|False>]]
                    [--lock-volume [<True|False>]]
                    <volume> <host>

Migrates volume to a new host.

Positional arguments:
  <volume>          ID of volume to migrate.
  <host>            Destination host. Takes the form: host@backend-
                    name#pool

Optional arguments:
  --force-host-copy [<True|False>]
                        Enables or disables generic host-based force-
                        migration, which bypasses driver optimizations.
                        Default=False.
  --lock-volume [<True|False>]
                        Enables or disables the termination of volume
                        migration caused by other commands. This option
                        applies to the available volume. True means it locks
                        the volume state and does not allow the migration to
                        be aborted. The volume status will be in maintenance
                        during the migration. False means it allows the volume
                        migration to be aborted. The volume status is still in
                        the original status. Default=False.
```

--force-host-copy : 该参数控制了迁移策略，为true时跳过后端迁移配置，采用文件拷贝的方式进行迁移（公共方案）

--lock-volume : 迁移时锁定volume的状态为 False

相关配置 migration_create_volume_timeout_secs 默认300超过时间，会造成迁移失败。

调用链：

Cinder_API :

cinder.api.contrib.admin_action.VolumeAdminController -->> migrate_volume

migrate动作的主接口，提供给用户调用

cinder.api.contrib.admin_action.VolumeAdminController -->> _migrate_volume_completion

migrate完成时被调用，这个接口是给nova调用的！！

cinder.volume.api.API -->>migrate_volume

1. 根据传入的HOST的backend名称从services表获取对应的Service信息；
2. 检查volume状态、快照等迁移条件是否满足；
3. 更新volume的数据库状态为迁移；
4. 调用scheduler的RPCAPI，进行迁移

Cinder_Scheduler :

cinder.scheduler.manager.SchedulerManager ->> migrate_volume

1. cinder.scheduler.filter_scheduler.FilterScheduler ->> backend_passes_filters

调用_get_weighted_candidates获取可以迁入的pools，并判断target_pool在不在pools中。（保证迁移目标的pool存在，并且能够接受待迁移的LUN）

2. 调用VolumeAPI的migrate_volume方法

Cinder_Volume :

cinder.volume.manager.VolumeManager -->> migrate_volume

c-vol 有两套迁移方案

方案一：

使用backend后端的迁移功能，及driver中实现的migrate_volume方法。

对于NetApp来说即cinder.volume.drivers.netapp.dataontap.iscsi_cmode.NetAppCmodeISCSIDriver中的

migrate_volume方法，

但是很遗憾netapp没有实现这个驱动。

方案二：

如果方案一失败，或者迁移时配置了force_host_copy，则采用方案二（及_migrate_volume_generic方法）。

该方案通过文件拷贝执行迁移，优势在于对于attach的卷也可以实施。

分析 _migrate_volume_generic方法：

cinder.volume.manager.VolumeManager --> _migrate_volume_generic

0. 在target_Host上创建一个空白卷

1. 下面对于存在连接的卷和无连接的卷有不同操作

A. 对于detached的卷直接调用_copy_volume_data进行数据拷贝。在_copy_volume_data中volume节点同时attached到src_vol和target_vol，然后调用cinder.volume.utils.copy_volume进行文件拷贝，完成之后再detached两个volume。当拷贝完成之后，migrate_volume_completion被调用，volume的状态将会发生修改！

关键日志如下：

```
LOG.debug('copy_data_between_volumes %(src)s -> %(dest)s.', {'src': src_vol['name'], 'dest': dest_vol['name']})
```

B. 对于attached到VM卷cinder会调用nova的API，由nova完成数据拷贝，并在完成拷贝后调

用 migrate_volume_completion。

对应的方法为：nova.api.openstack.compute.volumes.VolumeAttachmentController.update

migrate_volume_completion

1. 包括删除删除原卷

2. 修改新卷的表记录等