

参考

[pike版本Nova使用Cinder 新attach/detach 的API](#)
[Cinder新attach/detach API](#)

Nova如何使用新的API

attach流程：

1. (api)：调用attachment_create创建一个attachment记录。请求中不会携带connector参数，因此Cinder只创建一个空的attachment记录，并将卷状态改为reserved，不会返回target的连接信息。返回的attachment_id会被更新到BDM的记录中。
2. (compute)：调用attachment_update接口，该接口返回target的连接信息，并且在Cinder完成所有attach相关操作，卷状态被设定为in-use（也就是说Cinder这时候认为挂载已经挂完了，没我的事了）。这时候Nova Compute会等待Volume的状态更新完成，如果状态更新发生了错误，那么会将虚机的状态设为ERROR。
3. (compute)：使用os-brick连接磁盘到Host。
4. (compute)：将设备提供给VM使用。

detatch:

1. (api)：检查请求是否合法；
2. (compute)：从VM分离Volume，如果失败request停止；
3. (compute)：调用os-brick断开host和backend的连接；
4. (compute)：如果BDM记录中设定attachment_id，那么使用新的接口detach，否则使用旧接口，如果此时发生错误VM状态会被设为ERROR

对数据模型的影响

nova在blockdevicemapping表中可能不在保存connection_info信息，只保存attachment_id。

需要连接信息时，通过cinder接口获取。

Data model impact

When using the new API flow, we no longer need to store the `connection_info`, as we don't need to pass that back to Cinder. Instead we just store the `attachment_id` for each host the volume is attached to, and any time we need the `connection_info` we fetch that from Cinder.

When an `attachment_id` is populated, we use the new flow to do all attach or detach operations. When not present, we use the old flow.