

本节我们讨论 Suspend/Resume 和 Rescue/Unrescue 这两组操作。

**Suspend/Resume**

有时需要长时间暂停 instance，可以通过 Suspend 操作将 instance 的状态保存到宿主机的磁盘上。当需要恢复的时候，执行 Resume 操作，从磁盘读回 instance 的状态，使之继续运行。

这里需要对 Suspend 和 Pause 操作做个比较：

**相同点**两者都是暂停 instance 的运行，并保存当前状态，之后可以通过 Resume 操作恢复

**不同点**1. Suspend 将 instance 的状态保存在磁盘上；Pause 是保存在内存中，所以 Resume 被 Pause 的 instance 要比 Suspend 快。

2. Suspend 之后的 instance，其状态是 Shut Down；而被 Pause 的 instance 状态是Paused。

3. 虽然都是通过 Resume 操作恢复，Pause 对应的 Resume 在 OpenStack 内部被叫作 “Unpause”；Suspend 对应的 Resume 才是真正的 “Resume”。这个在日志中能体现出来。

Suspend/Resume 的日志分析留给大家做练习。

**Rescue/Unrescue**

从这节开始，我们将讨论几种 instance 故障恢复的方法，不同方法适用于不同的场景。

首先我们考虑操作系统故障。

有时候由于误操作或者突然断电，操作系统重启后却起不来了。

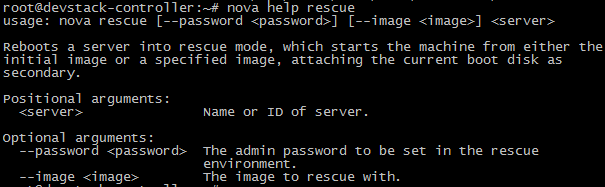
为了最大限度挽救数据，我们通常会使用一张系统盘将系统引导起来，然后在尝试恢复。

问题如果不太严重，完全可以通过这种方式让系统重新正常工作。

比如某个系统文件意外删除， root 密码遗忘等

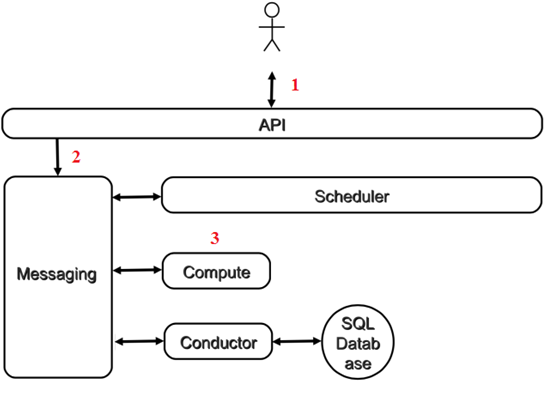
Nova 也提供了这种故障恢复机制，叫做 Rescue。

我们来看看 rescue 的说明：



Rescue 用指定的 image 作为启动盘引导 instance，将 instance 本身的系统盘作为第二个磁盘挂载到操作系统上。

下面是 rescue instance 的流程图

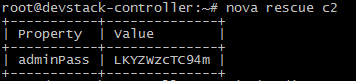


1. 向 nova-api 发送请求
2. nova-api 发送消息
3. nova-compute 执行操作

下面我们详细讨论每一个步骤。

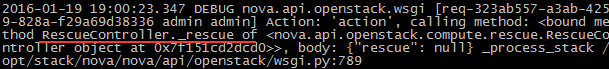
**向 nova-api 发送请求**

目前 Rescue 操作只能通过 CLI 执行



这里我们没有指明用哪个 image 作为引导盘，nova 将使用 instance 部署时使用的 image

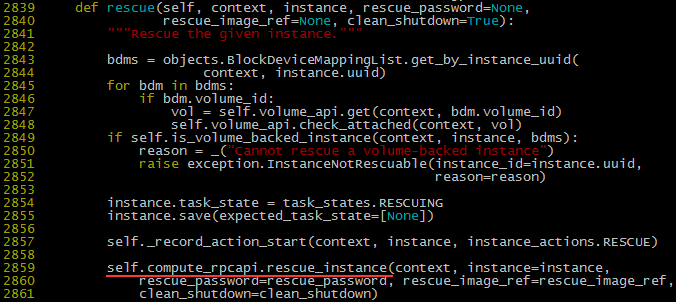
查看日志 /opt/stack/logs/n-api.log



**nova-api 发送消息**

nova-api 向 Messaging（RabbitMQ）发送了一条消息：“Rescue 这个 Instance”

源代码在 /opt/stack/nova/nova/compute/api.py，方法是 rescue。

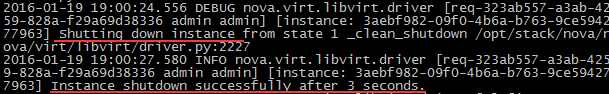


**nova-compute执行操作**

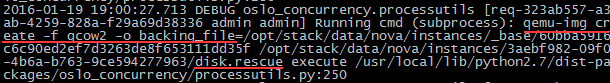
查看日志 /opt/stack/logs/n-cpu.log

**关闭 instance**

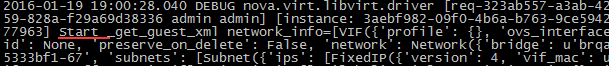
http://7xo6kd.com1.z0.glb.clouddn.com/upload-ueditor-image-20160517-1463495355335044496.jpg



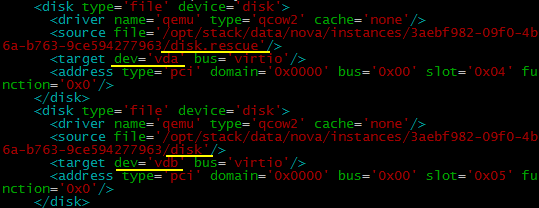
**通过 image 创建新的引导盘，命名为 disk.rescue**



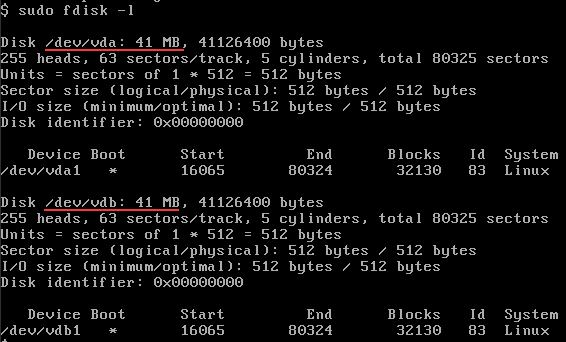
**启动 instance**



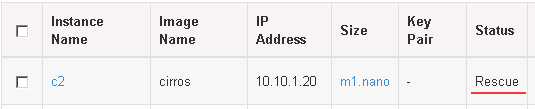
Rescue 执行成功后，可以通过 virsh edit <instance\_name> 查看 instance 的 XML 定义，disk.rescue 作为启动盘 vda，真正的启动盘 disk 作为第二个磁盘 vdb。



登录 instance，通过 fdisk 也可确认。

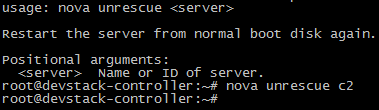


此时，instance 处于 Rescue 状态



Rescue 操作让我们有机会修复损坏的操作系统。

修好之后，使用 Unrescue 操作从原启动盘重新引导 instance。



Unrescue 的日志分析留给大家练习。