**nova-compute(openstack-ESSEX**)**工作流程**

请求：nova boot --image ttylinux --flavor 1 i-01  
nova-api 接受请求，一个tcp REST请求.  
nova-api 发送一个创建虚拟机的请求到消息队列，并会存数据库，带uuid.

nova-scheduler 接受这个消息，并进行过滤，根据请求的虚拟资源，即flavor的信息.  
scheduler会找到一个可用的主机(装有nova-compute的物理主机),如果没有找到就虚拟机的状态设置成ERROR，  
如果有可用主机，就发消息到nova-network，就进入下一步，配置网络，注：此过程虚拟机处于scheduling任务状态。

nova-network 接收到消息就，从fixed IP表(数据库)里拿出一个可用IP，  
并设置dnsmsq（DHCP server），确保拿出的IP可以与对应的MAC地址(生成的)对应，  
确保虚拟机可以被赋予对应的IP，并设置IPTABLE.  
对fixed IP 进行地址转换，使虚拟机可以访问外网，  
注:对于Vlan模式，nova-network所在的物理主机，要对应放置一个空闲的fixed到网桥上，  
为的是使物理主机和虚拟机处于同一个vlan中，这样NAT才会生效，会在之后的nova-network中详细讨论.

设置好network之后，会发消息到消息队列，使要在其上创建虚拟机的物理计算节点就收到创建虚拟机的消息，  
计算节点接收到消息后，就开始创建虚拟机，首先会download镜像从glance上(注：如果有此镜像的缓存，这不需要download，

这也是为什么第二次在同一台物理主机上创建虚拟机会非常快)，然后会根据之前生成的uuid，MAC,镜像位置，

创建一个启动虚拟机的xml文件，然后会调用libvirt接口，根据xml配置创建虚拟机，虚拟机创建完成之后，

会把虚拟机状态改成ACTIVE，至此，一台虚拟机发布完成

**一些经验**  
每次重启nova-compute会把所在主机上的虚拟机全部重启一遍，并会根据对应数据库中虚拟机的状态进行改变。

如果数据库中丢失了某条虚拟机的数据/虚拟为得了ted，会把对应不匹配的虚拟的删除。

另：nova-volume会有周期性遍历，把于数据库不匹配的vg下的lv删除.

虚拟机的运行虚拟硬盘默认放置在/var/lib/nova/instances下面.  
其中\_base文件夹中放置的虚拟机对应镜像的基本镜像，即初始镜像，而对应的  
instance000000xxx中对应的是虚拟机差异文件，在instance0000xxx中用  
kvm-img info disk 可以清晰看出两者的关系，其实这样做最大的好处是当有第二台相同镜像的虚拟机，可以直接使用\_base下的初始镜像，

省去了下载镜像的这一大块时间。

可以通过kvm-img convert命令把两者合并成一个镜像，openstack的snapshot（注：非volume-snapshot）就是这么做的，  
另:openstack的这个snapshot其实是一个backup，而不算是一个snapshot，个人觉得，  
可以发现kvm-img snapshot这个命令，目前不太了解为什么不使用这个，而使用完全备份。也许更可靠吧，但也非常占硬盘。

关于floating ip的添加，首先会把floating  
ip绑定到对应的网卡口上，根据你的配置，然后会做NAT对应的fixed  
IP,请求通过物理主机绑定的floating ip到你的fxied  
ip，到虚拟机，中间会有一些规则，根据你secgroup的设置，只接受允许的协议和端口。

openstack的novnc是使用HTML5的websocket协议，所以如果所在网络有限制，很可能不能访问，当然如果同在一个局域网另说。

关于制定义filter机制，在nova开发文档中有详细说明，在下面的nova.conf配置中也有对应说明

openstack日志都在/var/log/nova下面，要注意的是虚拟机的创建日志会在对应所在物理机的log中。

openstack平台中服务的状态(nova-manage service list)是通过计算数据库中对应服务字段的update  
time，用now-updatetime，是否大于更新频率来确定服务是否down了的，所以，如果在多主机openstack环境下，发现运行nova-manage  
service list时，本机的服务都是正常，其他主机的服务都是down的话，先确保主机间是否时间同步了

使用kvm-ok来确认是否支持kvm，主要是来确认是否支持windows系统，对于qemu是没有virtio模式的。

虚拟机真正的运行配置是在/etc/libvirt/qemu下面的，所以如果想手动改其配置，应该改这里的

**关于nova.conf的一些配置**

[复制代码](javascript:void(0);)

# DESC:

# --vlan\_interface=eth0 ,Configure LAN IP On this device

# --public\_interface=eth1 ,Configure public IP on this device

# IF Only have one device , ALL use eth0 ,

# --vlan\_interface=eth0

# --public\_interface=eth0

# --my\_ip=[this host's IP]

# --sql\_connection=mysql://openstack:hisoft@[the controller's IP]/openstack

# --rabbit\_host=[the controller's IP]

# --glance\_api\_servers=[the controller's IP]

# --ec2\_dmz\_host=[the controller's IP]

# --ec2\_host=[the controller's IP]

# --instances\_path=[the/path/to/store/instances] #replace default path /var/bin/nova/instances

# if use flat DHCP for flatDHCP netwokor config

# --network\_manager=nova.network.manager.FlatDHCPManager

# --public\_interface=eth0

# --flat\_interface=eth0

# --flat\_network\_bridge=br100

# --fixed\_range=10.0.0.0/27

# --network\_size=32

#

( cat | sudo tee /etc/nova/nova.conf ) <<EOF

--dhcpbridge\_flagfile=/etc/nova/nova.conf

--dhcpbridge=/usr/bin/nova-dhcpbridge

--logdir=/var/log/nova

--state\_path=/var/lib/nova

--lock\_path=/var/lock/nova

--force\_dhcp\_release

--iscsi\_helper=tgtadm

--libvirt\_use\_virtio\_for\_bridges

--connection\_type=libvirt

--root\_helper=sudo nova-rootwrap

--verbose

--ec2\_private\_dns\_show\_ip

--network\_manager=nova.network.manager.VlanManager

--fixed\_range=10.0.0.0/8

--vlan\_interface=eth0

--public\_interface=eth0

--fixed\_range=10.0.0.0/27

--network\_size=32

--auth\_strategy=keystone

--my\_ip=[host ip]

--sql\_connection=mysql://openstack:hisoft@127.0.0.1/openstack

--rabbit\_host=[host ip]

--glance\_api\_servers=[host ip]:9292

--ec2\_dmz\_host=[host ip]

--ec2\_host=[host ip]

# the follow 6 config is for change the default quota of one tenant

--quota\_cores=200

--quota\_floating\_ips=50

--quota\_gigabytes=3000

--quota\_instances=100

--quota\_ram=300000

--quota\_volumes=100

--novncproxy\_base\_url=http://[controller ip]:6080/vnc\_auto.html

--vncserver\_proxyclient\_address=[host ip]

--vncserver\_listen=[host ip]

EOF

[复制代码](javascript:void(0);)

#IF use customize scheduler filter , add the follow config .

( cat | sudo tee -a /etc/nova/nova.conf ) <<EOF

--scheduler\_driver=nova.scheduler.filter\_scheduler.FilterScheduler

--scheduler\_available\_filters=[your filter impl import path]

--scheduler\_default\_filters=ResourceFilter

EOF