

本站文章大部分为作者原创，非商业用途转载无需作者授权，但务必在文章标题下面注明作者 刘世民 (Sammy Liu) 以及可点击的
本博客地址超级链接 <http://www.cnblogs.com/sammyliu/> , 谢谢合作



www.cnblogs.com

世民谈云计算

(声明 : 本站文章皆基于公开来源信息 , 仅代表作者个人观点 , 与作者所在公司无
关)

昵称 : SammyLiu
园龄 : 2年6个月
荣誉 : 推荐博客
粉丝 : 470
关注 : 30
[+加关注](#)

<	2017年5月						>
日	一	二	三	四	五	六	
30	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	31	1	2	3	
4	5	6	7	8	9	10	

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

我的标签

[GRE\(1\)](#)
[Neutron\(1\)](#)
[Open vSwitch\(1\)](#)
[OpenStack\(1\)](#)

随笔分类(254)

[Ceilometer\(3\)](#)
[Ceph\(13\)](#)
[Cinder\(6\)](#)
[Docker\(8\)](#)
[Glance](#)
[Heat\(2\)](#)
[K8S](#)
[Keystone\(1\)](#)
[KVM\(10\)](#)
[MessageQueue\(4\)](#)
[MySQL\(1\)](#)
[Neutron\(17\)](#)
[Nova\(10\)](#)
[OpenStack\(33\)](#)

[博客园](#) [首页](#) [新随笔](#) [订阅XML](#) [管理](#)

随笔-121 评论-504 文章-45

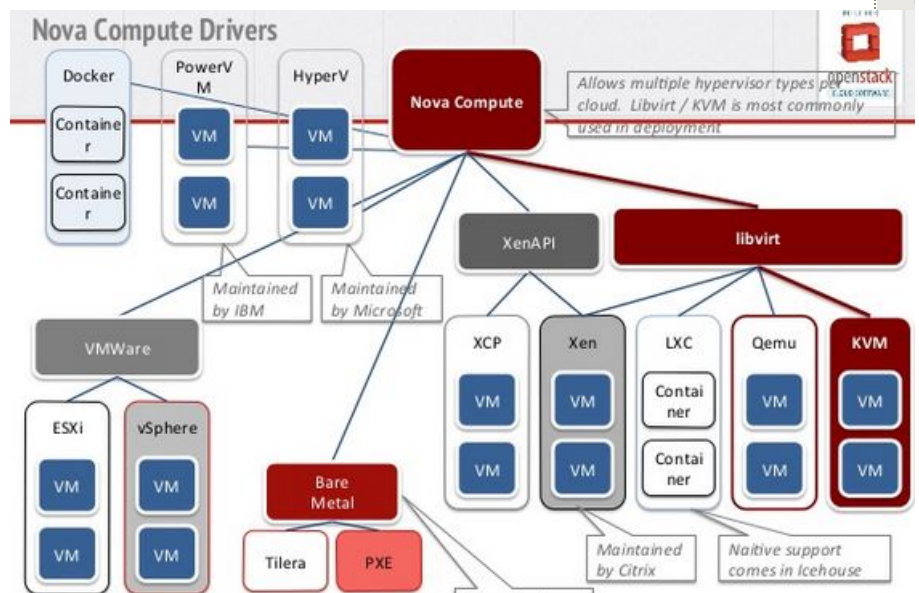
KVM 介绍 (6) : Nova 通过 libvirt 管理 QEMU/KVM 虚拟机 [Nova Libvirt QEMU/KVM Domain]

学习 KVM 的系列文章 :

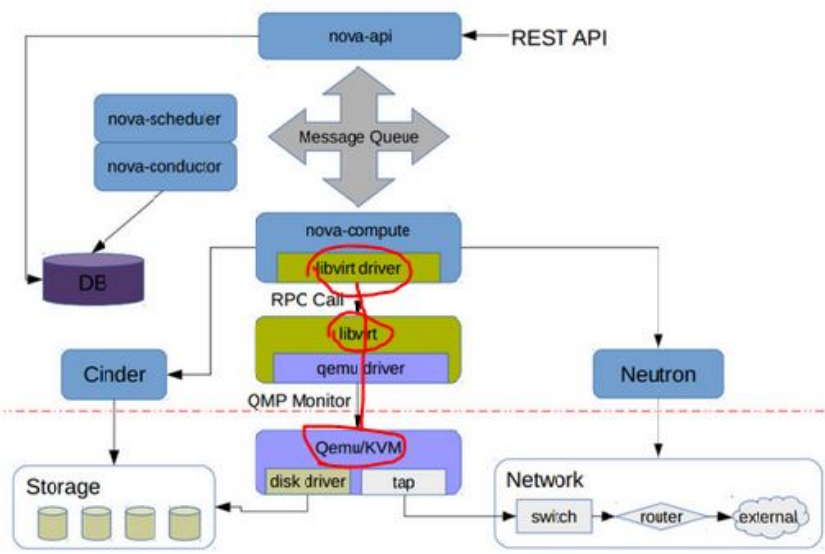
- (1) 介绍和安装
- (2) CPU 和 内存虚拟化
- (3) I/O QEMU 全虚拟化和准虚拟化 (Para-virtualization)
- (4) I/O PCI/PCIe设备直接分配和 SR-IOV
- (5) libvirt 介绍
- (6) Nova 通过 libvirt 管理 QEMU/KVM 虚拟机
- (7) 快照 (snapshot)
- (8) 迁移 (migration)

1. Libvirt 在 OpenStack 架构中的位置

在 Nova Compute 节点上运行的 nova-compute 服务调用 Hypervisor API 去管理运行在该 Hypervisor 的虚拟机。Nova 使用 libvirt 管理 QEMU/KVM 虚拟机 , 还使用别的 API 去管理别的虚拟机。



Sahara
Storage(1)
Swift(3)
Trove
Ubuntu(3)
VMware(3)
安装和配置(1)
版本(4)
备份(1)
大数据(5)
翻译(4)
高可用 (HA) (6)
基础知识(19)
监控(1)
容器(4)
容器编排
使用案例(4)
网络(8)
问题定位(3)
行业(14)
性能(4)
虚拟化(7)
原理(22)
云Cloud(29)
随笔档案(121)
2017年5月 (1)
2017年3月 (1)
2017年1月 (1)
2016年10月 (7)
2016年9月 (5)
2016年8月 (4)
2016年7月 (1)
2016年6月 (5)
2016年5月 (1)
2016年4月 (1)
2016年3月 (9)
2016年2月 (4)
2016年1月 (2)
2015年12月 (7)
2015年11月 (7)
2015年10月 (4)
2015年9月 (4)
2015年8月 (5)
2015年7月 (9)
2015年6月 (10)
2015年5月 (3)



libvirt 的实现代码在 /nova/virt/libvirt/driver.py 文件中。

这里是 [OpenStack Hypervisor Matrix](#)。

这里是 [每个 Linux 发行版里面 libvirt , QEMU/KVM 的版本号](#)。

请注意Juno 版本 Nova 对 libvirt 和 QEMU 的各种最低版本要求：

功能	最低 libvirt 版本	最低 QEMU 版本	不支持的后果
所有	0.9.11		Nova 不能使用 libvirt driver
支持 device callback	1.1.1		不支持的话, 就无法支持 Detach PCI/SR-IOV 设备
Live snapshot	1.3.0	1.3.0	只能使用 Clod Snapshot
挂载卷时设置卷的 block 大小 (Block IO)	0.10.2		不能使用的话, 就不能设置卷的特定 block size, 只能使用其默认的 block size。
Block Job Info	1.1.1		不能在线删除卷的快照 (online deletion of volume snapshots)
Discard	1.0.6	1.6.0	不支持 image 设置 hw_disk_discard 属性, 具体参考 Blueprint
NUMA topology	1.0.4		无法获取 node 的 NUMA topology 信息, 就无法将虚拟机的 vCPU 指定到特定的 node CPU 上, 会影响虚拟机的性能

2. Nova 中 libvirt 的使用

Nova 使用 libvirt 来管理虚拟机，包括：

- 创建虚拟机
- 虚拟机的生命周期管理（[参考这篇文档](#)）
- 添加和删除连接到别的网络的网卡（interface）
- 添加和删除 Cinder 卷（volume）

2.1 创建 QEMU/KVM 虚拟机

2015年4月 (11)
 2015年3月 (2)
 2015年2月 (6)
 2015年1月 (5)
 2014年12月 (6)

文章分类(21)

Ceph(1)
 GlusterFS
 Web 服务器(2)
 操作系统(1)
 存储
 大数据(2)
 分布式系统
 服务器(1)
 网络(11)
 虚拟化(3)
 云

文章档案(42)

2016年10月 (2)
 2016年9月 (1)
 2016年6月 (1)
 2016年5月 (3)
 2015年12月 (4)
 2015年10月 (5)
 2015年9月 (2)
 2015年6月 (1)
 2015年4月 (23)

积分与排名

积分 - 286831
 排名 - 535

最新评论

1. Re:Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network]
 eth1 - 公共网络 (untagged), 管理网络 (tag=102), 存储网络 (tag=103) 不好意思, 大家共用同一个eth1端口的时候, 请问这里交换机端口是配置为tagged还是untagged.....
 --xianke9
2. Re:理解Docker (5) : Docker 网络
 1.12版本上网络的表现如何?
 --幽灵狼
3. Re:理解Docker (5) : Docker 网络

创建虚拟机的配置有几个来源:

- 用户的选项, 包括虚拟机的基本信息, 比如 name, flavor, image, network, disk等。
- image 的属性, 比如 hw_vif_model, hw_scsi_model 等。完整的供 libvirt API 使用的属性列表 [在这里](#)。
- 管理员在 nova.conf 中的配置

(注意: image 的元数据属性的优先级高于 nova.conf 中的配置。只有在没有property的情况下才使用nova.conf中的配置)

创建虚拟机的过程的几个主要阶段:

(1) 消息由 nova-api 路由到某个 nova compute 节点 (API -> Scheduler -> Compute (manager) -> Libvirt Driver)

(2) 调用 Neutron REST API 去准备网络。其返回的数据类似:

```
[VIF({'profile': {}, 'ovs_interfaceid': u'59cfa0b8-2f5c-481a-89a8-7a8711b368a2', 'network': Network({'bridge': 'br-int', 'subnets': [Subnet({'ips': [FixedIP({'meta': {}, 'version': 4, 'type': 'fixed', 'floating_ips': [], 'address': u'10.0.10.14'})]}], 'version': 4, 'meta': {'dhcp_server': u'10.0.10.11', 'dns': [], 'routes': [], 'cidr': u'10.0.10.0/24', 'gateway': IP({'meta': {}, 'version': 4, 'type': 'gateway', 'address': u'10.0.10.1'})}]}], 'meta': {'injected': False, 'tenant_id': u'74c8ada23a3449f888d9e19b76d13aab'}, 'id': u'a924e87a-826b-4109-bb03-523a8b3f6f9e', 'label': u'demo-net2'}, 'devname': u'tap59cfa0b8-2f', 'vnic_type': u'normal', 'qbg_params': None, 'meta': {}, 'details': {'port_filter': True, 'ovs_hybrid_plug': True, 'address': u'fa:16:3e:e0:30:e7', 'active': False, 'type': u'ovs', 'id': u'59cfa0b8-2f5c-481a-89a8-7a8711b368a2', 'qbg_params': None})]
```

(3) 从 image 启动话, nova 会调用 Glance REST API 后者 image metadata 和准备本地启动盘

image metadata:

```
{u'status': u'active', u'deleted': False, u'container_format': u'bare', u'min_ram': 0, u'updated_at': u'2015-04-26T04:34:40.000000', u'min_disk': 0, u'owner': u'74c8ada23a3449f888d9e19b76d13aab', u'is_public': False, u'deleted_at': None, u'properties': {}, u'size': 13167616, u'name': u'image', u'checksum': u'64d7c1cd2b6f60c92c14662941cb7913', u'created_at': u'2015-04-26T04:34:39.000000', u'disk_format': u'qcow2', u'id': u'bb9318db-5554-4857-a309-268c6653b9ff'}
```

本地启动盘:

```
{'disk_bus': 'virtio', 'cdrom_bus': 'ide', 'mapping': {'disk': {'bus': 'virtio', 'boot_index': '1', 'type': 'disk', 'dev': u'vda'}, 'root': {'bus': 'virtio', 'boot_index': '1', 'type': 'disk', 'dev': u'vda'}, 'disk.local': {'bus': 'virtio', 'type': 'disk', 'dev': 'vdb'}, 'disk.swap': {'bus': 'virtio', 'type': 'disk', 'dev': 'vdc'}}
```

本地启动盘的文件信息:

我想请问一下运行docker quickstart terminal时一直卡在“waiting for an IP”应该如何解决呢？希望楼主能解答一下。

--silentbell

4. Re:理解Docker (6) : 若干企业生产环境中的容器网络方案写得真好！加油。

--itbj00

5. Re:理解Docker (5) : Docker网络

非常好，写得很详细。加油！

--itbj00

阅读排行榜

1. Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network](22087)
2. 理解 OpenStack 高可用 (HA) (1) : OpenStack 高可用和灾备方案 [OpenStack HA and DR](13707)
3. Neutron 理解 (3): Open vSwitch + GRE/VxLAN 组网 [Netruon Open vSwitch + GRE/VxLAN Virutal Network](13434)
4. 探索 OpenStack 之 (9) : 深入块存储服务 Cinder (功能篇) (12921)
5. 理解 OpenStack + Ceph (1) : Ceph + OpenStack 集群部署和配置 (12444)

评论排行榜

1. Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network](63)
2. Neutron 理解 (14) : Neutron ML2 + Linux bridge + VxLAN 组网 (54)
3. Neutron 理解 (8): Neutron 是如何实现虚拟机防火墙的 [How Neutron Implements Security Group](34)
4. Neutron 理解 (3): Open vSwitch + GRE/VxLAN 组网 [Netruon Open vSwitch + GRE/VxLAN Virutal Network](25)
5. Neutron 理解 (5) : Neutron 是如何向 Nova 虚拟机分配固定IP地址的 (How Neutron Allocates Fixed IPs to Nova Instance) (21)

推荐排行榜

1. Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network](9)
2. 我所了解的 京东、携程、eBay、小米的 OpenStack 云(9)
3. 理解 OpenStack 高可用 (HA)



```
root@compute2:/home/s1# qemu-img info
/var/lib/nova/instances/02699155-940f-4401-bc01-36220db80639/disk.local
image: /var/lib/nova/instances/02699155-940f-4401-bc01-36220db80639/disk.local
file format: qcow2
virtual size: 1.0G (1073741824 bytes) #由 flavor.ephemeral_disk 指定其 size
disk size: 324K
cluster_size: 65536
backing file:
/var/lib/nova/instances/_base/ephemeral_1_default
Format specific information:
  compat: 1.1
  lazy refcounts: false

root@compute2:/home/s1# qemu-img info
/var/lib/nova/instances/02699155-940f-4401-bc01-36220db80639/disk.swap
image: /var/lib/nova/instances/02699155-940f-4401-bc01-36220db80639/disk.swap
file format: qcow2
virtual size: 30M (31457280 bytes) # 由 flavor.swap_disk 指定其size
disk size: 196K
cluster_size: 65536
backing file: /var/lib/nova/instances/_base/swap_30
Format specific information:
  compat: 1.1
  lazy refcounts: false
```



(4) 根据这些信息，生成 domain xml，然后生成其配置使得它是一个持久性虚拟机（调用 libvirt Python DefineXML API）。

一个从 image 启动的 Domain 的配置 XML 实例（蓝色部分是注释说明）：



```
<domain type="qemu">
  <uuid>8352e969-0a25-4abf-978f-d9d0ec4de0cd</uuid>
  <name>instance-0000002f</name>
  <memory>51200</memory> # guest.memory = flavor.memory_mb
  * units.Ki 即 50 * 1024 = 51200
  <vcpu cpuset="0">1</vcpu> #flavor.vcpus
  <metadata>
    <nova:instance
xmlns:nova="http://openstack.org/xmlns/libvirt/nova/1.0">
      <nova:package version="2014.2.2"/>
      <nova:name>vm11</nova:name> #input.name
      <nova:creationTime>2015-06-09
23:54:04</nova:creationTime>
      <nova:flavor name="tiny2"> #input.flavor
        <nova:memory>50</nova:memory>
        <nova:disk>1</nova:disk>
        <nova:swap>30</nova:swap>
        <nova:ephemeral>1</nova:ephemeral>
        <nova:vcpus>1</nova:vcpus>
      </nova:flavor>
```


(1) : OpenStack 高可用和灾备方案 [OpenStack HA and DR](6)
 4. Neutron 理解 (2): 使用 Open vSwitch + VLAN 组网 [Neutron Open vSwitch + VLAN Virtual Network](6)
 5. 理解 OpenStack 高可用 (HA)
 (2) : Neutron L3 Agent HA 之虚拟路由冗余协议 (VRRP) (5)

```
<nova:owner>
  <nova:user
uuid="bcd37e6272184f34993b4d7686ca4479">admin</nova:user>
    <nova:project
uuid="74c8ada23a3449f888d9e19b76d13aab">admin</nova:project>
  </nova:owner>
  <nova:root type="image" uuid="bb9318db-5554-4857-a309-268c6653b9ff"/> #input.source
</nova:instance>
</metadata>
<sysinfo type="smbios"> # Nova 中写死的
  <system>
    <entry name="manufacturer">OpenStack
Foundation</entry>
    <entry name="product">OpenStack Nova</entry>
    <entry name="version">2014.2.2</entry>
    <entry name="serial">03bb1a0f-ae04-4765-9f3c-d200a2540675</entry>
    <entry name="uuid">8352e969-0a25-4abf-978f-d9d0ec4de0cd</entry>
  </system>
</sysinfo>
<os>
  <type>hvm</type> #表示 Guest OS 需要 full virtualization 支持
  <boot dev="hd"/> #指定启动盘
  <smbios mode="sysinfo"/> #去读取 <sysinfo> 的定义
</os>
<features>
  <acpi/> # Soft Reboot 需要 ACPI 的支持, 否则只能使用 Hard
reboot。 https://bugs.launchpad.net/horizon/+bug/1346741
  <apic/> # 没 APIC 的话, Windows Guest 会在 Xen 或者 KVM
上崩溃。 https://bugs.launchpad.net/nova/+bug/1086352
</features>
  <clock offset="utc"/> #如果Guest OS 是 MS, 则是
localtime, 否则都是 utc
  <cpu mode="host-model" match="exact"> # 对于 KVM, 如果
CONF.libvirt.cpu_mode 是 none, mode 则设为 "host-model"。具体
可参考 https://wiki.openstack.org/wiki/LibvirtXMLCPUModel
    <topology sockets="1" cores="1" threads="1"/> #默认的时候,
sockets 数目设为 vcpu 的数目, cores 和 threads 都设为 1. 可以
通过设置 image 的 hw_cpu_topology 属性来改变这里的设置, 具体请
参考 https://blueprints.launchpad.net/nova/+spec/support-libvirt-vcpu-topology 以及
https://wiki.openstack.org/wiki/VirtDriverGuestCPUMemoryPlacement
  </cpu>
<devices>
  <disk type="file" device="disk"> # 从 image 启动时候的启动盘 (flavor.root_disk)
    <driver name="qemu" type="qcow2" cache="none"/>
    <source file="/var/lib/nova/instances/8352e969-0a25-4abf-978f-d9d0ec4de0cd/disk"/>
    <target bus="virtio" dev="vda"/> #对于 KVM, disk 的
bus 为 "virtio", cdrom 的 bus 为 "ide", floppy 的 bus 为 "fdc"
  </disk>
  <disk type="file" device="disk"> #临时分区 (flavor.ephemeral_disk)
    <driver name="qemu" type="qcow2" cache="none"/>
```

```

<source file="/var/lib/nova/instances/8352e969-0a25-4abf-978f-d9d0ec4de0cd/disk.local"/>
<target bus="virtio" dev="vdb"/>
</disk>
<disk type="file" device="disk"> #swap 分区
(flavor.swap_disk)
<driver name="qemu" type="qcow2" cache="none"/>
<source file="/var/lib/nova/instances/8352e969-0a25-4abf-978f-d9d0ec4de0cd/disk.swap"/>
<target bus="virtio" dev="vdc"/>
</disk>
<interface type="bridge"> # 虚拟机通过网桥连接到 OVS
<mac address="fa:16:3e:e0:30:e7"/>
<model type="virtio"/> #该 type 可以由 image metadata
hw_vif_type 指定。未指定的话,如果配置了
conf.libvirt.use_virtio_for_bridges = true (默认就是 true)的
话, QEMU/KVM 会使用 virtio 类型。
<driver name="qemu"/>
<source bridge="qbr59cfa0b8-2f"/> #qbr59cfa0b8-2f 连
接虚拟机的 vNIC tap59cfa0b8-2f 和 qvb59cfa0b8-2f ,而
qvb59cfa0b8-2f 连到 OVS 的 br-int 上。
<target dev="tap59cfa0b8-2f"/>
</interface>
<serial type="file"> 当 CONF.serial_console.enabled =
true 时, type 为 "tcp", 使用 config 配置, 其 XML 为 <serial
type="tcp"> <source host="127.0.0.1" mode="bind"
service="10000"/> </serial>; 当为 false 时, 使用
console.log 文件, type 为 file。
<source path="/var/lib/nova/instances/8352e969-0a25-4abf-978f-d9d0ec4de0cd/console.log"/>
</serial>
<serial type="pty"/> #每个domain都有 type 为 "pty" 的
serial 配置。
<input type="tablet" bus="usb"/> #当 CONF.vnc_enabled
= true 或者 CONF.spice.enabled = true 并且
CONF.spice.agent_enabled = false 时添加 tablet, type 和 bus
都是固定的。
<graphics type="vnc" autoport="yes" keymap="en-us"
listen="0.0.0.0"/> #如果 CONF.vnc_enabled = true ,那么
keymap=CONF.vnc_keymap ; listen=CONF.vncserver_listen
<video> #如果 CONF.vnc_enabled 或者 CONF.spice.enabled ,
则添加该 video 配置
<model type="cirrus"/> #如果 CONF.spice.enabled ,则
type 为 qxl ; 否则为 cirrus。
</video>
<memballoon model="virtio"> #如果
CONF.libvirt.mem_stats_period_seconds >0 则添加
memballoon ; 对 KVM , model 固定为 "virtio"
<stats period="10"/>
</memballoon>
</devices>
</domain>

```

从 bootable volume 启动的话, disk 部分为:

```

<disk type="file" device="disk">
<driver name="qemu" type="qcow2" cache="none"/>

```

```

<source file="/var/lib/nova/instances/02699155-940f-4401-bc01-36220db80639/disk.local"/>
<target bus="virtio" dev="vdb"/>
</disk>
<disk type="file" device="disk">
  <driver name="qemu" type="qcow2" cache="none"/>
  <source file="/var/lib/nova/instances/02699155-940f-4401-bc01-36220db80639/disk.swap"/>
  <target bus="virtio" dev="vdc"/>
</disk>
<disk type="block" device="disk">
  <driver name="qemu" type="raw" cache="none"/>
  <source dev="/dev/disk/by-path/ip-10.0.2.41:3260-iscsi-iqn.2010-10.org.openstack:volume-26446902-5a56-4c79-b839-a8e13a66dc7a-lun-1"/>
  <target bus="virtio" dev="vda"/>
  <serial>26446902-5a56-4c79-b839-a8e13a66dc7a</serial>
</disk>

```



(5). 启动 domain (调用 libvirt Python createWithFlags API)

2.2 添加 volume 到虚拟机 (nova volume-attach)

(1) 使用 volume id 通过 volume driver 找到指定的 volume

(2) 调用 volume driver 来建立主机和 Volume 之间的连接

主机信息为：

```
{'ip': '192.168.1.15', 'host': 'compute2', 'initiator': 'iqn.1993-08.org.debian:01:a9f2b45c24f9'}
```

建立的 iSCSI 连接信息为：

```
{u'driver_volume_type': u'iscsi', u'data':
{u'access_mode': u'rw', u'target_discovered': False,
u'encrypted': False, u'qos_specs': None, u'target_iqn':
u'iqn.2010-10.org.openstack:volume-51da0d1f-0a17-4e7f-aeff-27438963348a', u'target_portal': u'10.0.2.41:3260',
u'volume_id': u'51da0d1f-0a17-4e7f-aeff-27438963348a',
u'target_lun': 1, u'auth_password':
u'hXG64qrzEjNt8MDKnERA', u'auth_username':
u'fKSAe6vhgyeG88U9kcBV', u'auth_method': u'CHAP'}}
```

volume 在主机上的磁盘为：



```

root@compute2:/home/s1# ls /dev/disk/by-path/ -ls
total 0
0 lrwxrwxrwx 1 root root 9 Jun 10 12:18 ip-10.0.2.41:3260-iscsi-iqn.2010-10.org.openstack:volume-51da0d1f-0a17-4e7f-aeff-27438963348a-lun-1 -> ../../sdc

```

```

Disk /dev/sdc: 1073 MB, 1073741824 bytes
34 heads, 61 sectors/track, 1011 cylinders, total 2097152 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes

```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/sdc doesn't contain a valid partition table
```



(3) 通过 domain name 来找到指定 domain 对象 (通过调用 lookupByName API)

(4) 生成 volume 连接的配置 xml, 比如:



```
<disk type="block" device="disk">
  <driver name="qemu" type="raw" cache="none"/>
  <source dev="/dev/disk/by-path/ip-10.0.2.41:3260-iscsi-iqn.2010-10.org.openstack:volume-51da0d1f-0a17-4e7f-aeff-27438963348a-lun-1"/>
  <target bus="virtio" dev="vdd"/>
  <serial>51da0d1f-0a17-4e7f-aeff-27438963348a</serial>
</disk>
```



(5) 调用 attachDeviceFlags API 将 volume 挂载到该虚拟机

2.3 添加连接到新的网络的 interface 给虚拟机 (nova interface-attach)

(1) 运行 nova interface-attach, 传入 network-id, Neutron 会分配如下 network info 给 Nova

```
VIF({'profile': {}, 'ovs_interfaceid': u'0142efee-7382-43ef-96e8-d0084ecc893c', 'network': Network({'bridge': u'br-int', 'subnets': [Subnet({'ips': [FixedIP({'meta': {}, 'version': 4, 'type': u'fixed', 'floating_ips': [], 'address': u'10.0.0.40'})], 'version': 4, 'meta': {'dhcp_server': u'10.0.0.3'}, 'dns': [], 'routes': [], 'cidr': u'10.0.0.0/24', 'gateway': IP({'meta': {}, 'version': 4, 'type': u'gateway', 'address': u'10.0.0.1'})}], 'meta': {'injected': False, 'tenant_id': u'74c8ada23a3449f888d9e19b76d13aab'}, 'id': u'01630966-b21f-4a6d-95ff-10c4575f1fe2', 'label': u'demo-net'}, 'devname': u'tap0142efee-73', 'vnic_type': u'normal', 'qbg_params': None, 'meta': {}, 'details': {'port_filter': True, 'ovs_hybrid_plug': True}, 'address': u'fa:16:3e:14:32:d9', 'active': True, 'type': u'ovs', 'id': u'0142efee-7382-43ef-96e8-d0084ecc893c', 'qbg_params': None})
```

(2) 执行下面的命令, 将 Neutron 分配的 port 连接到 OVS



```
#添加 linux bridge
brctl addbr qbr0142efee-73 #名字是 devname 的后半部分
brctl setfd qbr0142efee-73 0
brctl stp qbr0142efee-73 off
tee /sys/class/net/qbr0142efee-73/bridge/multicast_snooping
```



```
ip link add qvb0142efee-73 type veth peer name
qvo0142efee-73
ip link set qvb0142efee-73
ip link set qvb0142efee-73 promisc on

#在 ovs 上添加端口
ovs-vsctl --timeout=120 -- --if-exists del-port
qvo0142efee-73 -- add-port br-int qvo0142efee-73 -- set
Interface qvo0142efee-73 external-ids:iface-id=0142efee-
7382-43ef-96e8-d0084ecc893c external-ids:iface-
status=active external-ids:attached-mac=fa:16:3e:14:32:d9
external-ids:vm-uuid=8352e969-0a25-4abf-978f-d9d0ec4de0cd
```

(3) 生成 interface 配置的xml, 比如:

```
<interface type="bridge">
  <mac address="fa:16:3e:14:32:d9"/>
  <model type="virtio"/>
  <driver name="qemu"/>
  <source bridge="qbr0142efee-73"/>
  <target dev="tap0142efee-73"/>
</interface>
```

(4) 调用 attachDeviceFlags API 来挂载该 interface 到虚拟机

至于其他的虚拟机操作, 会在另一篇文章中描述。

分类: [KVM,Nova](#)

好文要顶

关注我

收藏该文



SammyLiu

关注 - 30

粉丝 - 470

荣誉: 推荐博客

+加关注

2

推荐

0

反对

« 上一篇: [IBM云的商业动作之我见 \(1\) : IBM 收购 OpenStack 托管私有云公司 Blue Box \[IBM Acquired Blue Box\]](#)

» 下一篇: [Nova: 虚拟机的块设备总结 \[Nova Instance Block Device\]](#)

posted on 2015-06-11 17:57 [SammyLiu](#) 阅读(4262) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【报表】Excel 报表开发18 招式，人人都能做报表

【活动】阿里云海外云服务全面降价助力企业全球布局

【实用】40+篇云服务器操作及运维基础知识！



最新IT新闻:

- [知乎上线视频功能，以后看教程更方便了](#)
 - [一年只赚2万元：乐视游戏或被出售](#)
 - [Unity获得4亿美元投资，现估值为26亿美元](#)
 - [直播对陌陌的意义，就像王者荣耀之于腾讯游戏](#)
 - [死磕支付宝？苏宁金融发布“星辰计划”：扫码支付返888元](#)
- » [更多新闻...](#)



最新知识库文章:

- [程序员的工作、学习与绩效](#)
 - [软件开发为什么很难](#)
 - [唱吧DevOps的落地，微服务CI/CD的范本技术解读](#)
 - [程序员，如何从平庸走向理想？](#)
 - [我为什么鼓励工程师写blog](#)
- » [更多知识库文章...](#)

Powered by: [博客园](#) 模板提供: [沪江博客](#) Copyright ©2017 SammyLiu