**openstack网络模式之vlan分析**

**openstack neutron中定义了四种网络模式：**

# tenant\_network\_type = local

# tenant\_network\_type = vlan

# Example: tenant\_network\_type = gre

# Example: tenant\_network\_type = vxlan

**本文主要以vlan为例，并结合local来详细的分析下openstack的网络模式。**

**1. local模式**

此模式主要用来做测试，只能做单节点的部署(all-in-one)，这是因为此网络模式下流量并不能通过真实的物理网卡流出，即neutron的integration bridge并没有与真实的物理网卡做mapping，只能保证同一主机上的vm是连通的，具体参见RDO和neutron的配置文件。

**(1)RDO配置文件(answer.conf)**

主要看下面红色的配置项，默认为空。

CONFIG\_NEUTRON\_OVS\_BRIDGE\_MAPPINGS

openswitch默认的网桥的映射到哪，即br-int映射到哪。 正式由于br-int没有映射到任何bridge或interface，所以只能br-int上的虚拟机之间是连通的。

CONFIG\_NEUTRON\_OVS\_BRIDGE\_IFACES

流量最后从哪块物理网卡流出配置项

# Type of network to allocate for tenant networks (eg. vlan, local,

# gre)

CONFIG\_NEUTRON\_OVS\_TENANT\_NETWORK\_TYPE=local

# A comma separated list of VLAN ranges for the Neutron openvswitch

# plugin (eg. physnet1:1:4094,physnet2,physnet3:3000:3999)

CONFIG\_NEUTRON\_OVS\_VLAN\_RANGES=

# A comma separated list of bridge mappings for the Neutron

# openvswitch plugin (eg. physnet1:br-eth1,physnet2:br-eth2,physnet3

# :br-eth3)

CONFIG\_NEUTRON\_OVS\_BRIDGE\_MAPPINGS=

# A comma separated list of colon-separated OVS bridge:interface

# pairs. The interface will be added to the associated bridge.

CONFIG\_NEUTRON\_OVS\_BRIDGE\_IFACES=

**(2)neutron配置文件(/etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini)**

[ovs]

# (StrOpt) Type of network to allocate for tenant networks. The

# default value 'local' is useful only for single-box testing and

# provides no connectivity between hosts. You MUST either change this

# to 'vlan' and configure network\_vlan\_ranges below or change this to

# 'gre' or 'vxlan' and configure tunnel\_id\_ranges below in order for

# tenant networks to provide connectivity between hosts. Set to 'none'

# to disable creation of tenant networks.

#

tenant\_network\_type = local

RDO会根据answer.conf中local的配置将neutron中open vswitch配置文件中配置为local

**2. vlan模式**

大家对vlan可能比较熟悉，就不再赘述，直接看RDO和neutron的配置文件。

**(1)RDO配置文件**

# Type of network to allocate for tenant networks (eg. vlan, local,

# gre)

CONFIG\_NEUTRON\_OVS\_TENANT\_NETWORK\_TYPE=vlan //指定网络模式为vlan

# A comma separated list of VLAN ranges for the Neutron openvswitch

# plugin (eg. physnet1:1:4094,physnet2,physnet3:3000:3999)

CONFIG\_NEUTRON\_OVS\_VLAN\_RANGES=physnet1:100:200 //设置vlan ID value为100~200

# A comma separated list of bridge mappings for the Neutron

# openvswitch plugin (eg. physnet1:br-eth1,physnet2:br-eth2,physnet3

# :br-eth3)

CONFIG\_NEUTRON\_OVS\_BRIDGE\_MAPPINGS=physnet1:br-eth1 //设置将br-int映射到桥br-eth1(会自动创建phy-br-eth1和int-br-eth1来连接br-int和br-eth1)

# A comma separated list of colon-separated OVS bridge:interface

# pairs. The interface will be added to the associated bridge.

CONFIG\_NEUTRON\_OVS\_BRIDGE\_IFACES=br-eth1:eth1 //设置eth0桥接到br-eth1上，即最后的网络流量从eth1流出 (会自动执行ovs-vsctl add br-eth1 eth1)

此配置描述的网桥与网桥之间，网桥与网卡之间的映射和连接关系具体可结合 《图1 vlan模式下计算节点的网络设备拓扑结构图》和 《图2 vlan模式下网络节点的网络设备拓扑结构图 》来理解。

思考：很多同学可能会碰到一场景：物理机只有一块网卡，或有两块网卡但只有一块网卡连接有网线

此时，可以做如下配置

**(2)单网卡：**

CONFIG\_NEUTRON\_OVS\_BRIDGE\_MAPPINGS=physnet1:br-eth0 //设置将br-int映射到桥br-eth10

# A comma separated list of colon-separated OVS bridge:interface

# pairs. The interface will be added to the associated bridge.

CONFIG\_NEUTRON\_OVS\_BRIDGE\_IFACES= //配置为空

这个配置的含义是将br-int映射到br-eth0，但是br-eth0并没有与真正的物理网卡绑定，这就需要你事先在所有的计算节点(或网络节点)上事先创建好br-eth0桥，并将eth0添加到br-eth0上，然后在br-eth0上配置好ip，那么RDO在安装的时候，只要建立好br-int与br-eth0之间的连接，整个网络就通了。

此时如果网络节点也是单网卡的话，可能就不能使用float ip的功能了。

**(3)双网卡，单网线**

CONFIG\_NEUTRON\_OVS\_BRIDGE\_MAPPINGS=physnet1:br-eth1 //设置将br-int映射到桥br-eth1

# A comma separated list of colon-separated OVS bridge:interface

# pairs. The interface will be added to the associated bridge.

CONFIG\_NEUTRON\_OVS\_BRIDGE\_IFACES=eth1 //配置为空

还是默认都配置到eth1上，然后通过iptables将eth1的流量forward到eth0(没有试验过，不确定是否可行)

**3. vlan网络模式详解**

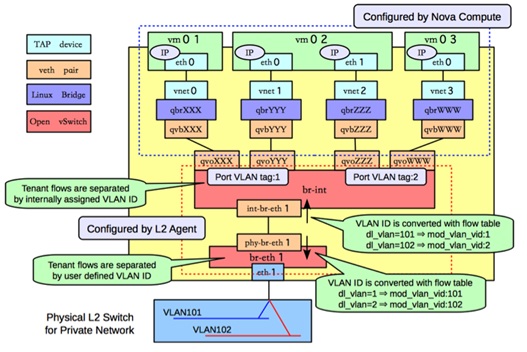


图1 vlan模式下计算节点的网络设备拓扑结构图

首先来分析下vlan网络模式下，计算节点上虚拟网络设备的拓扑结构。

**(1)qbrXXX 等设备**

前面已经讲过，主要是因为不能再tap设备vnet0上配置network ACL rules而增加的

**(2)qvbXXX/qvoXXX等设备**

这是一对veth pair devices，用来连接bridge device和switch，从名字猜测下：q-quantum, v-veth, b-bridge, o-open vswitch(quantum年代的遗留)。

**(3) int-br-eth1和phy-br-eth1**

这也是一对veth pair devices，用来连接br-int和br-eth1, 另外，vlan ID的转化也是在这执行的，比如从int-br-eth1进来的packets，其vlan id=101会被转化成1，同理，从phy-br-eth1出去的packets，其vlan id会从1转化成101

**(4)br-eth1和eth1**

packets要想进入physical network最后还得到真正的物理网卡eth1，所以add eth1 to br-eth1上，整个链路才完全打通

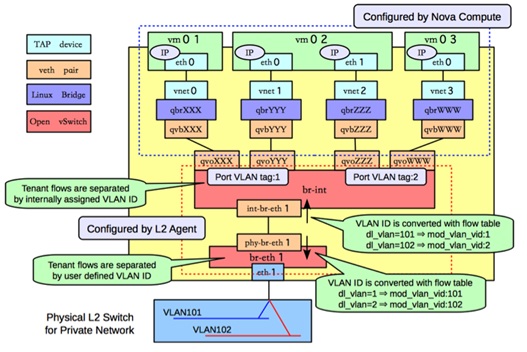


图2 vlan模式下网络节点的网络设备拓扑结构图

网络节点与计算节点相比，就是多了external network，L3 agent和dhcp agent。

**(1)network namespace**

每个L3 router对应一个private network，但是怎么保证每个private的ip address可以overlapping而又不相互影响呢，这就利用了linux kernel的network namespace

**(2)qr-YYY和qg-VVV等设备 (q-quantum, r-router, g-gateway)**

qr-YYY获得了一个internal的ip，qg-VVV是一个external的ip，通过iptables rules进行NAT映射。

**思考：phy-br-ex和int-br-ex是干啥的?**

坚持"所有packets必须经过物理的线路才能通"的思想，虽然 qr-YYY和qg-VVV之间建立的NAT的映射，归根到底还得通过一条物理链路，那么phy-br-ex和int-br-ex就建立了这条物理链路。