neutron 基础

基本概念

1. network
   是Neutron的一个二层网络的资源模型，支持的类型有：vlan、vxlan、local、flat、gre等。
   - vlan网络是配置在物理交换机上的网络。vlan 是一个二层的广播域，每个vlan都有一个vlan(12位)号，同一 vlan 中的 instance 可以通信，不同 vlan 只能通过 router 通信。
   - vxlan是基于隧道技术的 overlay 网络。vxlan 网络通过唯一的 segmentation ID（也叫VNI，24位）与其他 vxlan 网络区分。vxlan 中数据包会通过 VNI 封装成 UDP 包进行传输。
2. subnet
   一个subnet归属一一个network, 子网包含cidr网段、ip版本、子网路由等基本信息
3. port
   可以看作虚拟交换机上的一个端口，port上定义了mac地址和IP地址
4. router
   router用于不同网络虚机的通信，以及虚机连接外网
5. tap设备
   设备是一种工作在二层协议的点对点网络设备,与每个虚机port对应，工作在二层，收发的是mac层数据帧
6. veth pair 设备
   是一种成对出现的点对点网络设备，从一段输入的数据会从另一端改变方向输出
   创建命令：`ip link add veth01 type veth peer name veth02`
7. patch port设备
   ovs里的一种port类型，与veth pair类似，用于连接两个ovs桥
8. namespace
   在二层网络上，VLAN 可以将一个物理交换机分割成几个独立的虚拟交换机。类似地，在三层网络上，Linux network namespace（netns） 可以将一个物理三层网络分割成几个独立的虚拟三层网络。每个 netns 拥有独立的 （virtual）network devices, IP addresses, IP routing tables, /proc/net directory, ports 等等

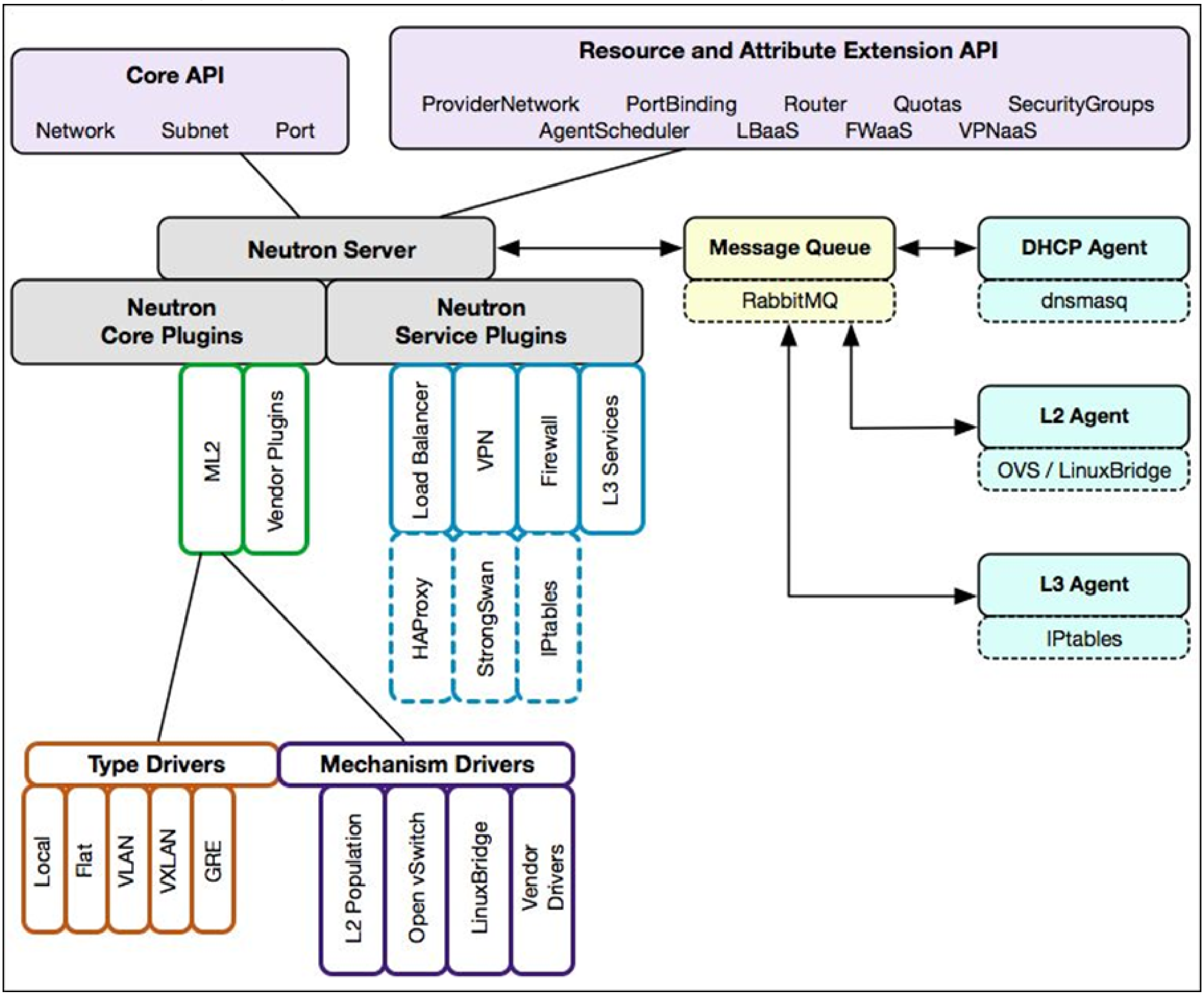
   ```
   ip netns add test_ns
   ip netns list
   ip netns delete test_ns
   ip netns exec test_ns bash
   ```

9. linux bridge
   Bridge 设备是linux基于内核实现的二层数据交换设备，其作用类似于现实世界中的二级交换机,基于MAC地址学习做转发

10. openvswitch
    ovs是一个虚拟交换机，可根据流表进行灵活的转发控制。
11. 地址解析协议arp
    链路层通信根据mac地址来确定目的端口，arp协议负责ip地址与mac地址之间的映射，
    - 同网段下主机A与主机B之间的arp解析流程
    - 不同网段下主机A与主机B之间的arp解析流程

neutron 架构

neutron 是openstack平台的一个不可缺少的组件，对外它提供了一套api接口，允许用户去创建定义一些网络资源，包括network、subnet、port、router等。neutron 对内利用linux原生（linux router 、linux bridge、iptables）以及其它虚拟网络功能(ovs)构建出真正的网。
从部署角度来说，Neutron分为三类节点：控制节点、计算节点、网络节点。网络节点和计算节点为VM构建了具体的网络，控制节点则对这些网络进行管理。
neutron通过plugin和agent等多个进程来提供网络服务：



1. neutron-server 进程

   - neutron-server运行在控制节点，其本质上是一个web server, 对外提供api供用户使用，neutron-server的启动过程中会加载ml2核心插件和其它service 插件
   - ml2插件处理network、subnet、port资源，实现了网络拓扑类型（Flat、VLAN、VXLAN、GRE）和底层虚拟网络（linux bridge、OVS）分离的机制，并分别通过Driver的形式进行扩展。其中，不同的网络拓扑类型对应着type driver，由type manager管理，不同的网络实现机制对应着Mechanism Driver（比如Linux bridge、OVS、NSX等），由Mechanism Manager管理。
   - service插件处理router、acl、pbr等其它扩展资源
   - neutron-server接收api请求后会交给对应的插件进行处理，包括数据库的读写，通知agent等。

2. l2 agent: ovs-neutron-agent进程

   - 运行在计算节点和网络节点，负责l2功能的实现，主要包括br-int、br-tun、br-floating桥上的流表的处理、安全组的处理
   - ovs-neutron-agent是控制ovs的，通过native或ovs_ofctl进行控制ovs
   - 主要处理流程是rpc_loop, 循环监听ovs db server数据的变化，进行相应的处理逻辑

3. l3 agent: neutron-l3-agent进程

   - 负责三层转发、浮动ip等的实现，l3-agent整体逻辑就是sync_router或者接收rpc消息，然后把router加入queue，再调度从queue中取出router然后处理。l3-agent也有自己的extensions，setup.cfg都有入口，和ovs-agent道理类似，只是它的manager调用extension的这些方法add_router，update_router，delete_router和ha_state_change，也有L3AgentExtensionAPI用于extension获取agent的数据。
   - 集中式模式下只运行在网络节点。dvr分布式模式下，运行在所有的网络节点和计算节点

4. neutron-dhcp-agent进程

   - 通常与网络节点复用，运行在网络节点，提供网络的dhcp功能。
   - neutron的dhcp功能借助于dnsmasq软件和命名空间来实现，每个网络都对应一个dhcp namespace, namespace里会起一个dnsmasq进程独立的为一个租户网络提供dhcp服务

组件通信

进程间rpc远程通信模式

Neutron中控制端neutron-plugin和设备端相应的neutron-agent间rpc通信是单向异步的， plugin和agent上都要开启Publisher和Consumer。由于同一类的agent往往不止一个，因此Neutron rpc采用"发布——订阅"模式在plugin和agent间传递消息，在Publisher和Consumer实例化时需要指定全局唯一的Topic。Neutron中Publisher类的命名规范为**AgentNotifierApi**，它们的实例可以向特定的Consumer发送消息，Consumer接收到消息后，通过dispatcher解封装，在调用**RpcCallBacks**在本地执行消息体。以acl代码为例：

```
# agent端向plugin
class AclCallbacks(object):
    target = oslo_messaging.Target(version='1.0')

    def __init__(self, plugin):
        super(AclCallbacks, self).__init__()
        self.plugin = plugin

    # set acl status on plugin
    def set_acl_status(self, context, acl_binding_id, status, **kwargs):
        """Agent uses this to set a acl's status."""
        pass


class AclAgentApi(object):
    """Plugin side of plugin to agent RPC API."""

    def __init__(self, topic, host):
        self.host = host
        target = oslo_messaging.Target(topic=topic, version='1.0')
        self.client = n_rpc.get_client(target)

    def _prepare_rpc_client(self, host=None):
        if host:
            return self.client.prepare(server=host)
        else:
            # historical behaviour (RPC broadcast)
            return self.client.prepare(fanout=True)

    def create_acl(self, context, acl, host=None):
        cctxt = self._prepare_rpc_client(host)
        # TODO(blallau) host param is not used on agent side (to be removed)
        cctxt.cast(context, 'create_acl', acl=acl, host=self.host)


class AclPlugin(acl_db.Acl_db_mixin):
    """ACL service plugin class"""
    supported_extension_aliases = [acl_ext.ALIAS]
    path_prefix = acl_ext.API_PREFIX

    def __init__(self):
        """Do the initialization for the acl service plugin here."""

        self.agent_rpc = AclAgentApi(constants.ACL_AGENT, cfg.CONF.host)
        acl_db.subscribe()

        rpc_worker = service.RpcWorker([self], worker_process_count=0)
        self.add_worker(rpc_worker)

    # This is called by RpcWorker initializer.
    def start_rpc_listeners(self):
        self.endpoints = [AclCallbacks(self)]
        self.conn = n_rpc.Connection()
        self.conn.create_consumer(constants.ACL_PLUGIN, self.endpoints, fanout=False)
        return self.conn.consume_in_threads()
```

进程内通信Messaging Callback System发布订阅模式

```
#事件发布
def _confirm_router_interface_not_in_use(self, context, router_id,
                                         subnet_id):
    try:
        registry.publish(
            resources.ROUTER_INTERFACE,
            events.BEFORE_DELETE, self,
            payload=events.DBEventPayload(
                context, metadata={'subnet_id': subnet_id},
                resource_id=router_id))
    except exceptions.CallbackFailure as e:
        # NOTE(armax): preserve old check's behavior
        if len(e.errors) == 1:
            raise e.errors[0].error
        raise l3_exc.RouterInUse(router_id=router_id, reason=e)
#事件订阅
def pbr_callback(resource, event, trigger, payload=None):
    # port up, then call tunnel_create
    LOG.debug('Pbr callback is called for resource router_interfac
e before_delete, payload: %s',
                  payload)
        pbr_plugin = directory.get_plugin('PBR')
        pbr_plugin.check_router_interface_not_in_use(payload)


    def subscribe():
        registry.subscribe(
            pbr_callback, resources.ROUTER_INTERFACE, events.BEFORE_DELETE)
```

流表

br-int流表

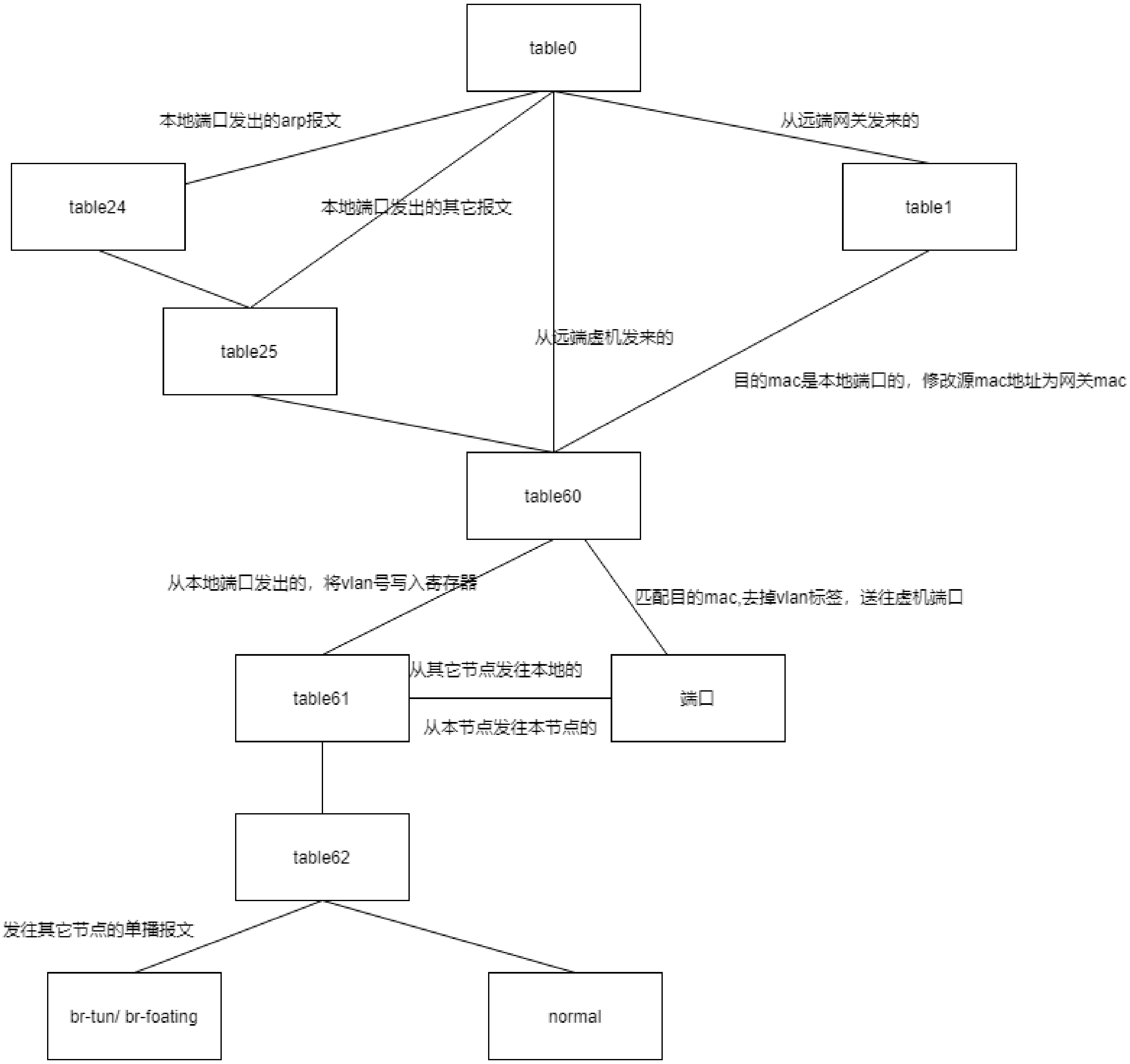
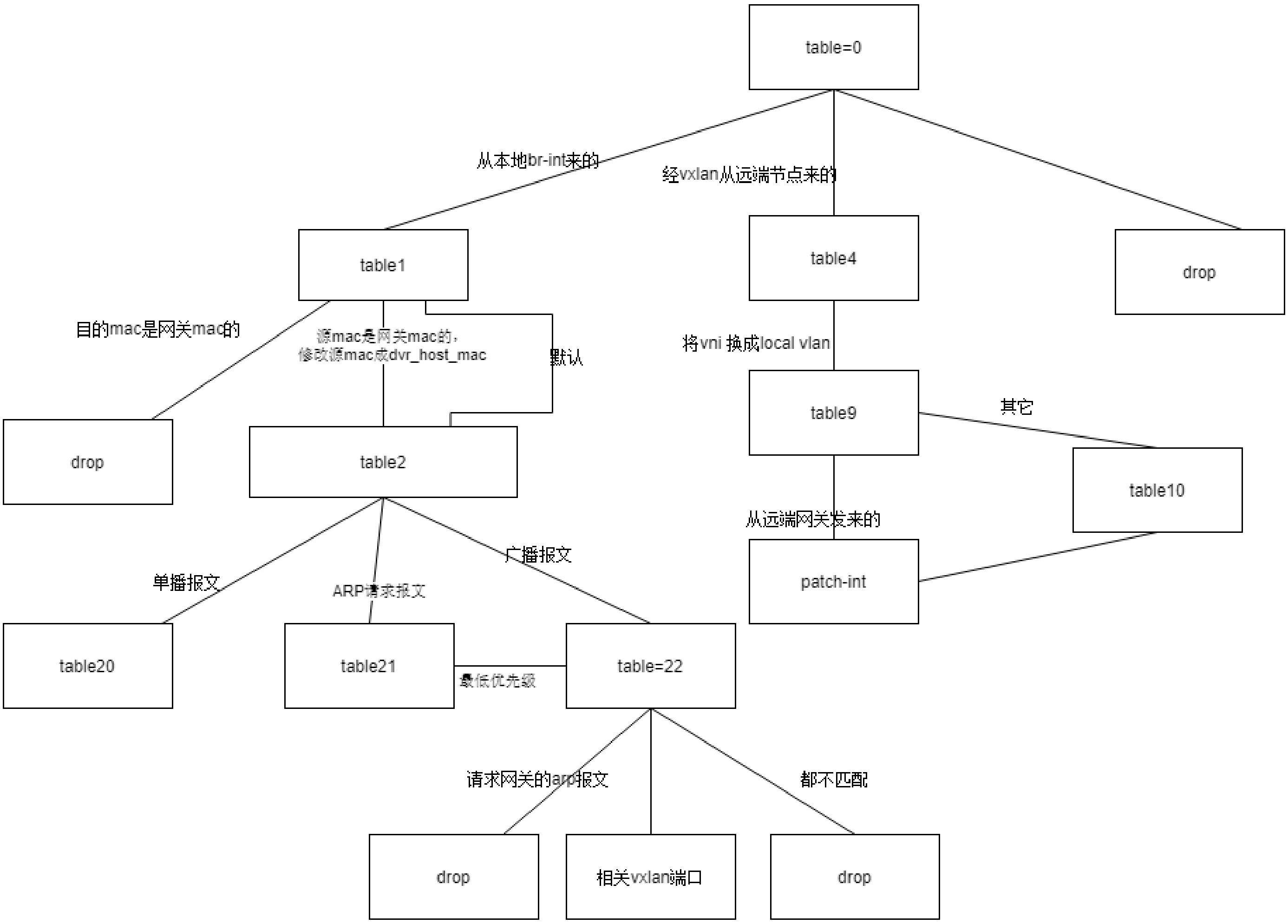


table0
本地端口发出的arp报文
从远端网关发来的
table24
本地端口发出的其它报文
table1
table25
从远端虚机发来的
目的mac是本地端口的，修改源mac地址为网关mac
table60
从本地端口发出的，将vlan号写入寄存器
匹配目的mac,去掉vlan标签，送往虚机端口
table61
从其它节点发往本地的
端口
从本节点发往本节点的
table62
发往其它节点的单播报文
br-tun/ br-foating
normal

br-tun流表

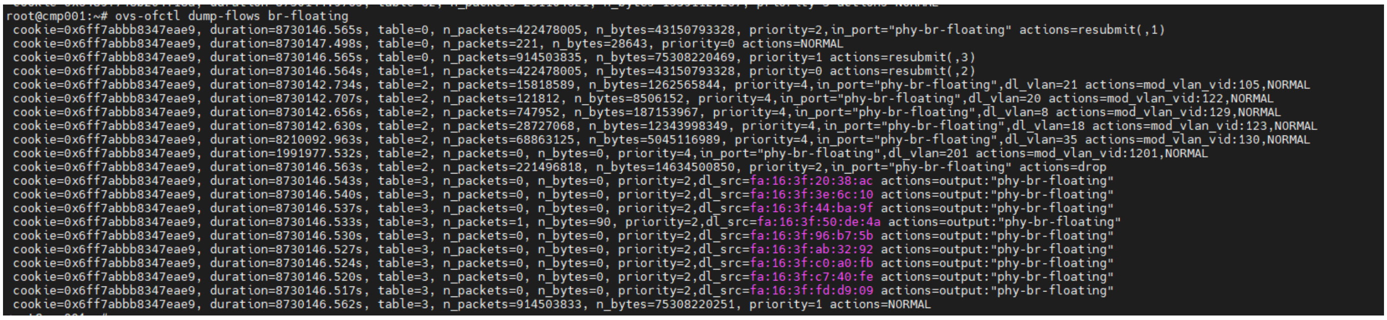


br-floating /br-smgt流表

```
    Bridge br-floating
        Controller "tcp:127.0.0.1:6633"
            is_connected: true
        fail_mode: secure
        Port phy-br-floating
            Interface phy-br-floating
                type: patch
                options: {peer=int-br-floating}
        Port patch-to-damesh
            Interface patch-to-damesh
                type: patch
                options: {peer=patch-to-floating}
        Port br-floating
            Interface br-floating
                type: internal

    Bridge br-damesh
        Port br-damesh
```

```
            Interface br-damesh
                type: internal
        Port patch-to-floating
            Interface patch-to-floating
                type: patch
                options: {peer=patch-to-damesh}
        Port data
            tag: 1016
            Interface data
                type: internal
        Port "bond1"
            Interface "bond1"

    Bridge br-smgt
        Controller "tcp:127.0.0.1:6633"
            is_connected: true
        fail_mode: secure
        Port phy-br-smgt
            Interface phy-br-smgt
                type: patch
                options: {peer=int-br-smgt}
        Port br-smgt
            Interface br-smgt
                type: internal
        Port patch-to-mgmesh
            Interface patch-to-mgmesh
                type: patch
                options: {peer=patch-to-smgt}

    Bridge br-mgmesh
        Port br-mgmesh
            Interface br-mgmesh
                type: internal
        Port "enp175s0f0"
            Interface "enp175s0f0"
        Port control
            tag: 1010
            Interface control
                type: internal
        Port patch-to-smgt
            Interface patch-to-smgt
                type: patch
```

```
                    options: {peer=patch-to-mgmesh}
```



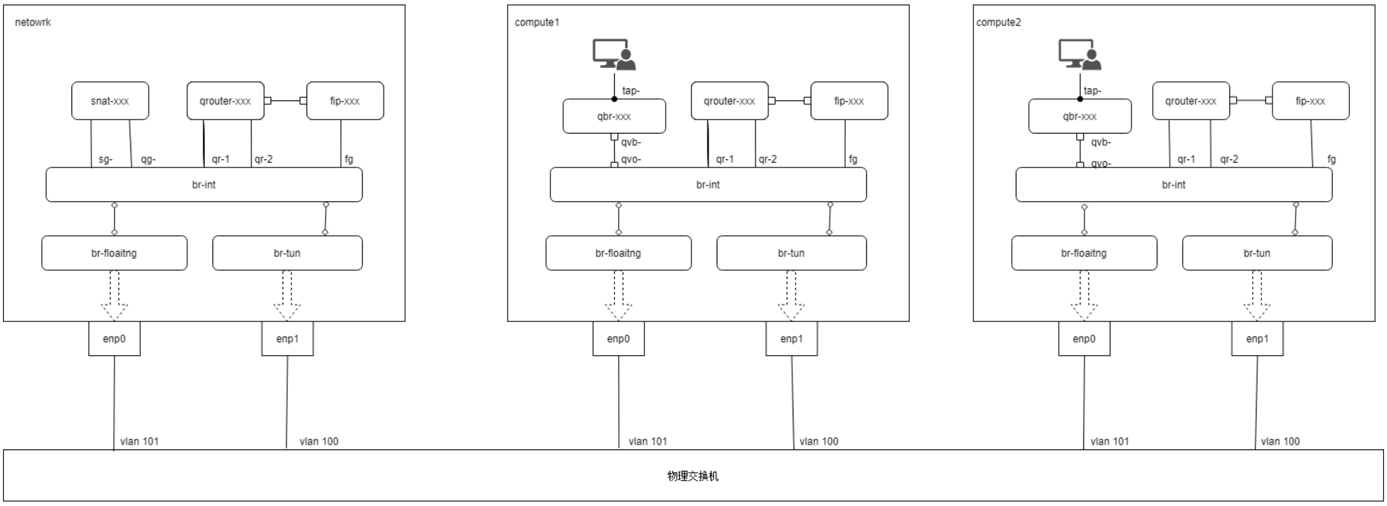
```
root@cmp001:~# ovs-ofctl dump-flows br-floating
 cookie=0x6ff7abbb8347eae9, duration=8730146.565s, table=0, n_packets=422478005, n_bytes=43150793328, priority=2,in_port="phy-br-floating" actions=resubmit(,1)
 cookie=0x6ff7abbb8347eae9, duration=8730147.498s, table=0, n_packets=221, n_bytes=28643, priority=0 actions=NORMAL
 cookie=0x6ff7abbb8347eae9, duration=8730146.565s, table=0, n_packets=914503835, n_bytes=75308220469, priority=1 actions=resubmit(,3)
 cookie=0x6ff7abbb8347eae9, duration=8730146.564s, table=1, n_packets=422478005, n_bytes=43150793328, priority=0 actions=resubmit(,2)
 cookie=0x6ff7abbb8347eae9, duration=8730142.734s, table=2, n_packets=15818589, n_bytes=1262565844, priority=4,in_port="phy-br-floating",dl_vlan=21 actions=mod_vlan_vid:105,NORMAL
 cookie=0x6ff7abbb8347eae9, duration=8730142.707s, table=2, n_packets=121812, n_bytes=8506152, priority=4,in_port="phy-br-floating",dl_vlan=20 actions=mod_vlan_vid:122,NORMAL
 cookie=0x6ff7abbb8347eae9, duration=8730142.656s, table=2, n_packets=747952, n_bytes=187153967, priority=4,in_port="phy-br-floating",dl_vlan=8 actions=mod_vlan_vid:129,NORMAL
 cookie=0x6ff7abbb8347eae9, duration=8730142.630s, table=2, n_packets=28727068, n_bytes=12343998349, priority=4,in_port="phy-br-floating",dl_vlan=18 actions=mod_vlan_vid:123,NORMAL
 cookie=0x6ff7abbb8347eae9, duration=8210092.963s, table=2, n_packets=68863125, n_bytes=5045116989, priority=4,in_port="phy-br-floating",dl_vlan=35 actions=mod_vlan_vid:130,NORMAL
 cookie=0x6ff7abbb8347eae9, duration=1991977.532s, table=2, n_packets=0, n_bytes=0, priority=4,in_port="phy-br-floating",dl_vlan=201 actions=mod_vlan_vid:1201,NORMAL
 cookie=0x6ff7abbb8347eae9, duration=8730146.563s, table=2, n_packets=221496818, n_bytes=14634500850, priority=2,in_port="phy-br-floating" actions=drop
 cookie=0x6ff7abbb8347eae9, duration=8730146.543s, table=3, n_packets=0, n_bytes=0, priority=2,dl_src=fa:16:3f:20:38:ac actions=output:"phy-br-floating"
 cookie=0x6ff7abbb8347eae9, duration=8730146.540s, table=3, n_packets=0, n_bytes=0, priority=2,dl_src=fa:16:3f:3e:6c:10 actions=output:"phy-br-floating"
 cookie=0x6ff7abbb8347eae9, duration=8730146.537s, table=3, n_packets=0, n_bytes=0, priority=2,dl_src=fa:16:3f:44:ba:9f actions=output:"phy-br-floating"
 cookie=0x6ff7abbb8347eae9, duration=8730146.533s, table=3, n_packets=1, n_bytes=90, priority=2,dl_src=fa:16:3f:50:de:4a actions=output:"phy-br-floating"
 cookie=0x6ff7abbb8347eae9, duration=8730146.530s, table=3, n_packets=0, n_bytes=0, priority=2,dl_src=fa:16:3f:96:b7:5b actions=output:"phy-br-floating"
 cookie=0x6ff7abbb8347eae9, duration=8730146.527s, table=3, n_packets=0, n_bytes=0, priority=2,dl_src=fa:16:3f:ab:32:92 actions=output:"phy-br-floating"
 cookie=0x6ff7abbb8347eae9, duration=8730146.524s, table=3, n_packets=0, n_bytes=0, priority=2,dl_src=fa:16:3f:c0:a0:fb actions=output:"phy-br-floating"
 cookie=0x6ff7abbb8347eae9, duration=8730146.520s, table=3, n_packets=0, n_bytes=0, priority=2,dl_src=fa:16:3f:c7:40:fe actions=output:"phy-br-floating"
 cookie=0x6ff7abbb8347eae9, duration=8730146.517s, table=3, n_packets=0, n_bytes=0, priority=2,dl_src=fa:16:3f:fd:d9:09 actions=output:"phy-br-floating"
 cookie=0x6ff7abbb8347eae9, duration=8730146.562s, table=3, n_packets=914503833, n_bytes=75308220251, priority=1 actions=NORMAL
```

l2population 驱动

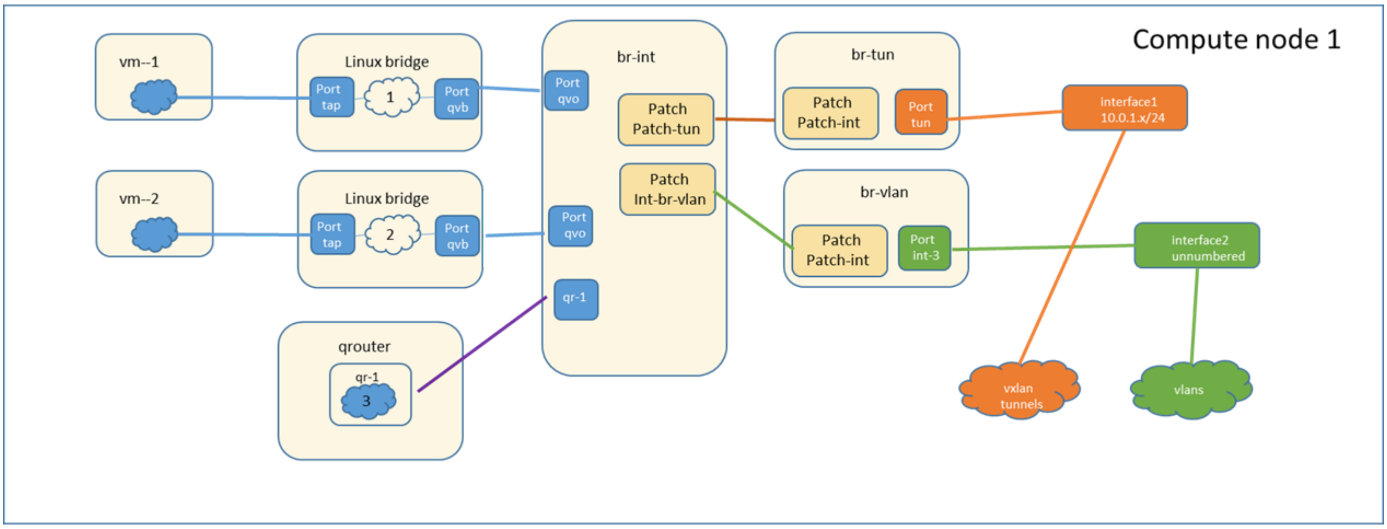
L2 Population 是用来提高 VXLAN 网络 Scalability 的。它的作用是在VTEP上提供了proxy arp功能，让VTEP知道虚机ip对应的mac地址和所属的host

```
ovs-ofctl dump-flows br-tun --names |grep fa:16:3e:07:17:a8
 cookie=0x75818fb661a1bde1, duration=614527.111s, table=20, n_packets=160, n_bytes=15176, priority=2,dl_vlan=3,dl_dst=fa:16:3e:07:17:a8 actions=strip_vlan,load:0x18->NXM_NX_TUN_ID[],output:"vxlan-1e000108"
 cookie=0x75818fb661a1bde1, duration=614527.113s, table=21, n_packets=3, n_bytes=126, priority=1,arp,dl_vlan=3,arp_tpa=192.168.1.99 actions=load:0x2->NXM_OF_ARP_OP[],move:NXM_NX_ARP_SHA[]->NXM_NX_ARP_THA[],move:NXM_OF_ARP_SPA[]->NXM_OF_ARP_TPA[],load:0xfa163e0717a8->NXM_NX_ARP_SHA[],load:0xc0a80163->NXM_OF_ARP_SPA[],move:NXM_OF_ETH_SRC[]->NXM_OF_ETH_DST[],mod_dl_src:fa:16:3e:07:17:a8,IN_PORT
 cookie=0x75818fb661a1bde1, duration=5216.856s, table=22, n_packets=193, n_bytes=12158, priority=1,dl_vlan=3 actions=strip_vlan,load:0x18->NXM_NX_TUN_ID[],output:"vxlan-1e000108",output:"vxlan-1e000106"

```

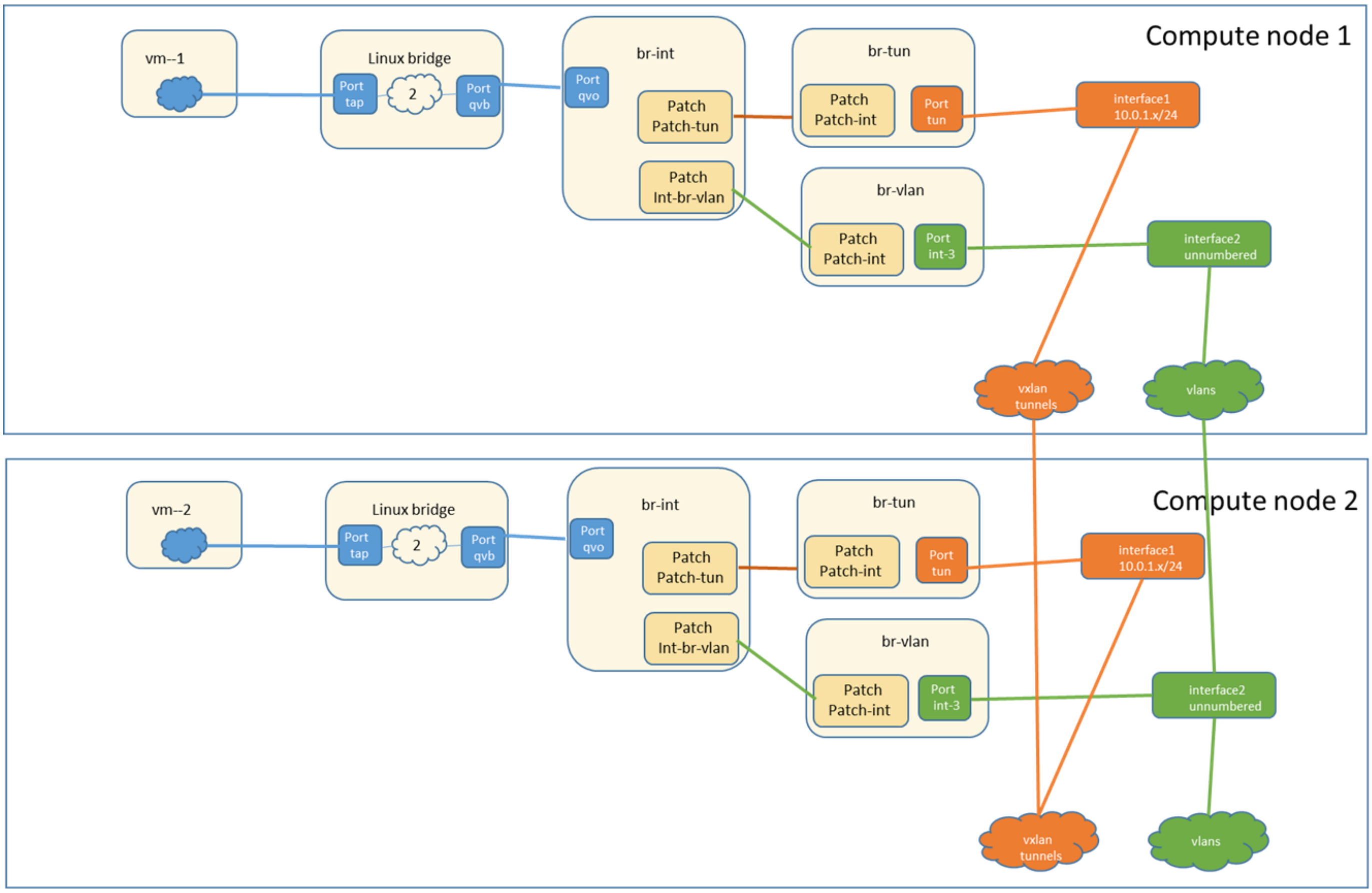
虚机流量走向



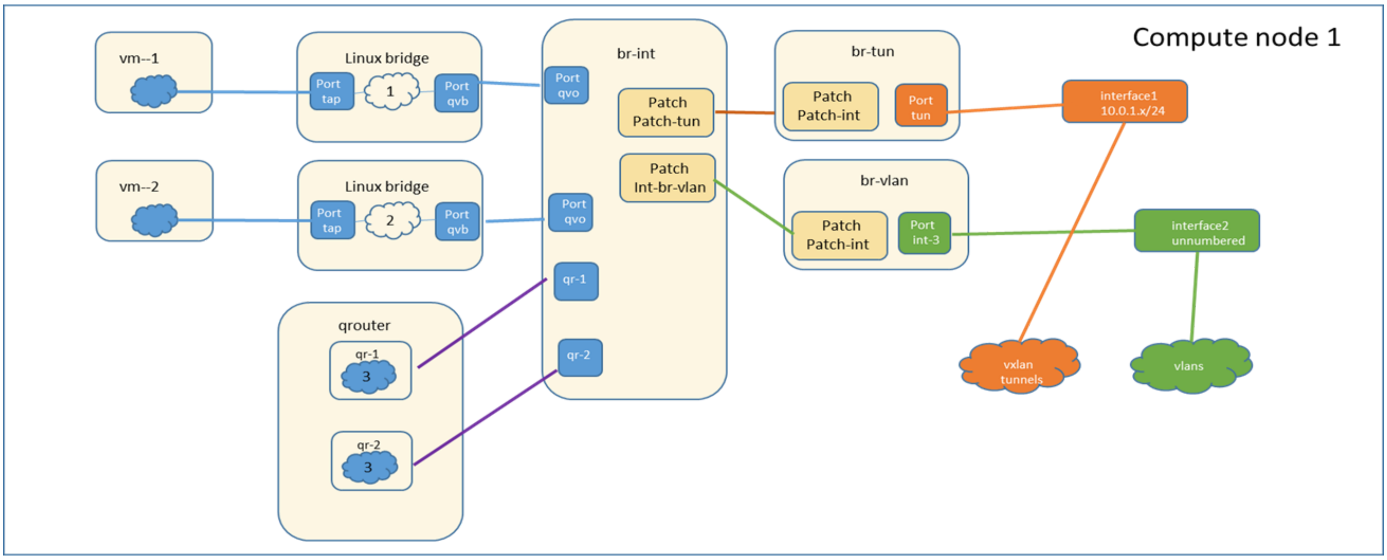
同网络同宿主机



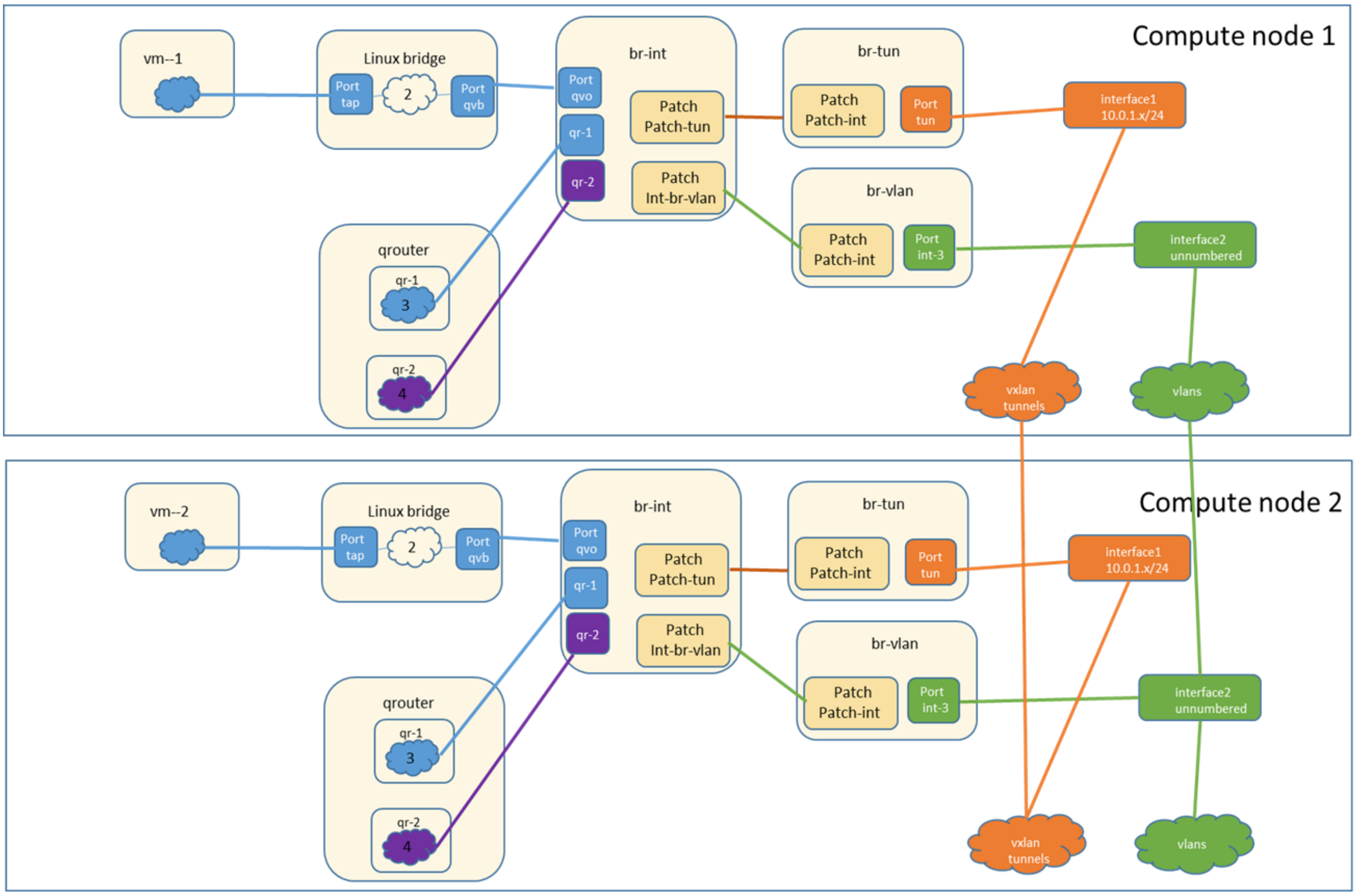
同网络不同宿主机



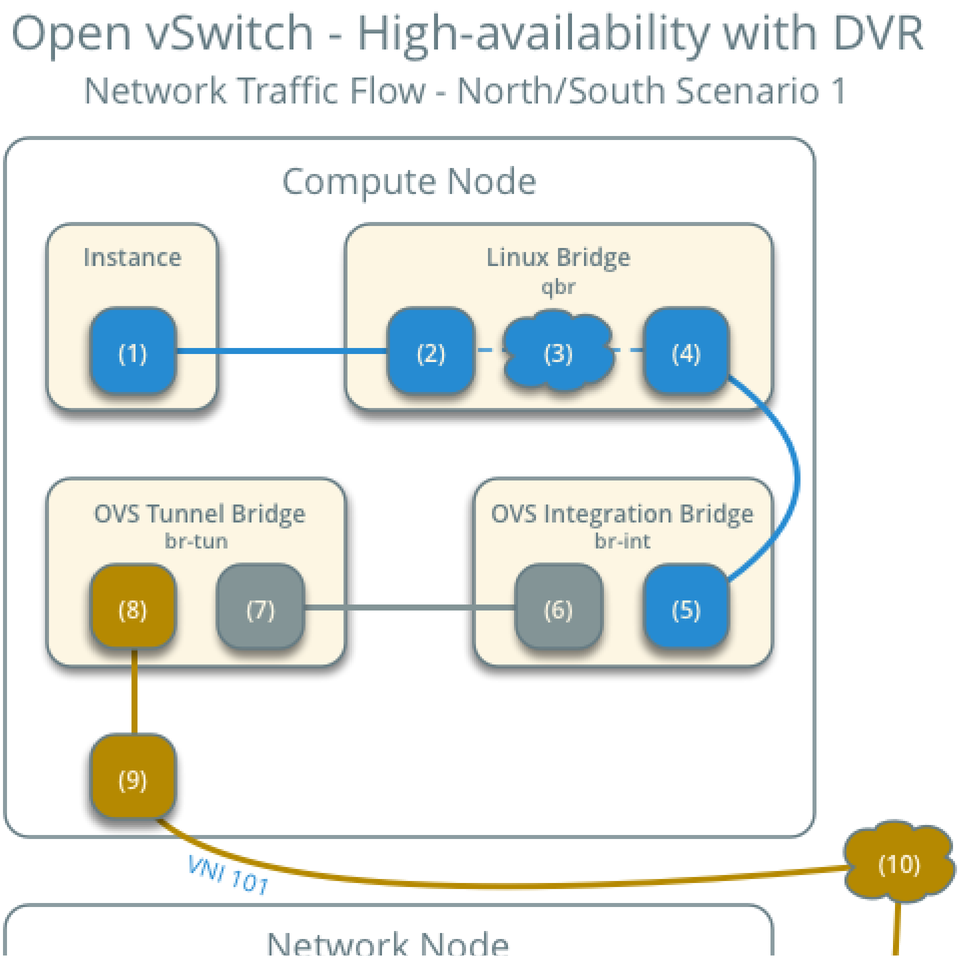
不同网络同宿主机



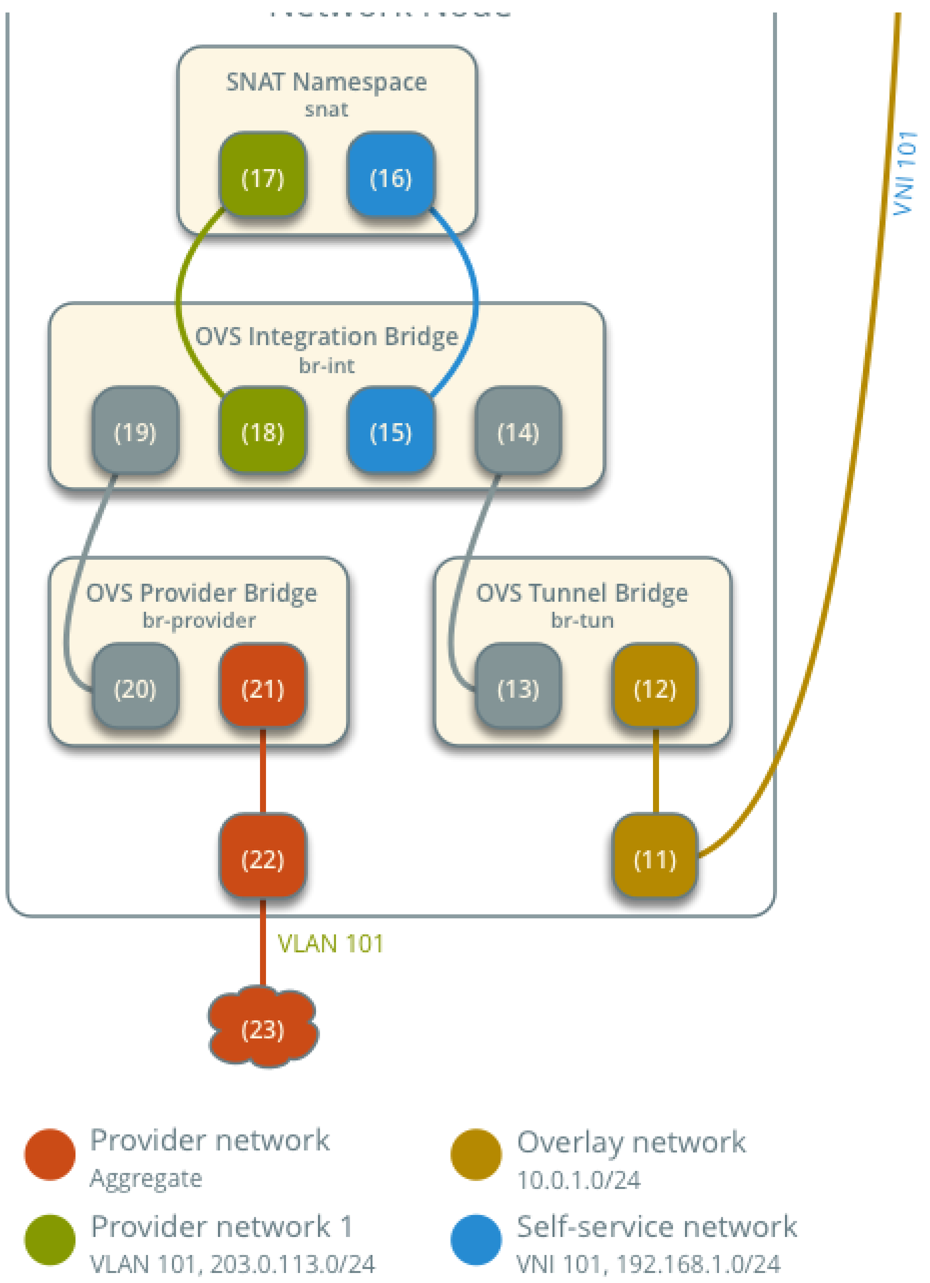
不同网络不同宿主机



没有浮动ip访问外网

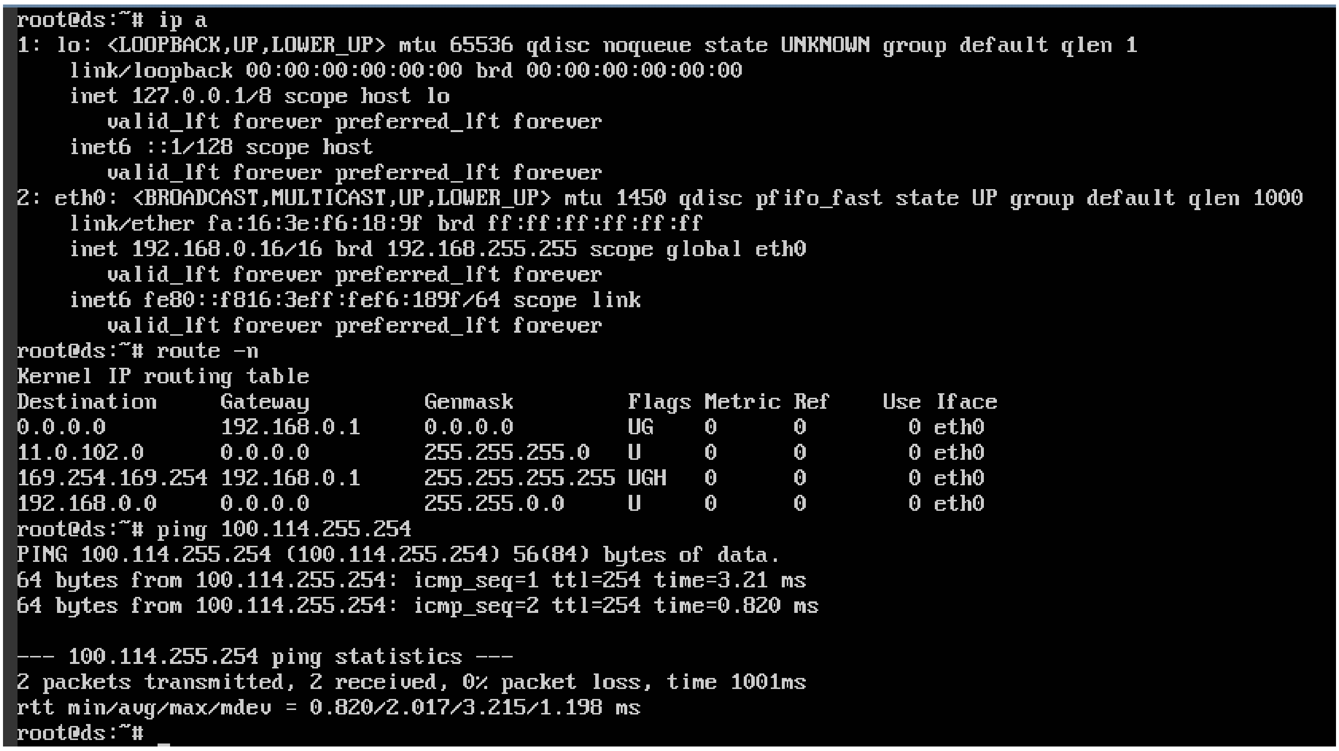

Open vSwitch - High-availability with DVR

Network Traffic Flow - North/South Scenario 1



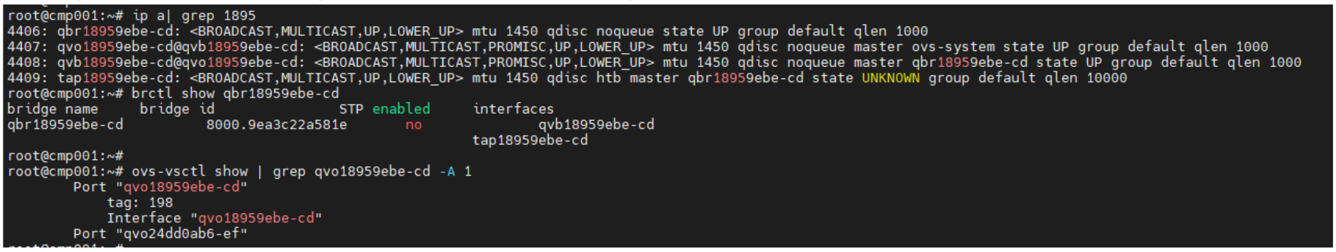
虚机192.168.0.16 ping浮动ip网关100.114.255.254流程：

1. 虚机首先查找自己的路由表，访问目标100.144.255.254，匹配到默认路由，将数据包通过eth0网卡发往网关192.168.0.1

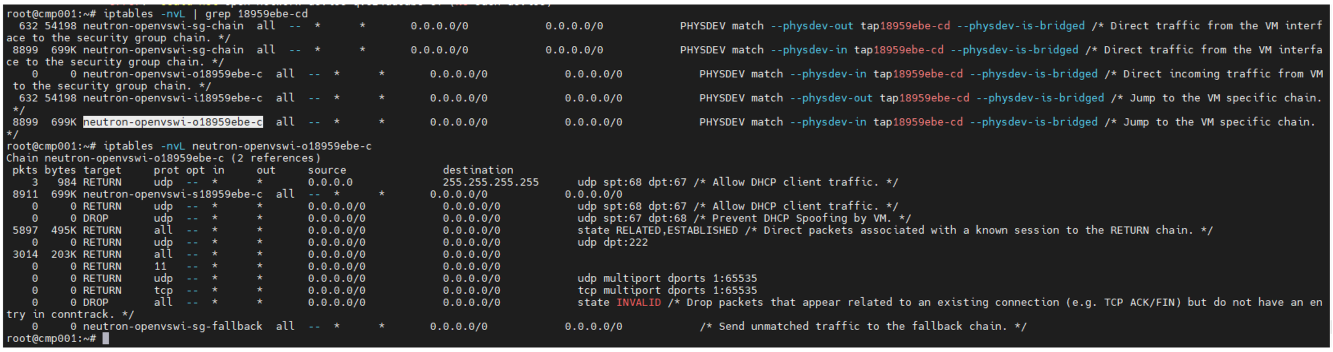


```
root@ds:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc pfifo_fast state UP group default qlen 1000
    link/ether fa:16:3e:f6:18:9f brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.16/16 brd 192.168.255.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fef6:189f/64 scope link
       valid_lft forever preferred_lft forever
root@ds:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
11.0.102.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.169.254 192.168.0.1     255.255.255.255 UGH   0      0        0 eth0
192.168.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth0
root@ds:~# ping 100.114.255.254
PING 100.114.255.254 (100.114.255.254) 56(84) bytes of data.
64 bytes from 100.114.255.254: icmp_seq=1 ttl=254 time=3.21 ms
64 bytes from 100.114.255.254: icmp_seq=2 ttl=254 time=0.820 ms

--- 100.114.255.254 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.820/2.017/3.215/1.198 ms
root@ds:~# _
```

2. 在虚机的宿主机上，流量从虚机port对应的tap口发出，tap口连在qbr桥上，经过安全组的过滤后从qbr桥上的另一个qvb口出去，从另一端qvo口进入br-int桥

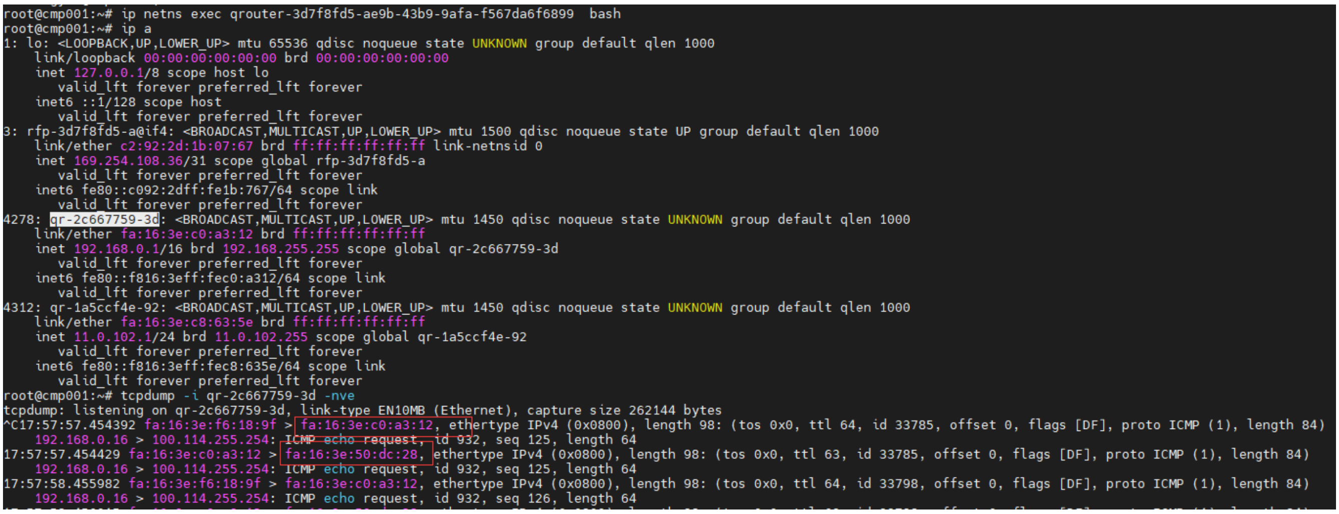


```
root@cmp001:~# ip a| grep 1895
4406: qbr18959ebe-cd: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UP group default qlen 1000
4407: qvo18959ebe-cd@qvb18959ebe-cd: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1450 qdisc noqueue master ovs-system state UP group default qlen 1000
4408: qvb18959ebe-cd@qvo18959ebe-cd: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1450 qdisc noqueue master qbr18959ebe-cd state UP group default qlen 1000
4409: tap18959ebe-cd: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc htb master qbr18959ebe-cd state UNKNOWN group default qlen 10000
root@cmp001:~# brctl show qbr18959ebe-cd
bridge name     bridge id               STP enabled     interfaces
qbr18959ebe-cd          8000.9ea3c22a581e       no              qvb18959ebe-cd
                                                        tap18959ebe-cd

root@cmp001:~#
root@cmp001:~# ovs-vsctl show | grep qvo18959ebe-cd -A 1
        Port "qvo18959ebe-cd"
            tag: 198
            Interface "qvo18959ebe-cd"
        Port "qvo24dd0ab6-ef"
```

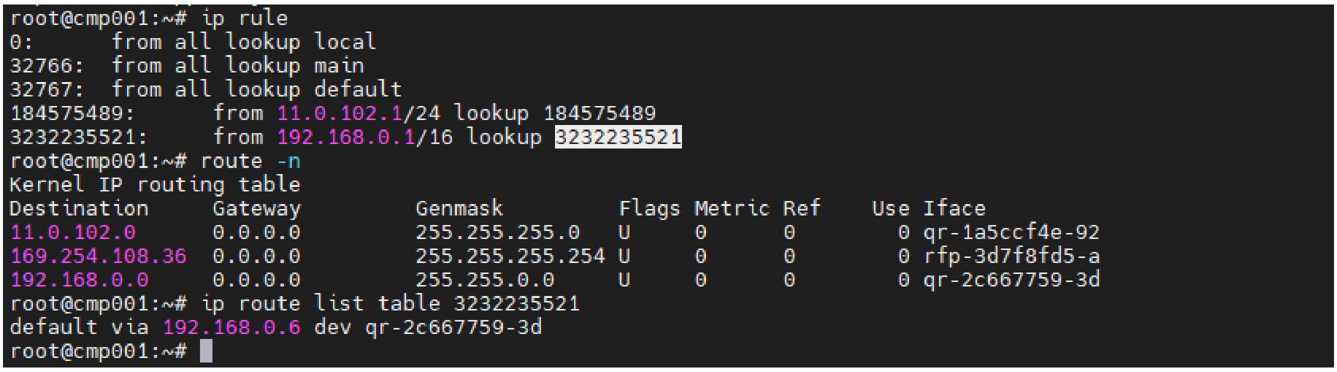


```
root@cmp001:~# iptables -nvL | grep 18959ebe-cd
  632 54198 neutron-openvswi-sg-chain  all  --  *      *       0.0.0.0/0            0.0.0.0/0            PHYSDEV match --physdev-out tap18959ebe-cd --physdev-is-bridged /* Direct traffic from the VM interf
ace to the security group chain. */
 8899  699K neutron-openvswi-sg-chain  all  --  *      *       0.0.0.0/0            0.0.0.0/0            PHYSDEV match --physdev-in tap18959ebe-cd --physdev-is-bridged /* Direct traffic from the VM interfa
ce to the security group chain. */
    0     0 neutron-openvswi-o18959ebe-c  all  --  *      *       0.0.0.0/0            0.0.0.0/0            PHYSDEV match --physdev-in tap18959ebe-cd --physdev-is-bridged /* Direct incoming traffic from VM
 to the security group chain. */
  632 54198 neutron-openvswi-i18959ebe-c  all  --  *      *       0.0.0.0/0            0.0.0.0/0            PHYSDEV match --physdev-out tap18959ebe-cd --physdev-is-bridged /* Jump to the VM specific chain.
 */
 8899  699K neutron-openvswi-o18959ebe-c  all  --  *      *       0.0.0.0/0            0.0.0.0/0            PHYSDEV match --physdev-in tap18959ebe-cd --physdev-is-bridged /* Jump to the VM specific chain.
*/
root@cmp001:~# iptables -nvL neutron-openvswi-o18959ebe-c
Chain neutron-openvswi-o18959ebe-c (2 references)
 pkts bytes target     prot opt in     out     source               destination
    3   984 RETURN     udp  --  *      *       0.0.0.0              255.255.255.255      udp spt:68 dpt:67 /* Allow DHCP client traffic. */
 8911  699K neutron-openvswi-s18959ebe-c  all  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 RETURN     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spt:68 dpt:67 /* Allow DHCP client traffic. */
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp spt:67 dpt:68 /* Prevent DHCP Spoofing by VM. */
 5897  495K RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0            state RELATED,ESTABLISHED /* Direct packets associated with a known session to the RETURN chain. */
    0     0 RETURN     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp dpt:222
 3014  203K RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 RETURN     11   --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 RETURN     udp  --  *      *       0.0.0.0/0            0.0.0.0/0            udp multiport dports 1:65535
    0     0 RETURN     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp multiport dports 1:65535
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            state INVALID /* Drop packets that appear related to an existing connection (e.g. TCP ACK/FIN) but do not have an en
try in conntrack. */
    0     0 neutron-openvswi-sg-fallback  all  --  *      *       0.0.0.0/0            0.0.0.0/0            /* Send unmatched traffic to the fallback chain. */
root@cmp001:~# 
```

3. 流量从qvo口进入br-int桥，经流表匹配，从qr口送出
4. qr口是qrouter名字空间中网关所在的设备，qr网关收报报文后，通过路由表得知下一跳通过qr口发往192.168.0.6，这个地址是在网络节点snat名字空间中sg口上的ip.

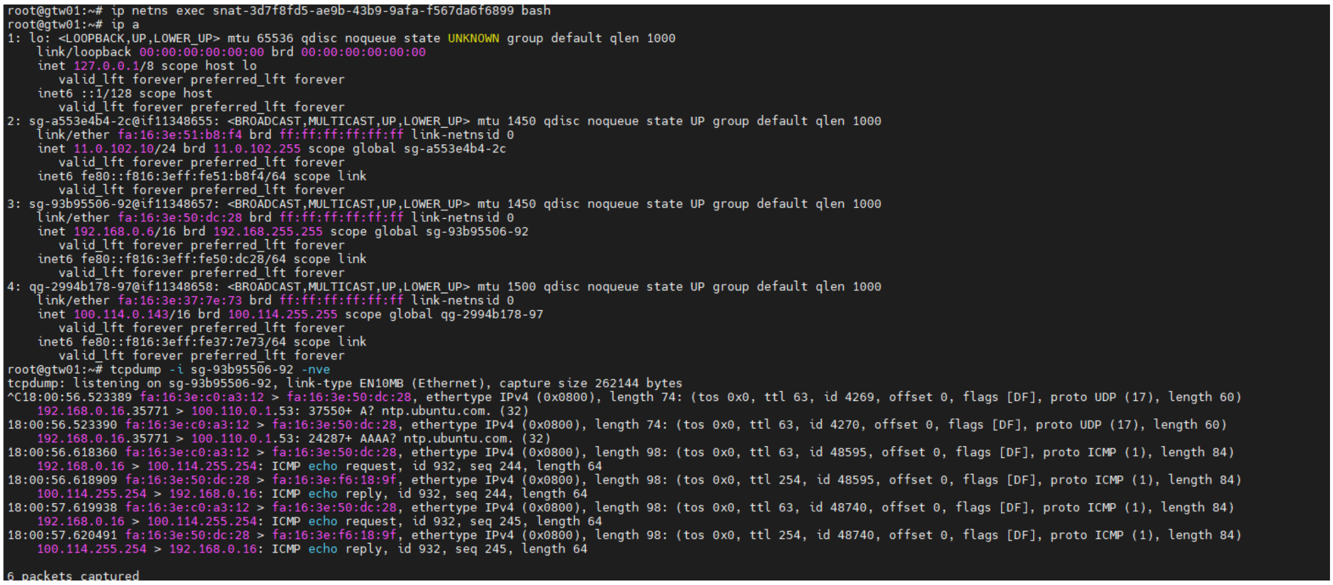


```
root@cmp001:~# ip netns exec qrouter-3d7f8fd5-ae9b-43b9-9afa-f567da6f6899  bash
root@cmp001:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: rfp-3d7f8fd5-a@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether c2:92:2d:1b:07:67 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 169.254.108.36/31 scope global rfp-3d7f8fd5-a
       valid_lft forever preferred_lft forever
    inet6 fe80::c092:2dff:fe1b:767/64 scope link
       valid_lft forever preferred_lft forever
4278: qr-2c667759-3d: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ether fa:16:3e:c0:a3:12 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/16 brd 192.168.255.255 scope global qr-2c667759-3d
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fec0:a312/64 scope link
       valid_lft forever preferred_lft forever
4312: qr-1a5ccf4e-92: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ether fa:16:3e:c8:63:5e brd ff:ff:ff:ff:ff:ff
    inet 11.0.102.1/24 brd 11.0.102.255 scope global qr-1a5ccf4e-92
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fec8:635e/64 scope link
       valid_lft forever preferred_lft forever
root@cmp001:~# tcpdump -i qr-2c667759-3d -nve
tcpdump: listening on qr-2c667759-3d, link-type EN10MB (Ethernet), capture size 262144 bytes
^C17:57:57.454392 fa:16:3e:f6:18:9f > fa:16:3e:c0:a3:12, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id 33785, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.0.16 > 100.114.255.254: ICMP echo request, id 932, seq 125, length 64
17:57:57.454429 fa:16:3e:c0:a3:12 > fa:16:3e:50:dc:28, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id 33785, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.0.16 > 100.114.255.254: ICMP echo request, id 932, seq 125, length 64
17:57:58.455982 fa:16:3e:f6:18:9f > fa:16:3e:c0:a3:12, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id 33798, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.0.16 > 100.114.255.254: ICMP echo request, id 932, seq 126, length 64
```



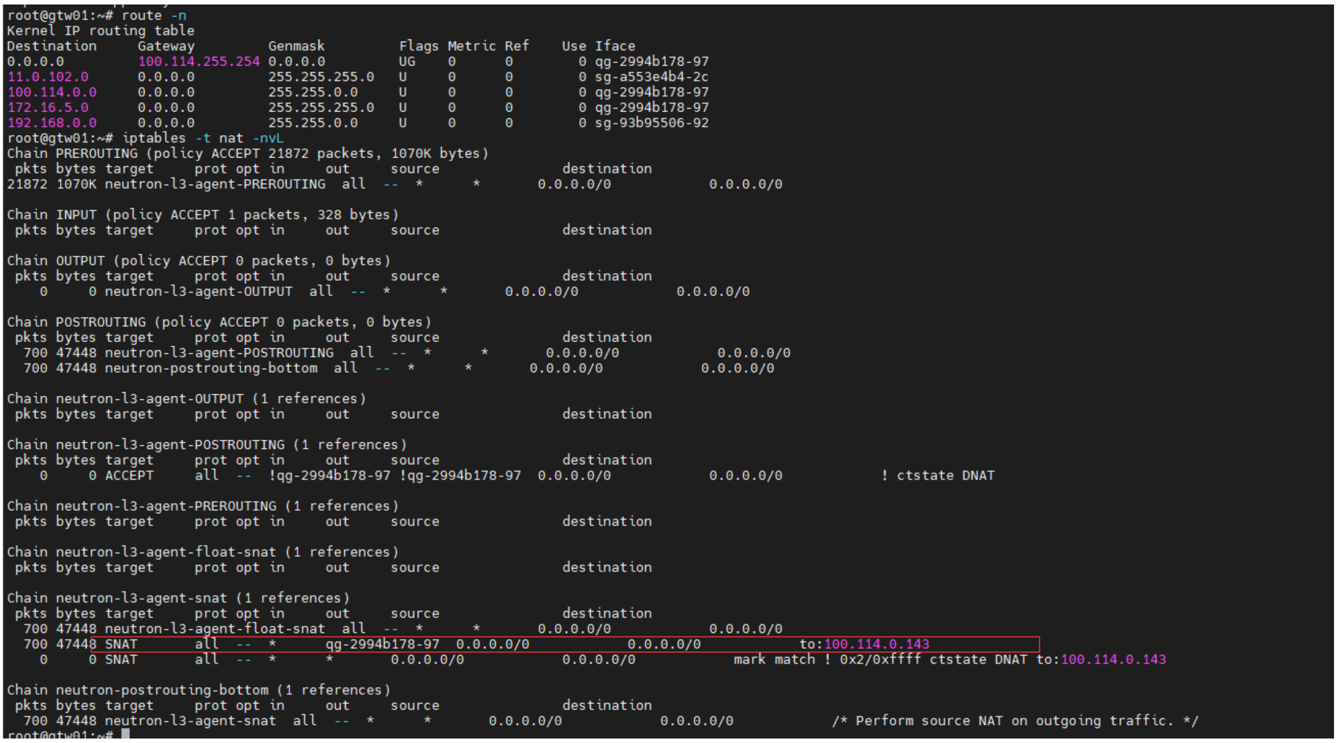
```
root@cmp001:~# ip rule
0:      from all lookup local
32766:  from all lookup main
32767:  from all lookup default
184575489:      from 11.0.102.1/24 lookup 184575489
3232235521:     from 192.168.0.1/16 lookup 3232235521
root@cmp001:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
11.0.102.0      0.0.0.0         255.255.255.0   U     0      0        0 qr-1a5ccf4e-92
169.254.108.36  0.0.0.0         255.255.255.254 U     0      0        0 rfp-3d7f8fd5-a
192.168.0.0     0.0.0.0         255.255.0.0     U     0      0        0 qr-2c667759-3d
root@cmp001:~# ip route list table 3232235521
default via 192.168.0.6 dev qr-2c667759-3d
root@cmp001:~#
```

5.流量发往192.168.0.6的过程就是同子网不同宿主机流量通信的过程，流量经vxlan隧道发往snat所在的网络节点



```
root@gtw01:~# ip netns exec snat-3d7f8fd5-ae9b-43b9-9afa-f567da6f6899 bash
root@gtw01:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: sg-a553e4b4-2c@if11348655: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UP group default qlen 1000
    link/ether fa:16:3e:51:b8:f4 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 11.0.102.10/24 brd 11.0.102.255 scope global sg-a553e4b4-2c
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe51:b8f4/64 scope link
       valid_lft forever preferred_lft forever
3: sg-93b95506-92@if11348657: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UP group default qlen 1000
    link/ether fa:16:3e:50:dc:28 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.0.6/16 brd 192.168.255.255 scope global sg-93b95506-92
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe50:dc28/64 scope link
       valid_lft forever preferred_lft forever
4: qg-2994b178-97@if11348658: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether fa:16:3e:37:7e:73 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 100.114.0.143/16 brd 100.114.255.255 scope global qg-2994b178-97
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe37:7e73/64 scope link
       valid_lft forever preferred_lft forever
root@gtw01:~# tcpdump -i sg-93b95506-92 -nve
tcpdump: listening on sg-93b95506-92, link-type EN10MB (Ethernet), capture size 262144 bytes
^C18:00:56.523389 fa:16:3e:c0:a3:12 > fa:16:3e:50:dc:28, ethertype IPv4 (0x0800), length 74: (tos 0x0, ttl 63, id 4269, offset 0, flags [DF], proto UDP (17), length 60)
    192.168.0.16.35771 > 100.110.0.1.53: 37550+ A? ntp.ubuntu.com. (32)
18:00:56.523390 fa:16:3e:c0:a3:12 > fa:16:3e:50:dc:28, ethertype IPv4 (0x0800), length 74: (tos 0x0, ttl 63, id 4270, offset 0, flags [DF], proto UDP (17), length 60)
    192.168.0.16.35771 > 100.110.0.1.53: 24287+ AAAA? ntp.ubuntu.com. (32)
18:00:56.618360 fa:16:3e:c0:a3:12 > fa:16:3e:50:dc:28, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id 48595, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.0.16 > 100.114.255.254: ICMP echo request, id 932, seq 244, length 64
18:00:56.618909 fa:16:3e:50:dc:28 > fa:16:3e:f6:18:9f, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 254, id 48595, offset 0, flags [DF], proto ICMP (1), length 84)
    100.114.255.254 > 192.168.0.16: ICMP echo reply, id 932, seq 244, length 64
18:00:57.619938 fa:16:3e:c0:a3:12 > fa:16:3e:50:dc:28, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 63, id 48740, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.0.16 > 100.114.255.254: ICMP echo request, id 932, seq 245, length 64
18:00:57.620491 fa:16:3e:50:dc:28 > fa:16:3e:f6:18:9f, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 254, id 48740, offset 0, flags [DF], proto ICMP (1), length 84)
    100.114.255.254 > 192.168.0.16: ICMP echo reply, id 932, seq 245, length 64

6 packets captured
```

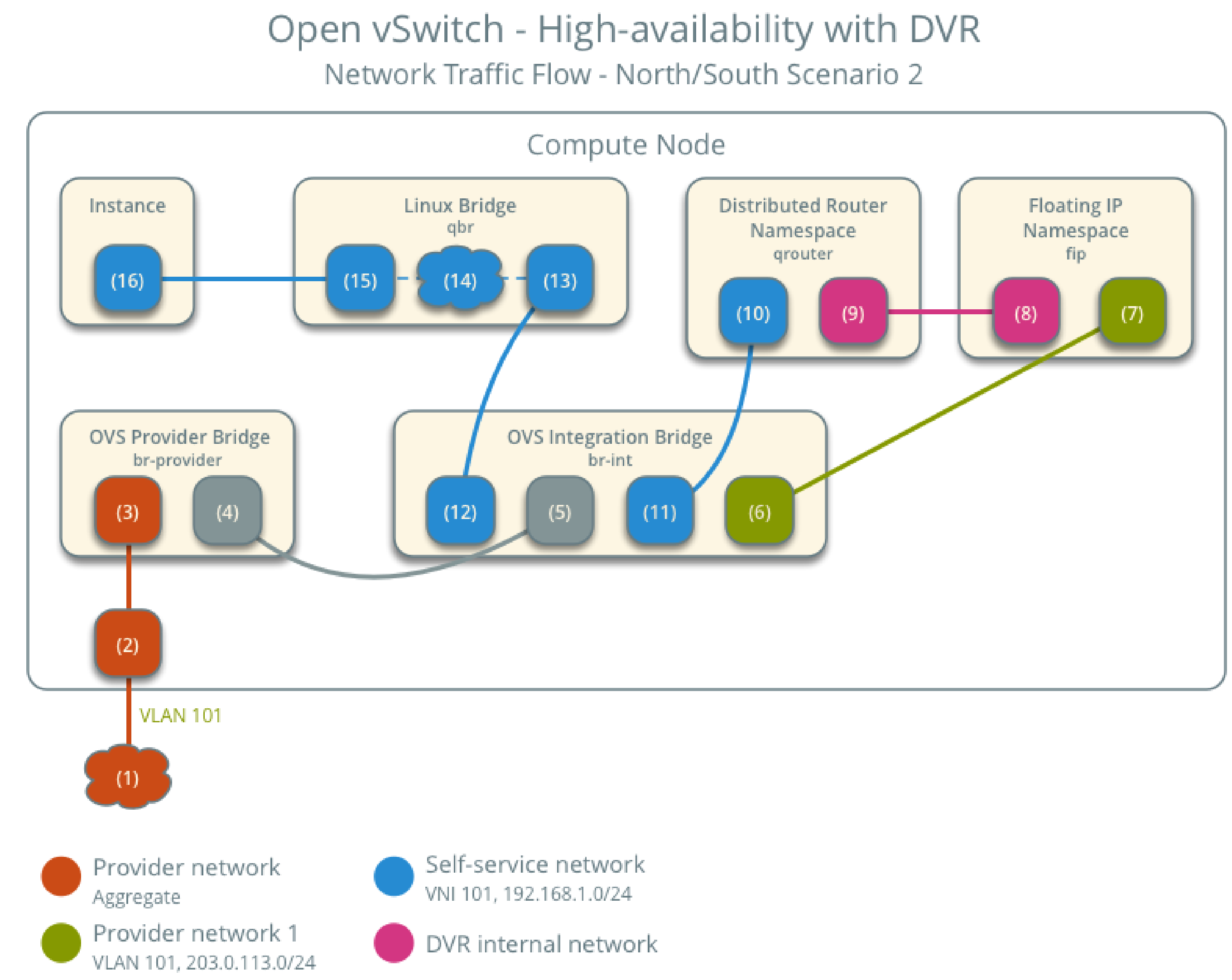
5. 流量到达sg口后，通过路由表得知吓一跳通过qg口送往浮动ip网关，在经过路由后会先通过snat将源地址换成sg口的地址。



```
root@gtw01:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         100.114.255.254 0.0.0.0         UG    0      0        0 qg-2994b178-97
11.0.102.0      0.0.0.0         255.255.255.0   U     0      0        0 sg-a553e4b4-2c
100.114.0.0     0.0.0.0         255.255.0.0     U     0      0        0 qg-2994b178-97
172.16.5.0      0.0.0.0         255.255.255.0   U     0      0        0 qg-2994b178-97
192.168.0.0     0.0.0.0         255.255.0.0     U     0      0        0 sg-93b95506-92
root@gtw01:~# iptables -t nat -nvL
Chain PREROUTING (policy ACCEPT 21872 packets, 1070K bytes)
 pkts bytes target     prot opt in     out     source               destination
21872 1070K neutron-l3-agent-PREROUTING  all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain INPUT (policy ACCEPT 1 packets, 328 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 neutron-l3-agent-OUTPUT  all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  700 47448 neutron-l3-agent-POSTROUTING  all  --  *      *       0.0.0.0/0            0.0.0.0/0
  700 47448 neutron-postrouting-bottom  all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain neutron-l3-agent-OUTPUT (1 references)
 pkts bytes target     prot opt in     out     source               destination

Chain neutron-l3-agent-POSTROUTING (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  !qg-2994b178-97 !qg-2994b178-97  0.0.0.0/0            0.0.0.0/0            ! ctstate DNAT

Chain neutron-l3-agent-PREROUTING (1 references)
 pkts bytes target     prot opt in     out     source               destination

Chain neutron-l3-agent-float-snat (1 references)
 pkts bytes target     prot opt in     out     source               destination

Chain neutron-l3-agent-snat (1 references)
 pkts bytes target     prot opt in     out     source               destination
  700 47448 neutron-l3-agent-float-snat  all  --  *      *       0.0.0.0/0            0.0.0.0/0
  700 47448 SNAT       all  --  *      qg-2994b178-97  0.0.0.0/0            0.0.0.0/0            to:100.114.0.143
    0     0 SNAT       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match ! 0x2/0xffff ctstate DNAT to:100.114.0.143

Chain neutron-postrouting-bottom (1 references)
 pkts bytes target     prot opt in     out     source               destination
  700 47448 neutron-l3-agent-snat  all  --  *      *       0.0.0.0/0            0.0.0.0/0            /* Perform source NAT on outgoing traffic. */
root@gtw01:~#
```

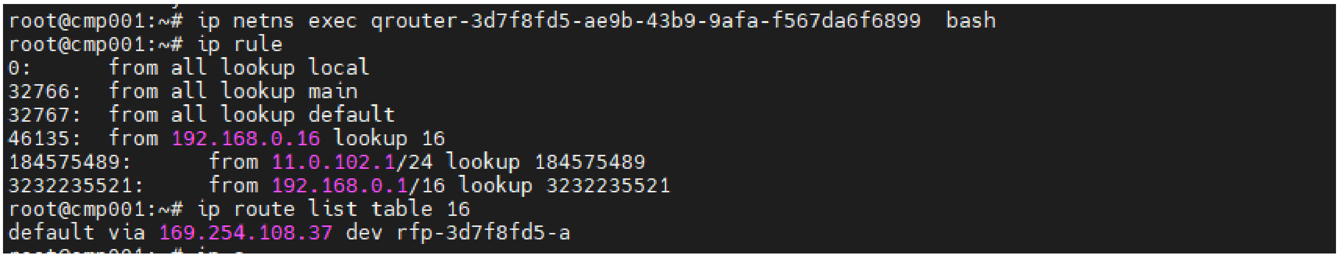
6. 流量从qg口送出，进出br-int桥，通过流表送往br-floating桥，最后在br-floating桥上转换vlan，从物理口送往交换机。
7. 回包从qg口回来，匹配路由后从sg口送出，送往虚机，该流程是同网络不同宿主机通信流程。

绑定浮动ip访问外网



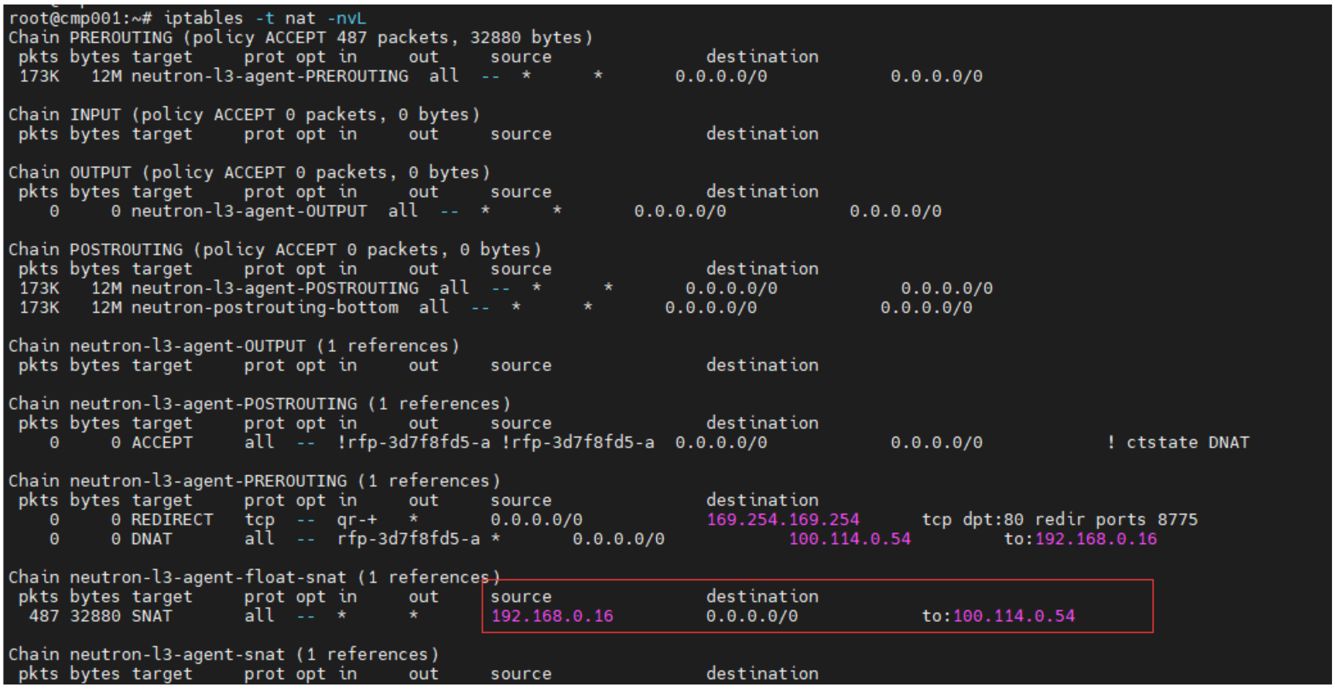
虚机192.168.0.16绑定浮动ip100.114.0.54后ping 浮动ip网关100.114.255.254流程：

1. 前面的流程一样，流量进去qrouter中，匹配到策略路由查询16号表,16号表中有条默认路由，将访问外网的流量通过rfp设备送到fip名字空间中的169.254.108.37



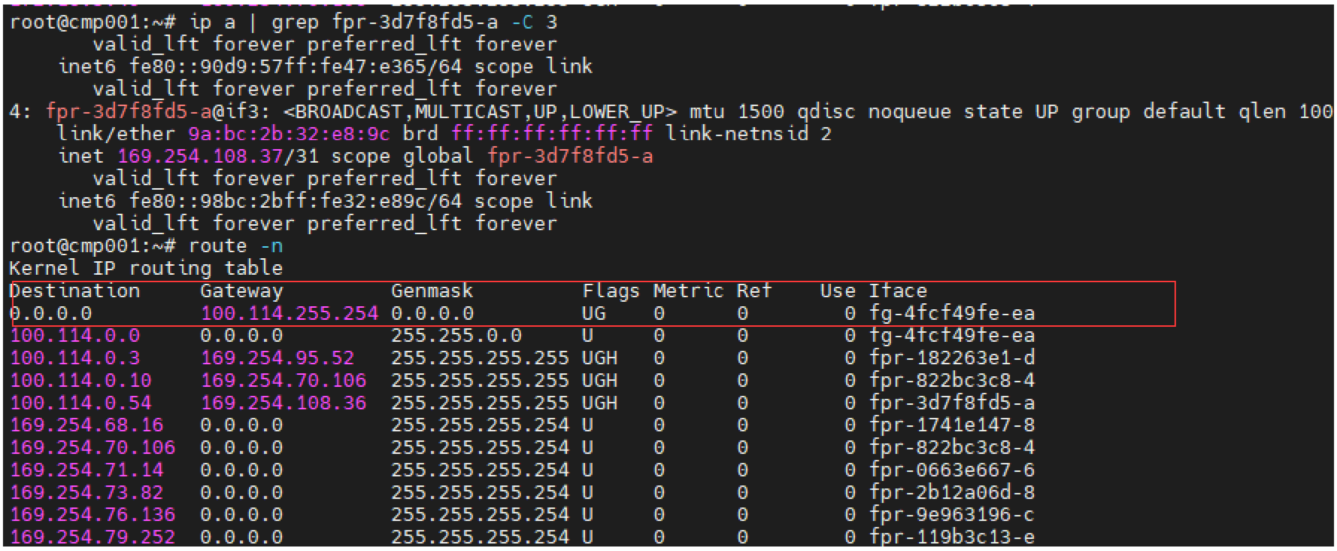
```
root@cmp001:~# ip netns exec qrouter-3d7f8fd5-ae9b-43b9-9afa-f567da6f6899  bash
root@cmp001:~# ip rule
0:      from all lookup local
32766:  from all lookup main
32767:  from all lookup default
46135:  from 192.168.0.16 lookup 16
184575489:      from 11.0.102.1/24 lookup 184575489
3232235521:     from 192.168.0.1/16 lookup 3232235521
root@cmp001:~# ip route list table 16
default via 169.254.108.37 dev rfp-3d7f8fd5-a
```

2. 路由后，先做snat，再从rfp口送出



```
root@cmp001:~# iptables -t nat -nvL
Chain PREROUTING (policy ACCEPT 487 packets, 32880 bytes)
 pkts bytes target     prot opt in     out     source               destination
 173K   12M neutron-l3-agent-PREROUTING  all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 neutron-l3-agent-OUTPUT  all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
 173K   12M neutron-l3-agent-POSTROUTING  all  --  *      *       0.0.0.0/0            0.0.0.0/0
 173K   12M neutron-postrouting-bottom  all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain neutron-l3-agent-OUTPUT (1 references)
 pkts bytes target     prot opt in     out     source               destination

Chain neutron-l3-agent-POSTROUTING (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  !rfp-3d7f8fd5-a !rfp-3d7f8fd5-a  0.0.0.0/0            0.0.0.0/0            ! ctstate DNAT

Chain neutron-l3-agent-PREROUTING (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 REDIRECT   tcp  --  qr-+   *       0.0.0.0/0            169.254.169.254      tcp dpt:80 redir ports 8775
    0     0 DNAT       all  --  rfp-3d7f8fd5-a *       0.0.0.0/0            100.114.0.54         to:192.168.0.16

Chain neutron-l3-agent-float-snat (1 references)
 pkts bytes target     prot opt in     out     source               destination
  487 32880 SNAT       all  --  *      *       192.168.0.16         0.0.0.0/0            to:100.114.0.54

Chain neutron-l3-agent-snat (1 references)
 pkts bytes target     prot opt in     out     source               destination
```

3. rfp口是一个veth pair设备，另一端在fip名字空间中，流量进入fip名字空间，匹配默认路由从fg口送出



```
root@cmp001:~# ip a | grep fpr-3d7f8fd5-a -C 3
       valid_lft forever preferred_lft forever
    inet6 fe80::90d9:57ff:fe47:e365/64 scope link
       valid_lft forever preferred_lft forever
4: fpr-3d7f8fd5-a@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 100
    link/ether 9a:bc:2b:32:e8:9c brd ff:ff:ff:ff:ff:ff link-netnsid 2
    inet 169.254.108.37/31 scope global fpr-3d7f8fd5-a
       valid_lft forever preferred_lft forever
    inet6 fe80::98bc:2bff:fe32:e89c/64 scope link
       valid_lft forever preferred_lft forever
root@cmp001:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         100.114.255.254 0.0.0.0         UG    0      0        0 fg-4fcf49fe-ea
100.114.0.0     0.0.0.0         255.255.0.0     U     0      0        0 fg-4fcf49fe-ea
100.114.0.3     169.254.95.52   255.255.255.255 UGH   0      0        0 fpr-182263e1-d
100.114.0.10    169.254.70.106  255.255.255.255 UGH   0      0        0 fpr-822bc3c8-4
100.114.0.54    169.254.108.36  255.255.255.255 UGH   0      0        0 fpr-3d7f8fd5-a
169.254.68.16   0.0.0.0         255.255.255.254 U     0      0        0 fpr-1741e147-8
169.254.70.106  0.0.0.0         255.255.255.254 U     0      0        0 fpr-822bc3c8-4
169.254.71.14   0.0.0.0         255.255.255.254 U     0      0        0 fpr-0663e667-6
169.254.73.82   0.0.0.0         255.255.255.254 U     0      0        0 fpr-2b12a06d-8
169.254.76.136  0.0.0.0         255.255.255.254 U     0      0        0 fpr-9e963196-c
169.254.79.252  0.0.0.0         255.255.255.254 U     0      0        0 fpr-119b3c13-e
```

4. 数据从fg口进入br-int桥，经流表处理送往br-floating桥，从br-floating桥上绑定的物理端口送往物理交换机。

外部ping浮动ip100.114.0.54

1. 首先要arp寻址100.114.0.54的mac地址，fip命名空间的fg口设置了arp proxy，当fg口收到arp请求后会代答arp，直接返回fg口的mac地址

```
net.ipv4.conf.fg-4fcf49fe-ea.proxy_arp = 1
```

2. arp代答返回那些fip请求的应答呢？fg口收到arp请求后，会查看是否存在到请求目标（100.114.0.54）的路由，如果存在则返回本接口的mac地址。
3. fg接口在收到100.114.0.54的请求后，根据路由通过fpr口送往qrouter命名空间
4. qrouter里的rfp口收到请求后会先做dnat转换，将目标地址换成192.168.0.16。然后根据路由从qr口发出送往主机。