Troubleshoot Cloud Networking Like a Pro

A. VM to VM flow (on same hosts)

1. Check vm's status on controller node

```
root@controller:~# nova list
+--------------------------------------+------+--------+------------+-------------+-----------------------------+
| ID                                   | Name | Status | Task State | Power State | Networks                    |
+--------------------------------------+------+--------+------------+-------------+-----------------------------+
| 1ec57719-7d3a-4e4a-af5b-8074bb3b9429 | vm_1 | ACTIVE | -          | Running     | net_1=1.1.1.4, 10.118.47.180 |
| 6aef9d84-0f29-4a4f-b1b9-9fd6450562a1 | vm_2 | ACTIVE | -          | Running     | net_1=1.1.1.5               |
```

2. Check vm's port-list. How to relate vm's vNIC/eth0 to tap/qbr/rvb/qvo interface on host so we can check if packets are arriving there.

```
root@controller:~# neutron port-list | grep 1.1.1.4
| 9409f1fe-477d-485c-acb0-d430277702ae |      | fa:16:3e:56:6b:11 | {"subnet_id": "07326022-9145-4c5b-b62f-efe5b8f8c6ae", "ip_address": "1.1.1.4"}
root@controller:~# neutron port-list | grep 1.1.1.5
| c911f327-21df-4e00-ac66-d3482a648cc8 |      | fa:16:3e:68:e9:e1 | {"subnet_id": "07326022-9145-4c5b-b62f-efe5b8f8c6ae", "ip_address": "1.1.1.5"}
```

3. Check vm's instance name

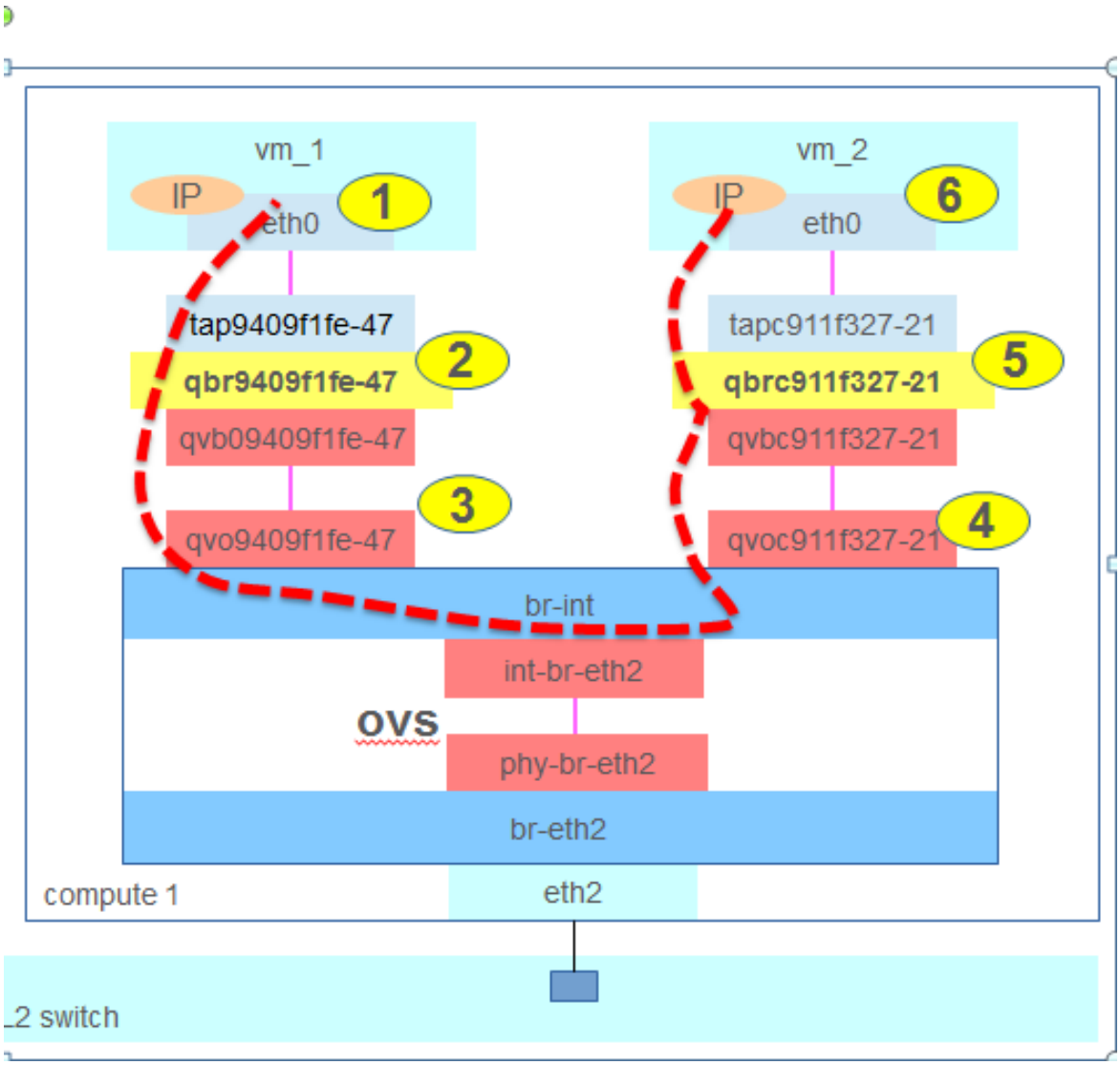
```
root@controller:~# nova list --fields name,instance_name | grep vm_1
| 1ec57719-7d3a-4e4a-af5b-8074bb3b9429 | vm_1 | instance-0000001d |
root@controller:~# nova list --fields name,instance_name | grep vm_2
| 6aef9d84-0f29-4a4f-b1b9-9fd6450562a1 | vm_2 | instance-0000001e |
root@controller:~#
```

4. Check these instances are running on compute host

```
root@compute:~# virsh list
 Id    Name                           State
----------------------------------------------------
 9     instance-0000001f              running
 10    instance-0000001d              running
 11    instance-00000020              running
 12    instance-0000001e              running
```

5. Check tap/qbr/qvb/qvo details from the vm's

```
root@compute:~# virsh domiflist instance-0000001d
Interface  Type       Source     Model       MAC
-------------------------------------------------------
tap9409f1fe-47 bridge     qbr9409f1fe-47 virtio      fa:16:3e:56:6b:11

root@compute:~# brctl show qbr9409f1fe-47
bridge name     bridge id               STP enabled     interfaces
qbr9409f1fe-47          8000.7250d9bc1424       no              qvb9409f1fe-47
                                                        tap9409f1fe-47
```




```
root@compute:~# virsh domiflist instance-0000001e
Interface  Type       Source     Model       MAC
-------------------------------------------------------
tapc911f327-21 bridge     qbrc911f327-21 virtio      fa:16:3e:68:e9:e1

root@compute:~# brctl show qbrc911f327-21
bridge name     bridge id               STP enabled     interfaces
qbrc911f327-21          8000.5e0498ef26d3       no              qvbc911f327-21
                                                        tapc911f327-21
```

6. Check ifconfig with grep of first 11 port-id on compute hosts (to verify all tap/qbr/qvb/qvo exists)

```
root@compute:~# ifconfig | grep 9409f1fe-47
qbr9409f1fe-47 Link encap:Ethernet  HWaddr 72:50:d9:bc:14:24
qvb9409f1fe-47 Link encap:Ethernet  HWaddr 72:50:d9:bc:14:24
qvo9409f1fe-47 Link encap:Ethernet  HWaddr c6:cd:4e:33:45:21
tap9409f1fe-47 Link encap:Ethernet  HWaddr fe:16:3e:56:6b:11
```

```
root@compute:~# ifconfig | grep c911f327-21
qbrc911f327-21 Link encap:Ethernet  HWaddr 5e:04:98:ef:26:d3
qvbc911f327-21 Link encap:Ethernet  HWaddr 5e:04:98:ef:26:d3
qvoc911f327-21 Link encap:Ethernet  HWaddr 5a:3b:64:cc:8c:fc
tapc911f327-21 Link encap:Ethernet  HWaddr fe:16:3e:68:e9:e1
root@compute:~#
```

7. Connect to vm's (through horizon console) and check the ip addresses / mac addresses

```
$ hostname
vm_1
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:56:6B:11
          inet addr:1.1.1.4  Bcast:1.1.1.15  Mask:255.255.255.240
          inet6 addr: fe80::f816:3eff:fe56:6b11/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:180921 errors:0 dropped:0 overruns:0 frame:0
          TX packets:179997 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17007759 (16.2 MiB)  TX bytes:17523136 (16.7 MiB)
```

```
$ hostname
vm_2
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:68:E9:E1
          inet addr:1.1.1.5  Bcast:1.1.1.15  Mask:255.255.255.240
          inet6 addr: fe80::f816:3eff:fe68:e9e1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:164273 errors:0 dropped:0 overruns:0 frame:0
          TX packets:164022 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15590214 (14.8 MiB)  TX bytes:15569216 (14.8 MiB)
```

8. vm_1 (1.1.1.4) ping vm_2 (1.1.1.5) – both are on the same host. Our diagram will show us, vm_1's vNIC (eth0) will send packet out. So from vm_1

```
$ hostname
vm_1
$ ping 1.1.1.5
PING 1.1.1.5 (1.1.1.5): 56 data bytes
64 bytes from 1.1.1.5: seq=0 ttl=64 time=0.870 ms
64 bytes from 1.1.1.5: seq=1 ttl=64 time=0.509 ms
64 bytes from 1.1.1.5: seq=2 ttl=64 time=0.623 ms
```

9. So now if we start a ping from vm_1 (1.1.1.4) to vm_2 (1.1.1.5), we should be able to see the packets on both tap device and in the corresponding bridge via tcpdump:

```
root@compute:~# tcpdump -i tap9409f1fe-47 -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tap9409f1fe-47, link-type EN10MB (Ethernet), capture size 262144 bytes
17:57:37.076960 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 6, length 64
17:57:37.077346 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 6, length 64
17:57:38.077070 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 7, length 64
17:57:38.077343 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 7, length 64
```

```
root@compute:~# tcpdump -i qbr9409f1fe-47 -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qbr9409f1fe-47, link-type EN10MB (Ethernet), capture size 262144 bytes
17:58:03.081885 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 32, length 64
17:58:03.082148 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 32, length 64
17:58:04.082091 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 33, length 64
17:58:04.082344 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 33, length 64
```

```
root@compute:~# tcpdump -i qvb9409f1fe-47 -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qvb9409f1fe-47, link-type EN10MB (Ethernet), capture size 262144 bytes
17:58:18.084819 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 47, length 64
17:58:18.085067 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 47, length 64
17:58:19.085056 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 48, length 64
17:58:19.085285 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 48, length 64
```

```
root@compute:~# tcpdump -i qvo9409f1fe-47 -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qvo9409f1fe-47, link-type EN10MB (Ethernet), capture size 262144 bytes
17:58:26.086303 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 55, length 64
17:58:26.086560 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 55, length 64
17:58:27.086548 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 56, length 64
17:58:27.086806 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 56, length 64
```

10. Now check the packets on ovs

```
root@compute:~# ovs-dpctl show
system@ovs-system:
        lookups: hit:541533 missed:54460 lost:0
        flows: 0
        masks: hit:1100254 total:1 hit/pkt:1.85
        port 0: ovs-system (internal)
        port 1: br-int (internal)
        port 2: br-ex (internal)
        port 3: eth2
        port 4: br-eth2 (internal)
        port 5: qvo9409f1fe-47
        port 6: qvoc911f327-21
        port 7: qvoe1c63f38-0c
        port 8: qvocb2a5d64-40
        port 9: qvo4e6e174a-ec
root@compute:~# ovs-dpctl dump-flows
recirc_id(0),in_port(5),eth(src=fa:16:3e:56:6b:11,dst=fa:16:3e:68:e9:e1),eth_type(0x0800),ipv4(frag=no), packets:0, bytes
:0, used:never, actions:6
recirc_id(0),in_port(6),eth(src=fa:16:3e:68:e9:e1,dst=fa:16:3e:56:6b:11),eth_type(0x0800),ipv4(frag=no), packets:0, bytes
:0, used:never, actions:5
root@compute:~#
```

11. So packets are pushed from one port to another. Let's check on vm_2 tap/qbr/qvb/qvo ports

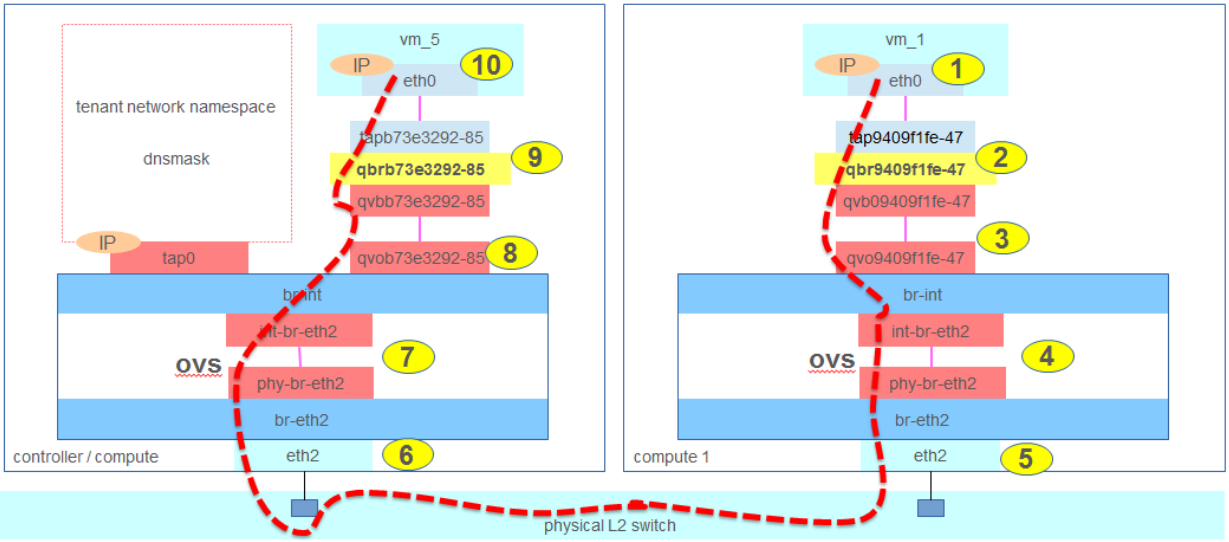
```
root@compute:~# tcpdump -i qvoc911f327-21 -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qvoc911f327-21, link-type EN10MB (Ethernet), capture size 262144 bytes
18:04:35.161588 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 424, length 64
18:04:35.161800 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 424, length 64
18:04:36.161726 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 425, length 64
18:04:36.162041 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 425, length 64
```

```
root@compute:~# tcpdump -i qvbc911f327-21 -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qvbc911f327-21, link-type EN10MB (Ethernet), capture size 262144 bytes
18:04:44.163721 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 433, length 64
18:04:44.163918 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 433, length 64
```

```
root@compute:~# tcpdump -i qbrc911f327-21 -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qbrc911f327-21, link-type EN10MB (Ethernet), capture size 262144 bytes
18:04:51.165491 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 440, length 64
18:04:51.165770 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 440, length 64
```

```
root@compute:~# tcpdump -i tapc911f327-21 -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tapc911f327-21, link-type EN10MB (Ethernet), capture size 262144 bytes
18:04:58.166978 IP 1.1.1.4 > 1.1.1.5: ICMP echo request, id 65025, seq 447, length 64
18:04:58.167193 IP 1.1.1.5 > 1.1.1.4: ICMP echo reply, id 65025, seq 447, length 64
```

B. VM to VM flow (on different hosts)



1. Check vm's status on controller node

```
root@controller:~# nova list
+--------------------------------------+------+--------+------------+-------------+------------------------------+
| ID                                   | Name | Status | Task State | Power State | Networks                     |
+--------------------------------------+------+--------+------------+-------------+------------------------------+
| 1ec57719-7d3a-4e4a-af5b-8074bb3b9429 | vm_1 | ACTIVE | -          | Running     | net_1=1.1.1.4, 10.118.47.180 |
| 6aef9d84-0f29-4a4f-b1b9-9fd6450562a1 | vm_2 | ACTIVE | -          | Running     | net_1=1.1.1.5                |
| e572f310-3c6d-4970-ba38-a296465f3986 | vm_3 | ACTIVE | -          | Running     | net_1=1.1.1.6                |
| 578c3003-4978-40c5-b35f-e17ce5c045aa | vm_4 | ACTIVE | -          | Running     | net_1=1.1.1.7                |
| f7f43d5c-c996-4549-a184-e6a194d7e7ee | vm_5 | ACTIVE | -          | Running     | net_1=1.1.1.8                |
+--------------------------------------+------+--------+------------+-------------+------------------------------+
```

2. Check vm's port-list. How to relate vm's vNIC/eth0 to tap/qbr/rvb/qvo interface on host so we can check if packets are arriving there.

```
root@controller:~# neutron port-list | grep 1.1.1.8
| b73e3292-85b2-44ae-a091-a8d4fa440da3 |      | fa:16:3e:64:37:ab | {"subnet_id": "07326022-9145-4c5b-b62f-efe5b8f8c6ae", "ip_address": "1.1.1.8"}
   |
root@controller:~# ifconfig | grep b73e3292-85
qbrb73e3292-85 Link encap:Ethernet  HWaddr c2:40:44:fe:11:2f
qvbb73e3292-85 Link encap:Ethernet  HWaddr c2:40:44:fe:11:2f
qvob73e3292-85 Link encap:Ethernet  HWaddr a6:8e:a5:b8:2c:de
tapb73e3292-85 Link encap:Ethernet  HWaddr fe:16:3e:64:37:ab
root@controller:~#
```

3. Connect to vm's (through horizon console) and check the ip addresses / mac addresses

```
# hostname
vm_5
# ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:64:37:AB
          inet addr:1.1.1.8  Bcast:1.1.1.15  Mask:255.255.255.240
          inet6 addr: fe80::f816:3eff:fe64:37ab/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2464 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2210 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:237522 (231.9 KiB)  TX bytes:210786 (205.8 KiB)
```

4. Check instance name, tap/qbr/qvb/qvo ports details and running on controller (our 2nd compute host)

```
root@controller:~# nova list --fields name,instance_name | grep vm_5
| f7f43d5c-c996-4549-a184-e6a194d7e7ee | vm_5 | instance-00000021 |
root@controller:~# virsh list
 Id    Name                           State
----------------------------------------------------
 3     instance-00000021              running

root@controller:~# virsh domiflist instance-00000021
Interface  Type       Source     Model       MAC
-------------------------------------------------------
tapb73e3292-85 bridge     qbrb73e3292-85 virtio      fa:16:3e:64:37:ab

root@controller:~# brctl show qbrb73e3292-85
bridge name     bridge id               STP enabled     interfaces
qbrb73e3292-85          8000.c24044fe112f       no              qvbb73e3292-85
                                                        tapb73e3292-85
```

5. vm_1 (1.1.1.4) to ping vm_5(1.1.1.8) – they are on different hosts

```
$ hostname
vm_1
$ ping 1.1.1.8
PING 1.1.1.8 (1.1.1.8): 56 data bytes
64 bytes from 1.1.1.8: seq=0 ttl=64 time=3.612 ms
64 bytes from 1.1.1.8: seq=1 ttl=64 time=1.059 ms
64 bytes from 1.1.1.8: seq=2 ttl=64 time=1.090 ms

--- 1.1.1.8 ping statistics ---
```

6. Packets are sent directly to eth2, I guess.

```
root@compute:~# ovs-dpctl show
system@ovs-system:
        lookups: hit:541645 missed:54487 lost:0
        flows: 1
        masks: hit:1100550 total:2 hit/pkt:1.85
        port 0: ovs-system (internal)
        port 1: br-int (internal)
        port 2: br-ex (internal)
        port 3: eth2
        port 4: br-eth2 (internal)
        port 5: qvo9409f1fe-47
        port 6: qvoc911f327-21
        port 7: qvoe1c63f38-0c
        port 8: qvocb2a5d64-40
        port 9: qvo4e6e174a-ec
root@compute:~# ovs-dpctl dump-flows
recirc_id(0),in_port(5),eth(src=fa:16:3e:56:6b:11,dst=fa:16:3e:64:37:ab),eth_type(0x0800),ipv4(frag=no), packets:0, bytes
:0, used:never, actions:push_vlan(vid=2020,pcp=0),3
recirc_id(0),in_port(3),eth(src=00:e0:2b:00:00:01,dst=00:e0:2b:00:00:00),eth_type(0/0xffff), packets:0, bytes:0, used:nev
er, actions:drop
recirc_id(0),in_port(3),eth(src=fa:16:3e:64:37:ab,dst=fa:16:3e:56:6b:11),eth_type(0x8100),vlan(vid=2020,pcp=0),encap(eth_
type(0x0800),ipv4(frag=no)), packets:0, bytes:0, used:never, actions:pop_vlan,5
root@compute:~#
```

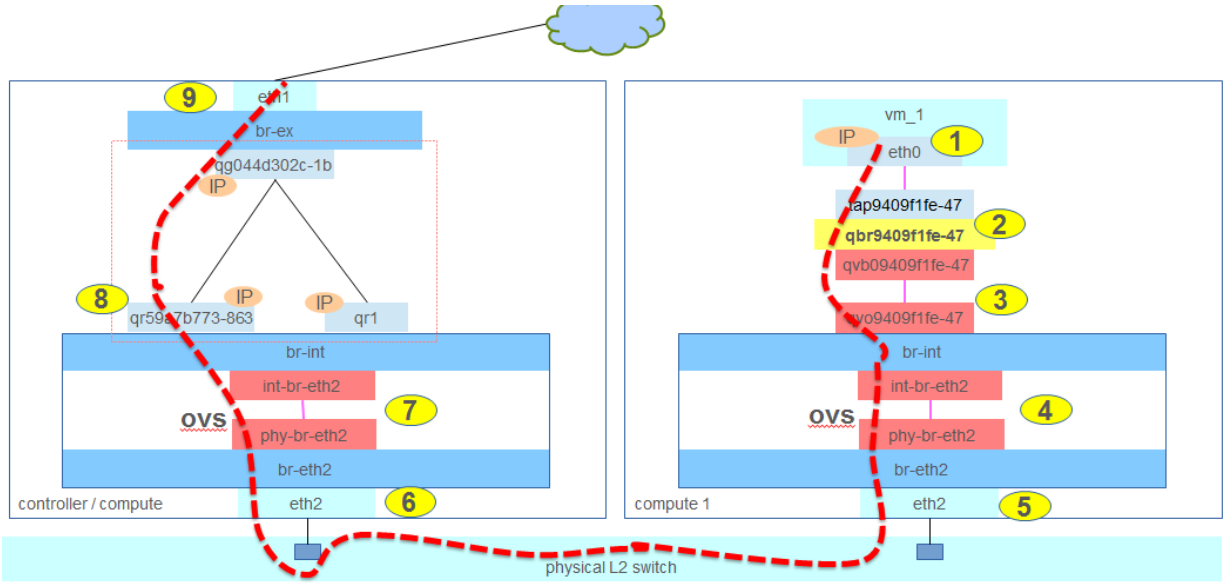
7. tcpdump -i eth2 host 1.1.1.4 –ln

```
root@compute:~# tcpdump -i eth2 host 1.1.1.4 -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 262144 bytes
18:28:42.092049 IP 1.1.1.4 > 1.1.1.8: ICMP echo request, id 770, seq 330, length 64
18:28:42.092819 IP 1.1.1.8 > 1.1.1.4: ICMP echo reply, id 770, seq 330, length 64
18:28:43.092339 IP 1.1.1.4 > 1.1.1.8: ICMP echo request, id 770, seq 331, length 64
18:28:43.093168 IP 1.1.1.8 > 1.1.1.4: ICMP echo reply, id 770, seq 331, length 64
```

8. On the other compute (host name is controller in this case), where vm_5 is, we can check the same points.

```
root@controller:~# ovs-dpctl show
system@ovs-system:
        lookups: hit:959267 missed:63897 lost:2
        flows: 2
        masks: hit:3788329 total:5 hit/pkt:3.70
        port 0: ovs-system (internal)
        port 1: br-mgmt (internal)
        port 2: eth0
        port 3: br-ex (internal)
        port 4: br-int (internal)
        port 5: eth2
        port 6: br-eth2 (internal)
        port 7: tap305ee78c-83 (internal)
        port 8: tap0107e4b2-a8 (internal)
        port 9: qg-044d302c-1b (internal)
        port 10: qr-59a7b773-86 (internal)
        port 11: qvob73e3292-85
        port 12: tap69ae9641-a1 (internal)
root@controller:~# ovs-dpctl dump-flows | grep fa:16:3e:56:6b:11
recirc_id(0),in_port(5),eth(src=fa:16:3e:56:6b:11,dst=fa:16:3e:64:37:ab),eth_type(0x8100),vlan(vid=2020,pcp=0),encap(eth_
type(0x0800),ipv4(frag=no)), packets:0, bytes:0, used:never, actions:pop_vlan,11
recirc_id(0),in_port(11),eth(src=fa:16:3e:64:37:ab,dst=fa:16:3e:56:6b:11),eth_type(0x0800),ipv4(frag=no), packets:0, byte
s:0, used:never, actions:push_vlan(vid=2020,pcp=0),5
root@controller:~#
```

C. VM to external flow



1. vm_1 (1.1.1.4) has a floating IP associated. So vm_1 to ping internet

```
root@controller:~# nova list
+--------------------------------------+------+--------+------------+-------------+---------------------------+
| ID                                   | Name | Status | Task State | Power State | Networks                  |
+--------------------------------------+------+--------+------------+-------------+---------------------------+
| 1ec57719-7d3a-4e4a-af5b-8074bb3b9429 | vm_1 | ACTIVE | -          | Running     | net_1=1.1.1.4, 10.118.47.180 |
```

2. Floating IP for vm_1: **`10.118.47.180`**

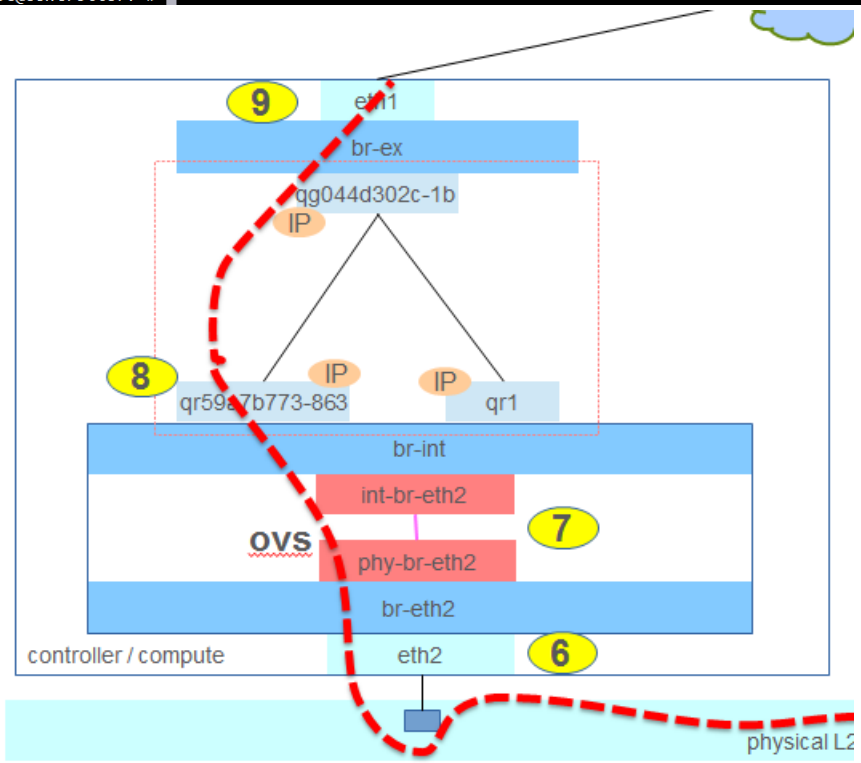
```
$ hostname
vm_1
$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         1.1.1.1         0.0.0.0         UG    0      0        0 eth0
1.1.1.0         0.0.0.0         255.255.255.240 U     0      0        0 eth0
169.254.169.254 1.1.1.1         255.255.255.255 UGH   0      0        0 eth0
$
```

3. In this case, the packet goes out on the same path on the host where vm_1 is. No difference from 2nd cases. The only point to note is, in this case, the packet will be sent to default gateway, which is 1.1.1.1

   Let's see what's going on in the controller where the router is: Let ping the external router interface IP.

```
# hostname
vm_1
# ping 10.118.47.177
PING 10.118.47.177 (10.118.47.177): 56 data bytes
64 bytes from 10.118.47.177: seq=0 ttl=254 time=0.785 ms
64 bytes from 10.118.47.177: seq=1 ttl=254 time=0.763 ms

--- 10.118.47.177 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
```

```
root@controller:~# ovs-dpctl show
system@ovs-system:
        lookups: hit:964795 missed:63919 lost:2
        flows: 2
        masks: hit:3811129 total:5 hit/pkt:3.70
        port 0: ovs-system (internal)
        port 1: br-mgmt (internal)
        port 2: eth0
        port 3: br-ex (internal)
        port 4: br-int (internal)
        port 5: eth2
        port 6: br-eth2 (internal)
        port 7: tap305ee78c-83 (internal)
        port 8: tap0107e4b2-a8 (internal)
        port 9: qg-044d302c-1b (internal)
        port 10: qr-59a7b773-86 (internal)
        port 11: qvob73e3292-85
        port 12: tap69ae9641-a1 (internal)
root@controller:~# ovs-dpctl dump-flows | grep fa:16:3e:56:6b:11
recirc_id(0),in_port(5),eth(src=fa:16:3e:56:6b:11,dst=fa:16:3e:71:3f:0d),eth_type(0x8100),vlan(vid=2020,pcp=0),encap(eth_
type(0x0800),ipv4(frag=no)), packets:0, bytes:0, used:never, actions:pop_vlan,10
recirc_id(0),in_port(10),eth(src=fa:16:3e:71:3f:0d,dst=fa:16:3e:56:6b:11),eth_type(0x0800),ipv4(frag=no), packets:0, byte
s:0, used:never, actions:push_vlan(vid=2020,pcp=0),5
root@controller:~#
```



4. Packet is sent to router namespace:

```
root@controller:~# ip netns
qdhcp-38fe4983-abad-415b-8158-83bcfc5b8d93
qrouter-0e29f8a8-7a9a-4111-a85a-1efb4360983c
```

5. Take 1st 11 character from gateway port (1.1.1.1)

```
root@controller:~# neutron port-list | grep "1.1.1.1"
| 4e6e174a-ecc7-4de4-b7ad-eae17fd489df |      | fa:16:3e:ed:43:08 | {"subnet_id": "61bfdbb4-671d-4f89-bb1f-854ef1a70e45", "ip_address": "1.1.1.19"}
| 59a7b773-8630-4e3f-8b74-a03ddf90baa9 |      | fa:16:3e:71:3f:0d | {"subnet_id": "07326022-9145-4c5b-b62f-efe5b8f8c6ae", "ip_address": "1.1.1.1"}
```

```
root@controller:~# ip netns exec qrouter-0e29f8a8-7a9a-4111-a85a-1efb4360983c ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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
23: qr-59a7b773-86: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:71:3f:0d brd ff:ff:ff:ff:ff:ff
    inet 1.1.1.1/28 brd 1.1.1.15 scope global qr-59a7b773-86
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe71:3f0d/64 scope link
       valid_lft forever preferred_lft forever
```

6. Tcpdump on network namespace

```
root@controller:~# ip netns
qdhcp-9159a5b4-d529-4bc4-95a9-651d6d343eca
qdhcp-38fe4983-abad-415b-8158-83bcfc5b8d93
qrouter-0e29f8a8-7a9a-4111-a85a-1efb4360983c
qdhcp-cffa93c1-2793-4862-86b8-c1de3eaaa5a6
root@controller:~# ip netns exec qrouter-0e29f8a8-7a9a-4111-a85a-1efb4360983c tcpdump -i any -ln
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
01:09:53.905441 ARP, Request who-has 1.1.1.1 tell 1.1.1.4, length 42
01:09:53.905479 ARP, Reply 1.1.1.1 is-at fa:16:3e:71:3f:0d, length 28
01:09:53.906474 IP 1.1.1.4 > 10.118.47.177: ICMP echo request, id 3842, seq 0, length 64
01:09:53.906523 IP 10.118.47.180 > 10.118.47.177: ICMP echo request, id 3842, seq 0, length 64
```

7. Let's check the NAT Rules in qrouter: (which will translate the private ip to floating ip)

```
root@controller:~# ip netns exec qrouter-0e29f8a8-7a9a-4111-a85a-1efb4360983c bash
root@controller:~# iptables -L -t nat
root@controller:~# iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
neutron-l3-agent-PREROUTING  all  --  anywhere             anywhere

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
neutron-l3-agent-OUTPUT  all  --  anywhere             anywhere

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
neutron-l3-agent-POSTROUTING  all  --  anywhere             anywhere
neutron-postrouting-bottom  all  --  anywhere             anywhere

Chain neutron-l3-agent-OUTPUT (1 references)
target     prot opt source               destination
DNAT       all  --  anywhere             10.118.47.180        to:1.1.1.4

Chain neutron-l3-agent-POSTROUTING (1 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ! ctstate DNAT

Chain neutron-l3-agent-PREROUTING (1 references)
target     prot opt source               destination
DNAT       all  --  anywhere             10.118.47.180        to:1.1.1.4
REDIRECT   tcp  --  anywhere             169.254.169.254      tcp dpt:http redir ports 9697

Chain neutron-l3-agent-float-snat (1 references)
target     prot opt source               destination
SNAT       all  --  1.1.1.4              anywhere             to:10.118.47.180

Chain neutron-l3-agent-snat (1 references)
target     prot opt source               destination
neutron-l3-agent-float-snat  all  --  anywhere             anywhere
SNAT       all  --  anywhere             anywhere             to:10.118.47.179
```

8. To be able to check the corresponding flow for this in OVS we need to know the MAC of associated floating IP. (get the 11 character from external gateway port)

```
root@controller:~# neutron port-list | grep 10.118.47.179
| 044d302c-1b28-4472-8b11-76941cf9e529 |      | fa:16:3e:7e:b2:ee | {"subnet_id": "4c70a1ef-d54d-455e-a597-b958e645e609", "ip_address": "10.118.47.17
9"} |
```

```
root@controller:~# ip netns exec qrouter-0e29f8a8-7a9a-4111-a85a-1efb4360983c ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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
23: qr-59a7b773-86: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:71:3f:0d brd ff:ff:ff:ff:ff:ff
    inet 1.1.1.1/28 brd 1.1.1.15 scope global qr-59a7b773-86
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe71:3f0d/64 scope link
       valid_lft forever preferred_lft forever
31: qg-044d302c-1b: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:7e:b2:ee brd ff:ff:ff:ff:ff:ff
    inet 10.118.47.179/29 brd 10.118.47.183 scope global qg-044d302c-1b
       valid_lft forever preferred_lft forever
    inet 10.118.47.180/32 brd 10.118.47.180 scope global qg-044d302c-1b
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe7e:b2ee/64 scope link
       valid_lft forever preferred_lft forever
```

9. Using this MAC, lets dump ovs-flows:

```
root@controller:~# ovs-dpctl show
system@ovs-system:
        lookups: hit:513574 missed:47595 lost:2
        flows: 6
        masks: hit:1907647 total:5 hit/pkt:3.40
        port 0: ovs-system (internal)
        port 1: br-mgmt (internal)
        port 2: eth0
        port 3: br-ex (internal)
        port 4: br-int (internal)
        port 5: eth2
        port 6: br-eth2 (internal)
        port 7: tap305ee78c-83 (internal)
        port 8: tap0107e4b2-a8 (internal)
        port 9: qg-044d302c-1b (internal)
        port 10: qr-59a7b773-86 (internal)
        port 11: qvob73e3292-85
root@controller:~# ovs-dpctl dump-flows | grep "fa:16:3e:7e:b2:ee"
recirc_id(0),in_port(9),eth(src=fa:16:3e:7e:b2:ee,dst=00:30:88:17:86:ed),eth_type(0x0800),ipv4(frag=no), packets:33, bytes:3234, used:0.392s, actions
:push_vlan(vid=68,pcp=0),2
recirc_id(0),in_port(2),eth(src=00:30:88:17:06:ed,dst=fa:16:3e:7e:b2:ee),eth_type(0x8100),vlan(vid=68,pcp=0),encap(eth_type(0x0800),ipv4(frag=no)), p
ackets:34, bytes:3468, used:0.392s, actions:pop_vlan,9
root@controller:~#
```

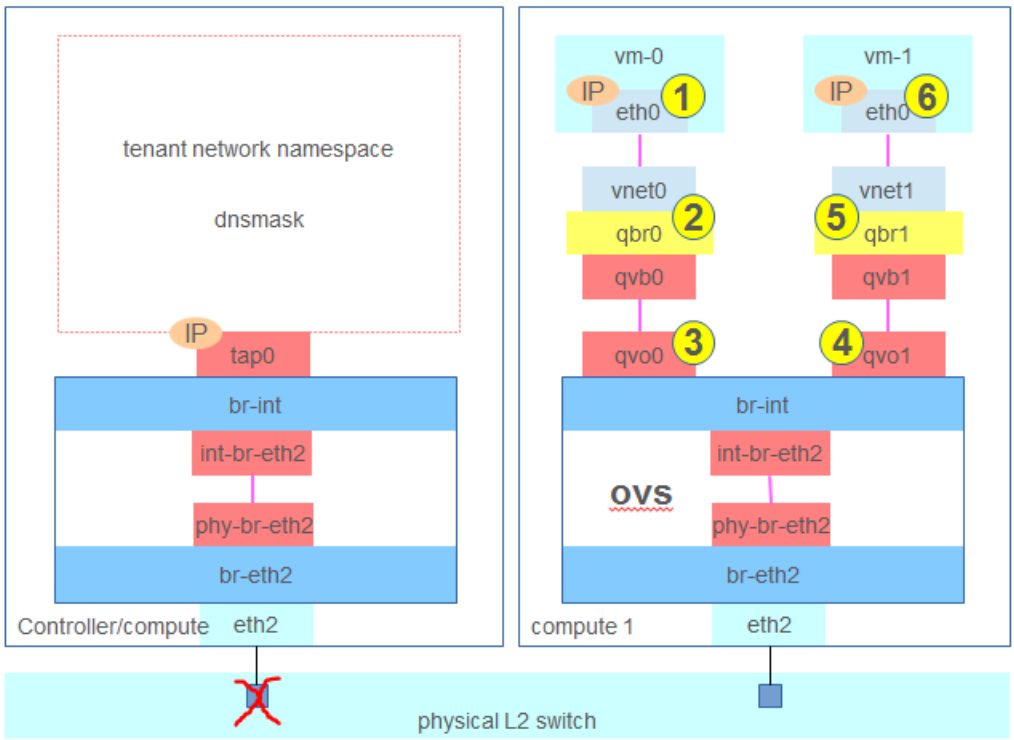
D.

1. **Working scenario**

```
root@controller:~/tmp# ./check.sh
Please enter valid VM Name or ID:vm_1
VM UUID is:  1ec57719-7d3a-4e4a-af5b-8074bb3b9429
VM Host is:  compute
VM is attached to network:  net_1
cffa93c1-2793-4862-86b8-c1de3eaaa5a6
VM's IP is:  1.1.1.4
VM Floating IP is:  10.118.47.180
VM is pingable from namespace
VM Neutron Port will have:  9409f1fe-47
tap9409f1fe-47
Namespace TAP is:  tap305ee78c-83
Namespace IP is:  1.1.1.2
External VLAN for cffa93c1-2793-4862-86b8-c1de3eaaa5a6 Network
is:  2020
Internal VLAN for cffa93c1-2793-4862-86b8-c1de3eaaa5a6 Network
is:  1
NS TAP is: tap305ee78c-83
NS MAC is: fa:16:3e:d4:4d:7b
NS DPCTL Port is:  7
External Port is:  5
Packets sent out from NS, continuing analysis...
Check Source MAC
tmp_MAC_in_flow fa:16:3e:d4:4d:7b
Correct fa:16:3e:d4:4d:7b is seen

===> OVS Datapath Shows *correct* behavior:
ARP on VLAN [2020] is expected on port [7]

Checking Destination MAC is correct
Checking flow is using correct VLAN IDs and forwarding on correct
Port/s


Starting Packet Capture...


Packet Capture successful...
```

```
Starting packet capture analysis...

Checking if ICMP Request is egressing with right parameters
ICMP Request from 1.1.1.2  sent to 1.1.1.4 on VLAN 2020


Next Step: Check correct ICMP Reply is received
ICMP Reply from 1.1.1.4 to 1.1.1.2  received on VLAN 2020


root@controller:~/tmp#
```

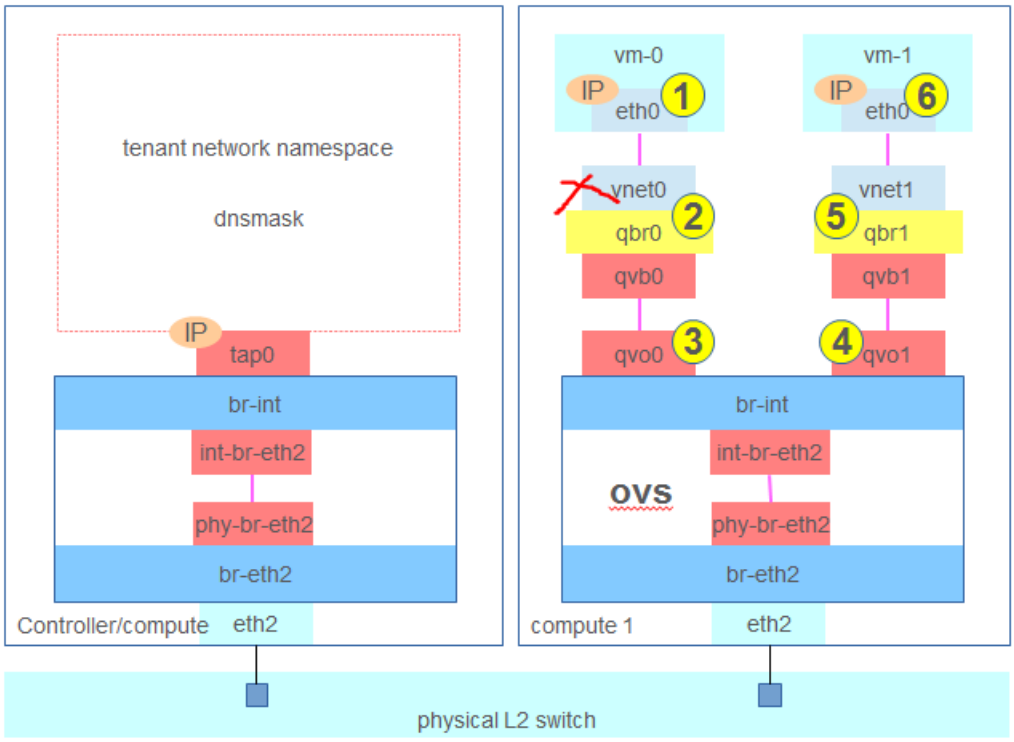
2. **Non-working scenario (failure detected to send the packet towards vm_1)**




```
root@controller:~/tmp# ./check.sh
Please enter valid VM Name or ID:vm_1
VM UUID is:  1ec57719-7d3a-4e4a-af5b-8074bb3b9429
VM Host is:  compute
VM is attached to network:  net_1
cffa93c1-2793-4862-86b8-c1de3eaaa5a6
VM's IP is:  1.1.1.4
```

```
      VM Floating IP is:  10.118.47.180

      ===>Error:
      VM is NOT pingable from namespace
      I'll try to find where the path is broken

      VM Neutron Port will have:  9409f1fe-47
      tap9409f1fe-47
      Namespace TAP is:  tap305ee78c-83
      Namespace IP is:  1.1.1.2
      External VLAN for cffa93c1-2793-4862-86b8-c1de3eaaa5a6 Network
      is:  2020
      Internal VLAN for cffa93c1-2793-4862-86b8-c1de3eaaa5a6 Network
      is:  1
      NS TAP is: tap305ee78c-83
      NS MAC is: fa:16:3e:d4:4d:7b
      NS DPCTL Port is:  7
      External Port is:  5
      Packets sent out from NS, continuing analysis...
      Check Source MAC
      tmp_MAC_in_flow fa:16:3e:d4:4d:7b
      Correct fa:16:3e:d4:4d:7b is seen

      ===> OVS Datapath Shows *correct* behavior:
      ARP on VLAN [2020] will be sent on port [5]

      Checking Destination MAC is correct
      Checking flow is using correct VLAN IDs and forwarding on correct
      Port/s


      Starting Packet Capture...


      ===>Error:
      No Packet is seen leaving host towards target VM
      Check Interfaces and Ports
root@controller:~/tmp#
```

3. **Non-working scenario (failure detected to receive the packet from vm_1)**




```
root@controller:~/tmp# ./check.sh
Please enter valid VM Name or ID:vm_1
VM UUID is:  1ec57719-7d3a-4e4a-af5b-8074bb3b9429
VM Host is:  compute
VM is attached to network:  net_1
cffa93c1-2793-4862-86b8-c1de3eaaa5a6
VM's IP is:  1.1.1.4
VM Floating IP is:  10.118.47.180

===>Error:
VM is NOT pingable from namespace
I'll try to find where the path is broken

VM Neutron Port will have:  9409f1fe-47
tap9409f1fe-47
Namespace TAP is:  tap305ee78c-83
Namespace IP is:  1.1.1.2
External VLAN for cffa93c1-2793-4862-86b8-c1de3eaaa5a6 Network
is:  2020
Internal VLAN for cffa93c1-2793-4862-86b8-c1de3eaaa5a6 Network
is:  1
NS TAP is: tap305ee78c-83
NS MAC is: fa:16:3e:d4:4d:7b
```

```
      NS DPCTL Port is:  7
      External Port is:  5
      Packets sent out from NS, continuing analysis...
      Check Source MAC
      tmp_MAC_in_flow fa:16:3e:d4:4d:7b
      Correct fa:16:3e:d4:4d:7b is seen

      ===> OVS Datapath Shows *correct* behavior:
      ARP on VLAN [2020] will be sent on port [5]

      Checking Destination MAC is correct
      Checking flow is using correct VLAN IDs and forwarding on correct
      Port/s


      Starting Packet Capture...


      Packet Capture successful...


      Starting packet capture analysis...

      Checking if ARP Request is egressing with right parameters
      ARP Request from 1.1.1.2 broadcasted for 1.1.1.4 on VLAN 2020


      Next Step: Check correct ARP Reply is received

      ===>ERROR:
      ARP Request sent successfully but Reply is not recieved
      Problem could be in ToR Switch or Compute Host

root@controller:~/tmp#
```