

[OpenStack 한국 커뮤니티 스터디 B반]

Book: Learning OpenStack Networking (Neutron)
by James Denton

Chapter 4. 가상 스위칭 인프라 만들기

2016년 4월

Ian Y. Choi

목차

- 가상 네트워크 장치
- Neutron에서 지원하는 네트워크 유형
- 플러그인 및 드라이버 선택
- LinuxBridge 사용시 트래픽 Flow 가상화
- OpenvSwitch 사용시 트래픽 Flow 가상화
- ML2 네트워크 플러그인 구성
- LinuxBridge 드라이버 및 에이전트 구성
- OpenvSwitch 드라이버 및 에이전트 구성

가상 네트워크 장치

- 가상 네트워크 인터페이스

Virtual network interfaces

OpenStack uses the libvirt KVM/QEMU driver to provide platform virtualization in default Nova configurations. When an instance is booted for the first time, Neutron assigns a virtual port to each network interface of the instance. KVM creates a virtual network interface called a **tap interface** on the compute node hosting the instance. The tap interface corresponds directly to a network interface within the guest instance. Through the use of a bridge, the host can expose the guest instance to a physical network.

Tip

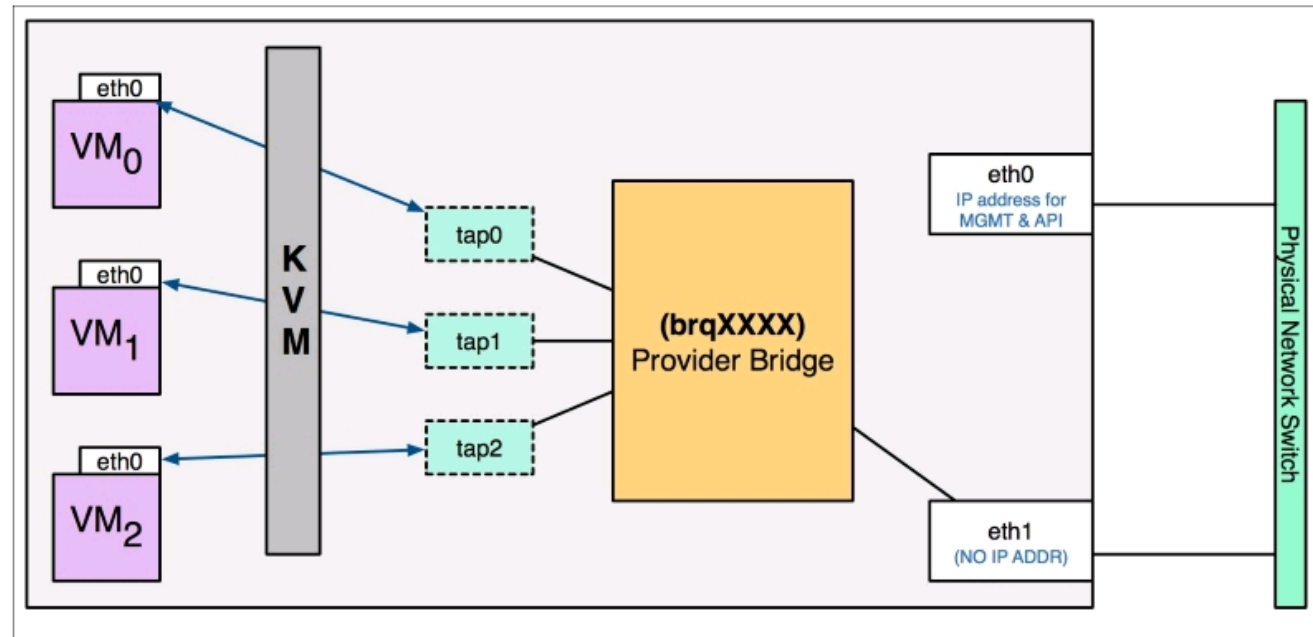
In OpenStack, the name of a tap interface associated with an instance corresponds to the Neutron port UUID, or unique identifier, which the instance is plugged into.

가상 네트워크 장치

- 가상 네트워크 스위치

A **Linux bridge** is a virtual switch on a host that connects multiple network interfaces. When using Neutron, a bridge usually connects a physical interface to one or more virtual or tap interfaces. A physical interface includes Ethernet interfaces, such as `eth1`, or bonded interfaces, such as `bond0`. A virtual interface includes VLAN interfaces, such as `eth1.100`, as well as tap interfaces created by KVM. You can connect multiple physical or virtual network interfaces to a Linux bridge.

The following diagram provides a high-level view of a Linux bridge leveraged by Neutron:



가상 네트워크 장치

- Bridge 인터페이스 구성

Configuring the bridge interface

In this installation, the `eth2` physical network interface will be utilized for bridging purposes. On the controller and compute nodes, configure the `eth2` interface within the `/etc/network/interfaces` file, as follows:

```
auto eth2
iface eth2 inet manual
```

Close and save the file and bring the interface up with the following command:

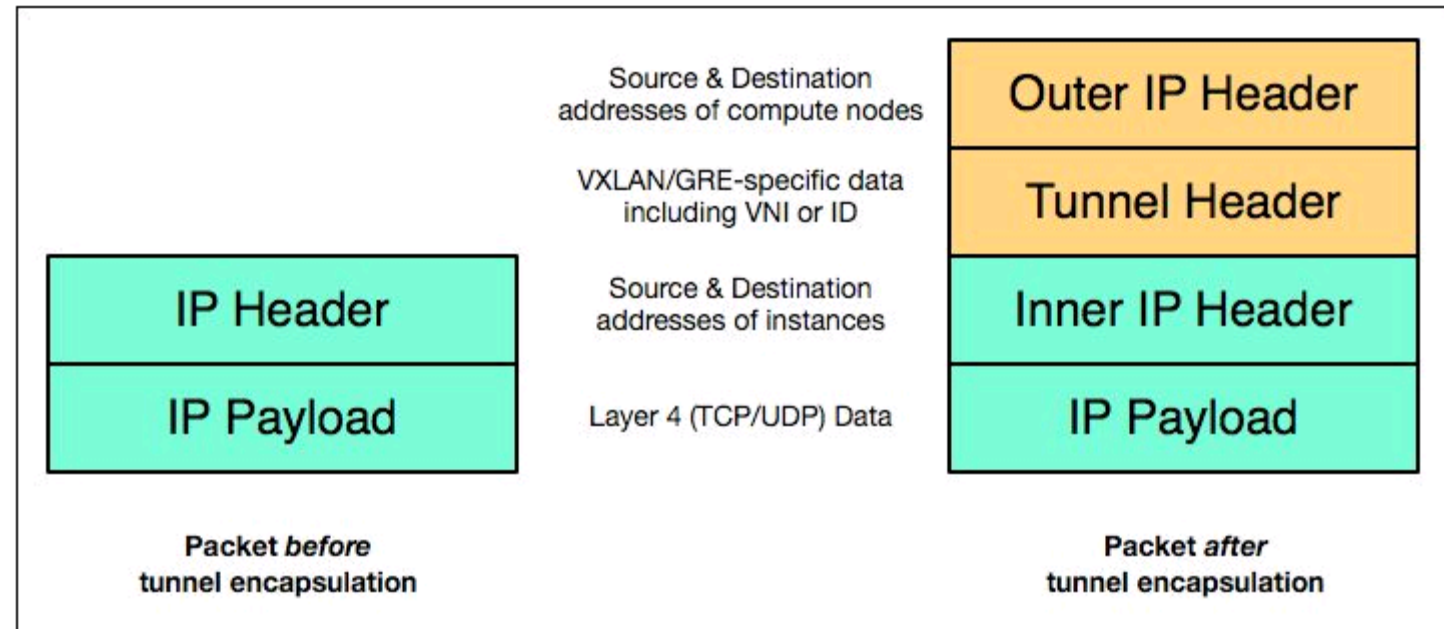
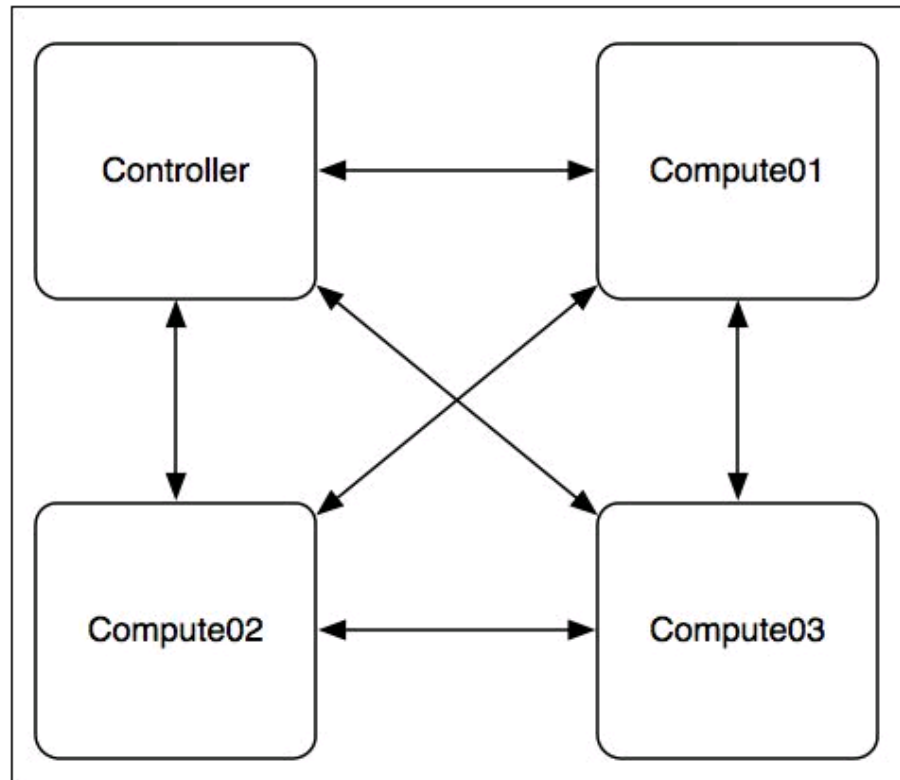
```
# ip link set dev eth2 up
```

Confirm that the interface is in an `UP` state using the `ip link show dev eth2` command, as shown in the following screenshot:

```
root@controller01:~# ip link set eth2 up
root@controller01:~# ip link show dev eth2
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 8c:ae:4c:fe:9a:d0 brd ff:ff:ff:ff:ff:ff
```

가상 네트워크 장치

- 오버레이 네트워크



가상 네트워크 장치

- 오버레이 네트워크 구성시 유의할 점
 - MTU 1500 넘는지 확인 필요

The DHCP agent and `dnsmasq` can be configured to push a lower MTU to instances within the DHCP lease offer. To configure a lower MTU, complete the following steps:

1. On the controller node, modify the DHCP configuration file at `/etc/neutron/dhcp_agent.ini` and specify a custom `dnsmasq` configuration file, as follows:

```
[DEFAULT]
dnsmasq_config_file = /etc/neutron/dnsmasq-neutron.conf
```

2. Next, create the custom `dnsmasq` configuration file at `/etc/neutron/dnsmasq-neutron.conf` and add the following contents:

```
dhcp-option-force=26,1450
```

3. Save and close the file. Restart the Neutron DHCP agent with the following command:

```
# service neutron-dhcp-agent restart
```

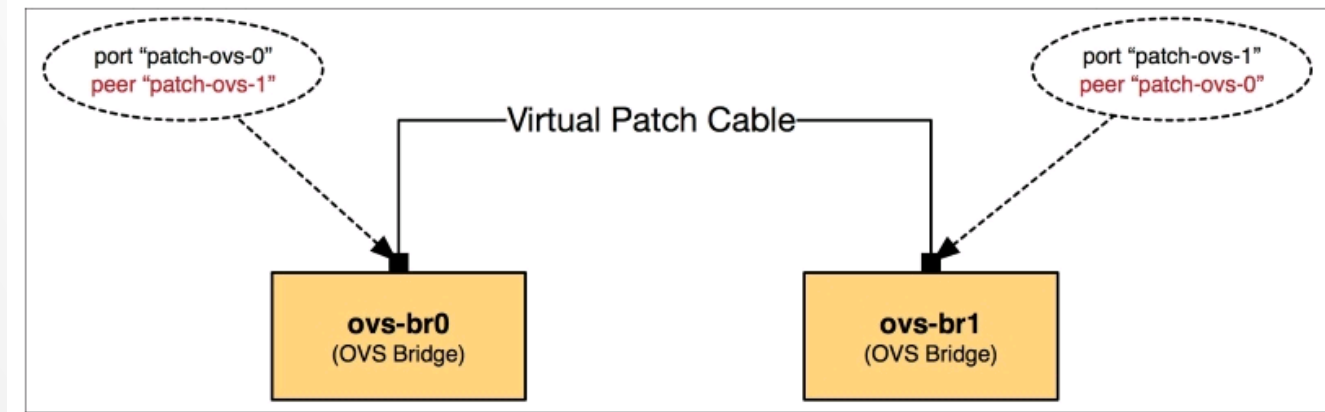
When the instances are later created, the MTU can be observed within the instance using the `ip link show <interface>` command.

Neutron에서 지원하는 네트워크 유형

- Local
 - 다른 네트워크 및 노드로부터 격리된 망 (동일 Compute 노드에서만 통신 가능)
- Flat
 - VLAN 태깅 또는 다른 네트워크 segmentation을 사용하지 않음
- VLAN
 - 802.1q 태깅을 사용하여 Layer 2 브로드캐스트 도메인으로 나뉨.
- VXLAN
 - VNI라고 하는 unique segmentation ID를 하용한 Layer 3 (오버레이)
- GRE
 - UDP 대신 IP Protocol 47을 사용한 encapsulation (Kilo: LinuxBridge로 사용 불가)
 - 참고: <https://support.Microsoft.com/ko-kr/kb/241251>

플러그인 및 드라이버 선택

- LinuxBridge 드라이버
 - Neutron에 의해 관리되는 인터페이스: Tap / Physical / VLAN / VXLAN 인터페이스, Linux Bridge
 - 탭 인터페이스: 게스트 OS가 호스트 내 가상 머신 인스턴스에 연결하기 위해 사용
- OpenvSwitch 드라이버
 - Kernel module
 - vSwitch daemon
 - Database drive (OVSDDB)
- L2 population 드라이버
 - 거대한 오버레이 네트워크에서 스케일 아웃을 위한 브로드캐스트, 멀티캐스트, 유니캐스트 트래픽 지원



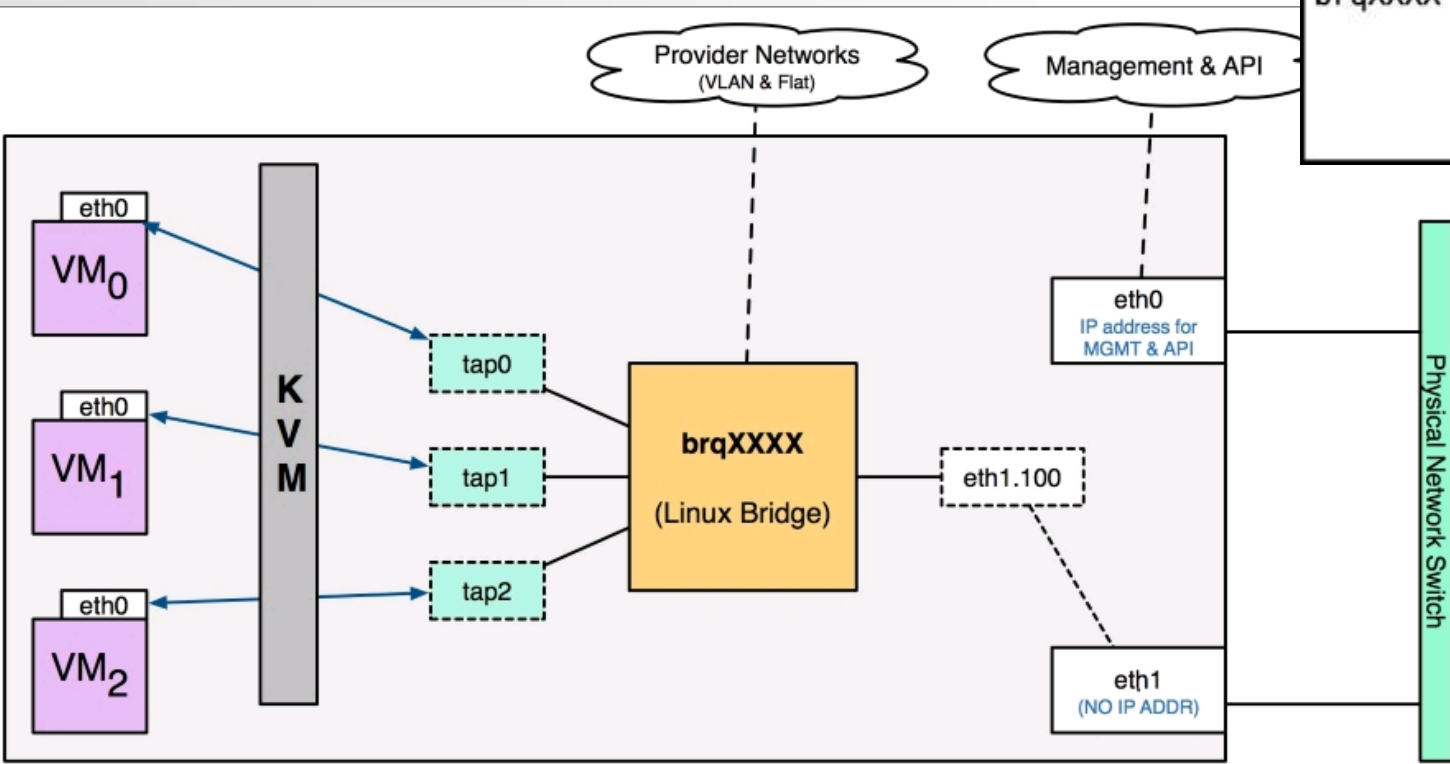
LinuxBridge 사용시 트래픽 시각화

- The tap interface: tapN
- The Linux bridge: brqXXXX
- The VXLAN interface: vxlan-Z (where z is the VNI)
- The VLAN interface: ethX.Y (where X is the interface and Y is the VLAN ID)
- The physical interface: ethX (where X is the interface)

LinuxBridge 사용시 트래픽 시각화 - VLAN

```
# brctl show
```

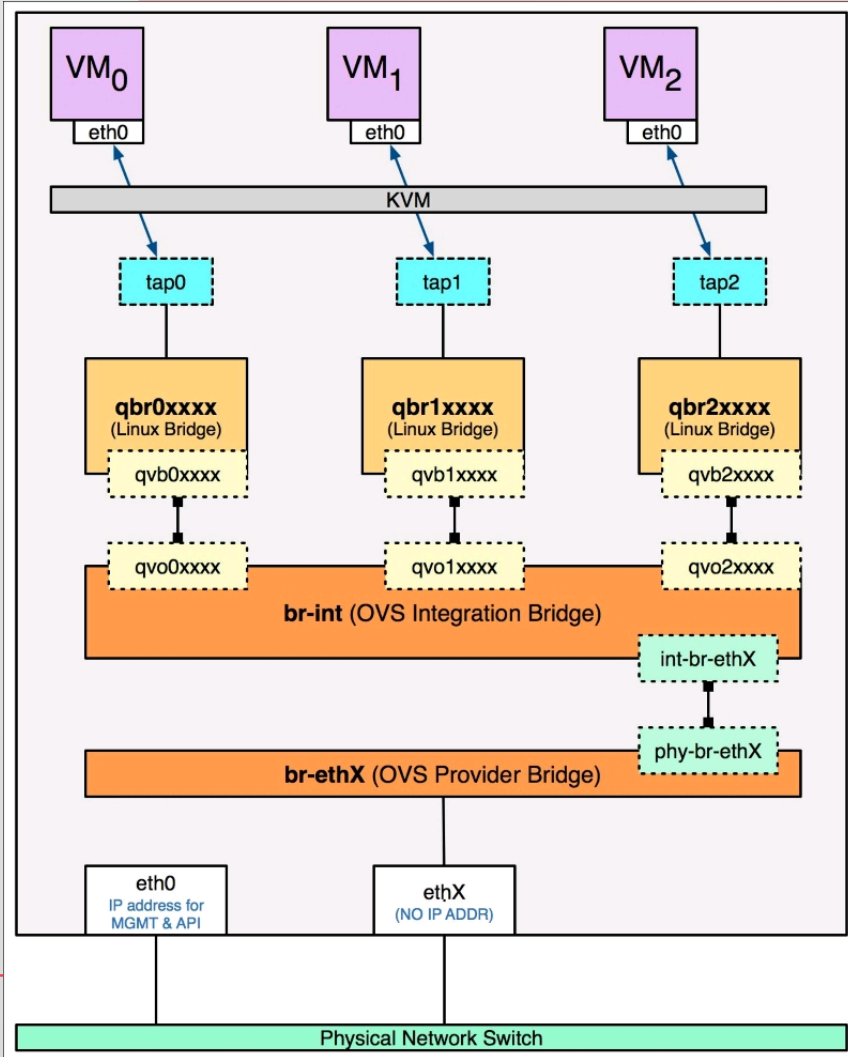
bridge name	bridge id	STP enabled	interfaces
brqXXXX	<based on NIC>	no	eth1.100 tap0 tap1 tap2



OpenvSwitch 사용시 트래픽 시각화

- The tap interface: tapXXXX
- The Linux bridge: qbrXXXX
- The veth pair: qvbXXXX, qvoXXXX
- The OVS integration bridge: br-int
- OVS patch ports: int-br-ethX and phy-br-ethX
- The OVS provider bridge: br-ethX
- The physical interface: ethX
- The OVS tunnel bridge: br-tun

OVS 사용시 트래픽 시각화



```

root@compute01:~# ovs-ofctl show br-int
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000de416cbe2b46
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
6(int-br-eth2): addr:2e:97:d1:79:57:44
  config: 0
  state: 0
  speed: 0 Mbps now, 0 Mbps max
7(patch-tun): addr:56:a3:b0:ab:99:eb
  config: 0
  state: 0
  speed: 0 Mbps now, 0 Mbps max
8(qvo017db302-dc): addr:36:75:da:ef:7a:f6
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
9(qvo7140bc00-75): addr:96:dd:15:3b:14:21
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
LOCAL(br-int): addr:de:41:6c:be:2b:46
  config: PORT_DOWN
  state: LINK_DOWN
  speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
    
```

ML2 구성 옵션

[ml2]
type drivers
mechanism drivers
tenant_network_types

[ml2_type_flat]
flat_networks

[ml2_type_vlan]
network_vlan_ranges

[ml2_type_gre]
tunnel_id_ranges

[ml2_type_vxlan]
vni_ranges

[securitygroup]
firewall_driver
enable_security_group
enable_ipset

```
[ml2]  
...  
type_drivers = local,flat,vlan,gre,vxlan
```

```
[ml2]  
...  
mechanism_drivers = linuxbridge,l2population
```

```
[ml2]  
...  
mechanism_drivers = openvswitch,l2population
```

```
[ml2]  
...  
tenant_network_types = vlan,vxlan
```

Finished!

THANK

YOU

very much!