

openvswitch vxlan 3

..... 4

Open vSwitch firewall 4

Open vSwitch Bridge VxLAN 5

Linux Bridge Open vSwitch Bridge , Veth Pair 6

libvirt network 가 7

NetworkManager-ovs vxlan 7

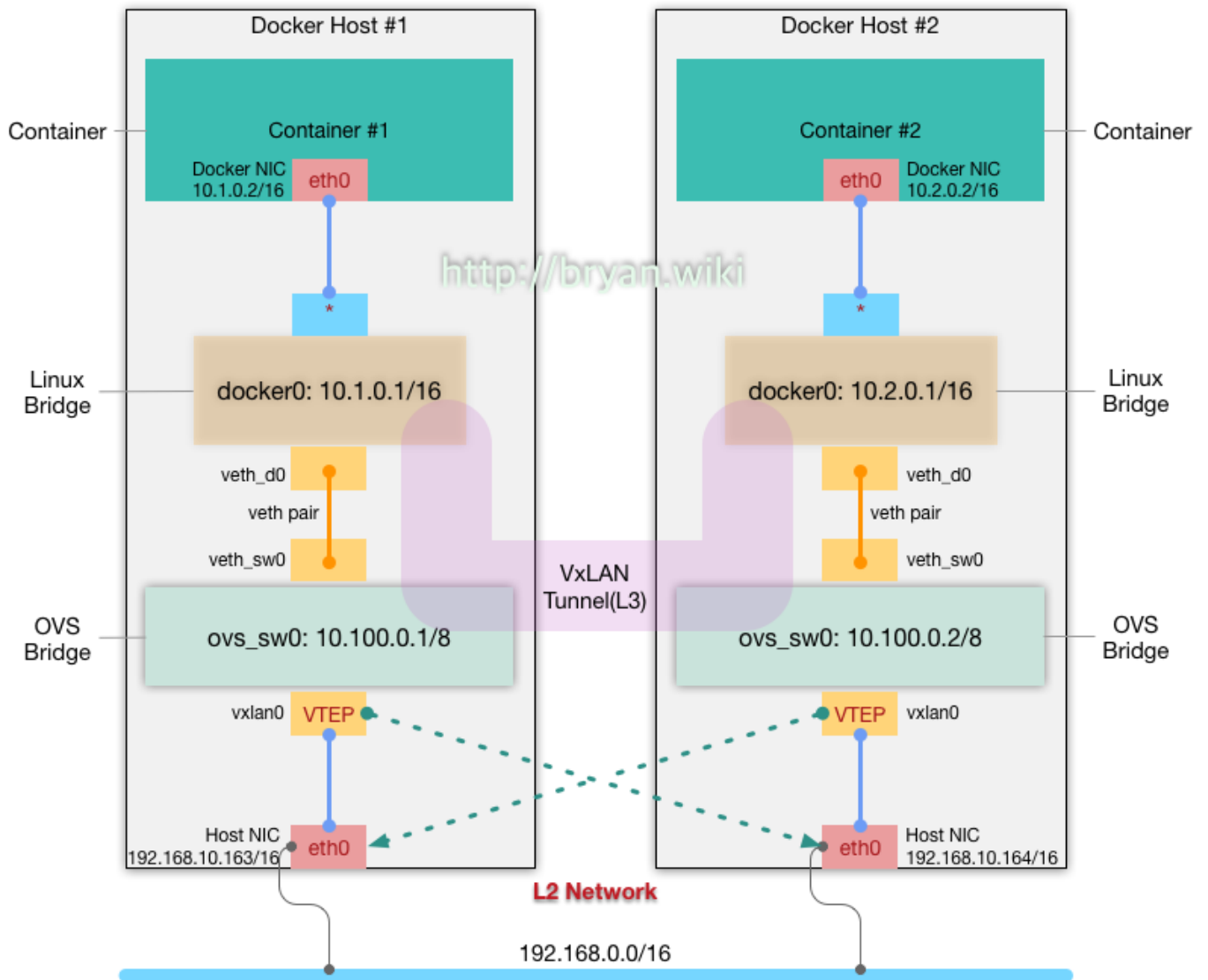
linux network-script 8

..... 9

 NetworkManager-ovs 9

openvswitch vxlan

: <https://bryan.wiki/277> [Bryan's Tech-Log]

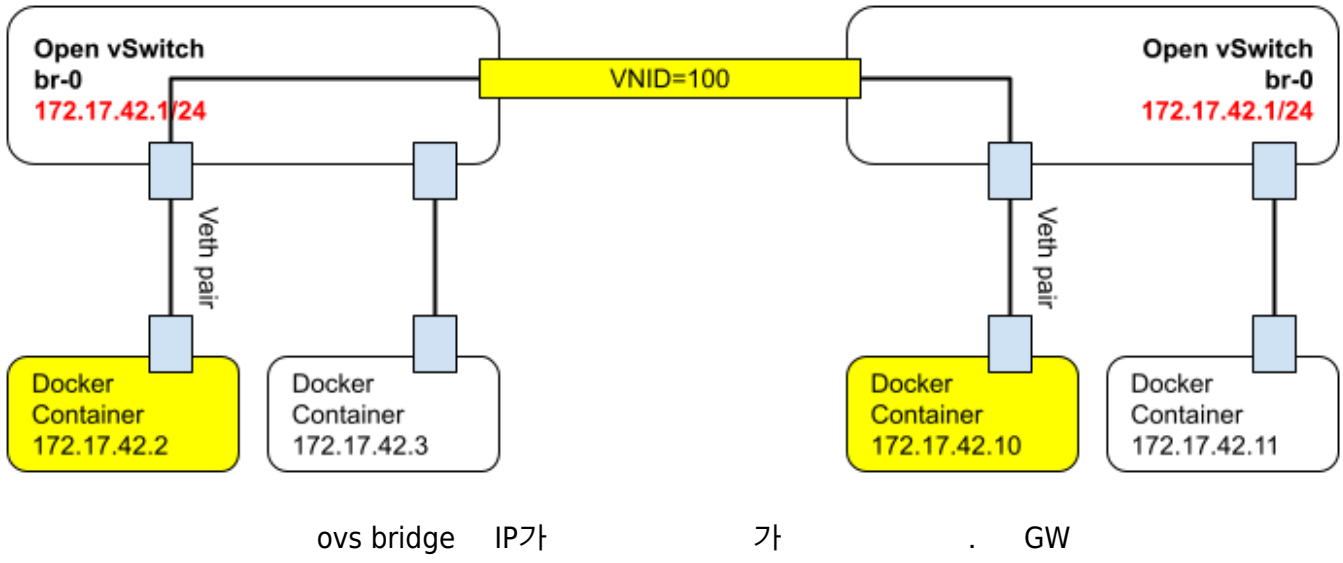


2 가 Flat L2 Network (SDN: Software-Defined-Network 가 , VxLAN

Underlay) 가 가 Overlay)가 가 가

L3 (SDN 가 가 가

Bridge(docker0) (10.x.0.0/16) OVS Bridge 가 (10.0.0.0/8)가 Linux



- Flat Network Docker Host(VM) 2
- Docker Host #1: CentOS 7.3 Minimal Server(1611 ver), 192.168.10.163/16
- Docker Host #2: CentOS 7.3 Minimal Server(1611 ver), 192.168.10.164/16

Open vSwitch firewall

- openvswitch CentOS 7 ovs

```
yum clean all
yum install -y epel-release https://www.rdoproject.org/repos/rdo-release.rpm
yum install -y firewalld docker openvswitch bridge-utils
yum update -y
systemctl start openvswitch firewalld
systemctl enable openvswitch firewalld
```

- VxLAN UDP 4789, 8472 . firewall 2

```
firewall-cmd --add-port=4789/udp --add-port=8472/udp
firewall-cmd --permanent --add-port=4789/udp --add-port=8472/udp
```

- trusted 10.*.*.* 가 (docker0 10.x.0.0/16 : 10.0.0.0/8)
- : docker0 172.17.0.0/16 172.18.0.0/16 ... 172.16.0.0/12

```
firewall-cmd --permanent --zone=trusted --add-source=10.0.0.0/8
```

Open vSwitch Bridge VxLAN

- docker01 Open vSwitch

```
[root@docker01 ~#] ovs-vsctl add-br ovs_sw0

[root@docker01 ~#] ip addr add 10.100.0.1/8 dev ovs_sw0 && ip link set dev ovs_sw0 up

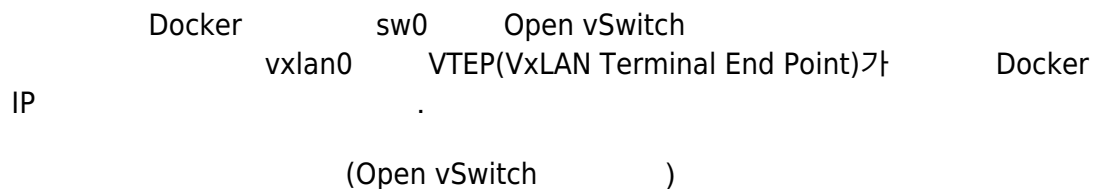
[root@docker01 ~#] ovs-vsctl add-port ovs_sw0 vxlan0 -- set Interface vxlan0 type=vxlan options:remote_ip=192.168.10.164
```

- docker02 Open vSwitch

```
[root@docker02 ~#] ovs-vsctl add-br ovs_sw0

[root@docker02 ~#] ip addr add 10.100.0.2/8 dev ovs_sw0 && ip link set dev ovs_sw0 up

[root@docker02 ~#] ovs-vsctl add-port ovs_sw0 vxlan0 -- set Interface vxlan0 type=vxlan options:remote_ip=192.168.10.163
```



```
[root@docker01 ~]# ping 10.100.0.2
PING 10.100.0.2 (10.100.0.2) 56(84) bytes of data:
64 bytes from 10.100.0.2: icmp_seq=1 ttl=64 time=2.43 ms
64 bytes from 10.100.0.2: icmp_seq=2 ttl=64 time=0.418 ms
64 bytes from 10.100.0.2: icmp_seq=3 ttl=64 time=0.333 ms
64 bytes from 10.100.0.2: icmp_seq=4 ttl=64 time=0.354 ms
64 bytes from 10.100.0.2: icmp_seq=5 ttl=64 time=0.410 ms
^C

--- 10.100.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 0.333/0.790/2.438/0.824 ms
```

10.100.0.1/8 10.100.0.2/8 , VxLAN

Linux Bridge Open vSwitch Bridge , Veth Pair

Linux Bridge (Linux Bridge) Docker
docker0 sw0 VETH pair (pair)
docker0(Linux Bridge) sw0(Open vSwitch Bridge) , Veth Pair End Point() veth_d0
docker0 , veth_sw0 sw0

```
[root@docker01 ~]# ip link add veth_sw0 type veth peer name veth_d0
[root@docker01 ~]# ovs-vsctl add-port ovs_sw0 veth_sw0
[root@docker01 ~]# brctl addif docker0 veth_d0
[root@docker01 ~]# ip link set dev veth_sw0 up
[root@docker01 ~]# ip link set dev veth_d0 up
```

docker02 . End Point docker01
script

```
[root@docker02 ~]# ip link add veth_sw0 type veth peer name veth_d0
[root@docker02 ~]# ovs-vsctl add-port ovs_sw0 veth_sw0
[root@docker02 ~]# brctl addif docker0 veth_d0
[root@docker02 ~]# ip link set dev veth_sw0 up
[root@docker02 ~]# ip link set dev veth_d0 up
```

Docker


```
[root@docker01 ~]# ovs-vsctl show
f3c8825d-73ba-4ee5-a136-3db14a32e990
  Bridge "ovs_sw0"
    Port "veth_sw0"
      Interface "veth_sw0"
    Port "ovs_sw0"
      Interface "ovs_sw0"
        type: internal
    Port "vxlan0"
      Interface "vxlan0"
        type: vxlan
        options: {remote_ip="192.168.10.164"}
  ovs_version: "2.5.2"
```

```
[root@docker01 ~]# brctl show
bridge name bridge id      STP enabled interfaces
docker0    8000.0242f84de852 no    veth_d0
```

libvirt network 가

kvm guest vxlan Virtual Network .
 vxnet0.xml .

```
<network>
  <name>vxnet0</name>
  <forward mode='bridge'/>
  <bridge name='ovs-sw0'/>
  <virtualport type='openvswitch'/>
</network>
```



VxLAN	libvirt Virtual Network	MTU	MTU	
VxLAN		MTU	1500	
가			VxLAN	Underlay
network	JumboFrame			

xml libvirt Virtual Network .

```
virsh net-define vxnet0.xml
virsh net-start vxnet0
virsh net-autostart vxnet0
```

GuestVM vxnet0 .

NetworkManager-ovs vxlan

```
nmcli conn add type ovs-bridge conn.interface ovs-sw0
nmcli conn add type ovs-port conn.interface ovs-sw0 master ovs-sw0
nmcli conn add type ovs-interface slave-type ovs-port conn.interface ovs-sw0
master ovs-sw0 ipv4.method manual ipv4.address 10.100.0.1/24

ovs-vsctl add-port ovs-sw0 vxlan0 -- set Interface vxlan0 type=vxlan
options:remote_ip=192.168.0.102
```



name ovs-bridge ovs-port, ovs-interface interface
가 vm network interface port kvm virtual network

linux network-script

linux network script

```
[root@kvm31 network-scripts]# vim /etc/sysconfig/network-scripts/ifcfg-ovsbr0  
DEVICE=ovsbr0  
ONBOOT=yes  
DEVICETYPE=ovs  
TYPE=OVSBridge  
BOOTPROTO=static  
IPV6INIT=no  
DELAY=0  
IPADDR=10.10.10.1  
PREFIX=8  
HOTPLUG=no
```

가

NIC

```
[root@kvm31 network-scripts]# vim /etc/sysconfig/network-scripts/ifcfg-eno16777736  
HWADDR="00:0C:29:62:F9:D3"  
TYPE="OVSPort"  
DEVICETYPE="ovs"  
OVS_BRIDGE="ovsbr0"  
BOOTPROTO="none"  
NAME="eno16777736"  
UUID="5cc31ab7-c26d-48b6-89f2-a7c933d53cb8"  
ONBOOT="yes"  
NM_CONTROLLED="no"
```

vxlan interface 가

```
DEVICE=vxnet0  
ONBOOT=yes
```



```

DEVICETYPE=ovs
TYPE=OVSTunnel
OVS_BRIDGE=ovsbr0
OVS_TUNNEL_TYPE=vxlan
OVS_TUNNEL_OPTIONS="options:remote_ip=192.168.0.31"

```

[ovs for legacy network-script](#)

- <https://bryan.wiki/277>
- <https://bryan.wiki/279>
- <https://www.joinc.co.kr/w/man/12/OpenVSwitch/VXLAN>
- <https://blog.christophersmart.com/2020/07/27/how-to-create-linux-bridges-and-open-vswitch-bridges-with-networkmanager/>
- <https://blog.oddbit.com/post/2020-02-15-configuring-open-vswitch-with/>

NetworkManager-ovs

```

# subscription-manager repos \
  --enable="rhel-7-server-rpms" \
  --enable="rhel-7-server-extras-rpms" \
  --enable="rhel-7-server-ose-4.6-rpms" \
  --enable="rhel-7-server-optional-rpms" \      <--- Add for
"NetworkManager-ovs" package
  --enable="rhel-7-fast-datapath-rpms"        <--- Add for
"openvswitch2.13" package

```

- <https://developer.gnome.org/NetworkManager/stable/nm-openvswitch.html>
- <https://blog.oddbit.com/post/2020-02-15-configuring-open-vswitch-with/>

From: <https://atl.kr/dokuwiki/> - AllThatLinux!

Permanent link: https://atl.kr/dokuwiki/doku.php/openvswitch%EB%A5%BC_%ED%86%B5%ED%95%9C_vxlan_%EA%B5%AC%EC%84%B1

Last update: 2021/07/05 14:31

