= Session-(5.2) =

Create private network with OVS between LXCs.

Create a private network bridge by OVS

At this step, we suppose that you have created containers and your containers are communicating through the ovs-bridge bridge0

Now to do the following lab we will need 3-Containers as follows;

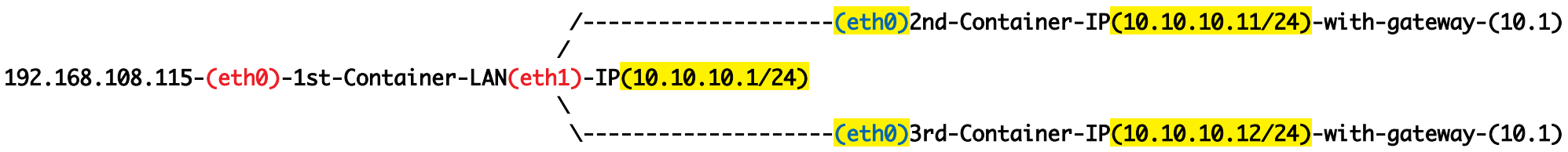
a) First Container will have two interfaces;
    -One will be connected to internet by bridge0
    -And by bridge1 (internal/private bridge) it will be connected to two other containers.

We will create the internal bridge bridge1 in this step.

b) Second Container will be connected to First Container by bridge1 (internal/private bridge). Also it will get internet via First-Container.

c) Third Container will be connected to First Container by bridge1 (internal/private bridge). Also it will get internet via First-Container.

Let's assume/consider our ip-network/diagram as follows;



Let's create the internal bridge:

```
ovs-vsctl add-br bridge1
ip link set bridge1 up
```

Add the line in file /etc/network/if-up.d/netup-script.sh, so that while reboot this bridge will up.

```
vim /etc/network/if-up.d/netup-script.sh
ip link set bridge1 up
```

Now copy two files which we created earlier and modify it by chaning bridge0 to bridge1 (we already have practised this earlier)

```
cp /etc/network/bridge0.dn /etc/network/bridge1.dn
```

```
cp /etc/network/bridge0.up /etc/network/bridge1.up
```

Modify the bridge name; (we might create this file as new, we copied it from previous file to save time)

```
vim /etc/network/bridge1.dn      ; chanage bridge0 to bridge1
```

<or> Run

```
sed -i 's/bridge0/bridge1/g' /etc/network/bridge1.dn
```

```
vim /etc/network/bridge1.up      ; chanage bridge0 to bridge1
```

<or> Run

```
sed -i 's/bridge0/bridge1/g' /etc/network/bridge1.up
```

Run... lxc-zfs-copy shell-script. two create additional containers (if not created yet); you have to stop the source container before copying.

```
lxc-zfs-copy    ; input the source-container-name (group1-node1-ct1) & new-container-name (group1-node1-ct2)
lxc-zfs-copy    ; input the source-container-name (group1-node1-ct1) & new-container-name (group1-node1-ct3)
```

OK, now we have the following 3-containers.

```
group1-node1-ct1    ; the 1st container
group1-node1-ct2    ; the 2nd container
group1-node1-ct3    ; the 3rd container
```

If the above containers were running, you have to stop them first;

And now for the first container add the network configuration lines to activate bridge1 for it,

```
vim /var/lib/lxc/group1-node1-ct1/config        ; first 4-line should already exists add 2nd 4-lines;

#First Network Configuration via bridge0
lxc.net.0.type = veth
lxc.net.0.flags = up
lxc.net.0.script.up = /etc/network/bridge0.up
lxc.net.0.script.down = /etc/network/bridge0.dn

#Second Network Configuration via bridge1
lxc.net.1.type = veth
lxc.net.1.flags = up
lxc.net.1.script.up = /etc/network/bridge1.up
lxc.net.1.script.down = /etc/network/bridge1.dn
```

For 2nd Container; add only network configuration via bridge1; network configuration lines should be as follows;

```
vim /var/lib/lxc/group1-node1-ct2/config

#Network Configuration via bridge1
lxc.net.0.type = veth
lxc.net.0.flags = up
lxc.net.0.script.up = /etc/network/bridge1.up
lxc.net.0.script.down = /etc/network/bridge1.dn
```

For 3rd Container; add only network configuration via bridge1; network configuration lines should be as follows

```
vim /var/lib/lxc/group1-node1-ct3/config

#Network Configuration via bridge1
lxc.net.0.type = veth
lxc.net.0.flags = up
lxc.net.0.script.up = /etc/network/bridge1.up
lxc.net.0.script.down = /etc/network/bridge1.dn
```

Now start those 3-containers;

```
lxc-start -n group1-node1_ct1 -d
lxc-start -n group1-node1_ct2 -d
lxc-start -n group1-node1_ct3 -d
```

Configure an IP 10.10.10.1/24 on eth1 of 1st-container; which is the private network (LAN).

Configure an IP 10.10.10.11/24 and Gateway 10.10.10.1 inside container group1-node1-ct2,
also configure caching-dns (8.8.8.8)

Configure an IP 10.10.10.12/24 and Gateway 10.10.10.1 inside container group1-node1-ct3,
also configure caching-dns (8.8.8.8)

Now verify the communication as follows;

do ping from 10.10.10.1 to 10.10.10.11 & vise-versa
do ping from 10.10.10.1 to 10.10.10.12 & vise-versa

do ping from 10.10.10.11 to 10.10.10.1 & vise-versa
do ping from 10.10.10.11 to 10.10.10.12 & vise-versa

do ping from 10.10.10.12 to 10.10.10.1 & vise-versa
do ping from 10.10.10.12 to 10.10.10.11 & vise-versa

= CONFIGURE NAT FOR PRIVATE NETWORK

We have to configure NAT (network address translation) in the 1st-container to provide internet to two internal containers.

#Open Terminal of 1st-container and run the following commands,

```
apt update
apt install iptables vim

sudo vim /usr/bin/nat.sh

# Now write the following content
```

```
#!/bin/bash
echo 1 > /proc/sys/net/ipv4/ip_forward

iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD
iptables -F -t mangle
iptables -F -t nat
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
iptables -t nat -X
iptables -t nat -Z
iptables --table nat -F
iptables --delete-chain
iptables --table nat --delete-chain
iptables -t mangle --delete-chain


iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.108.115
```

```
chmod +x /usr/bin/nat.sh
```

Run...

```
sudo nat.sh
```

= CONFIGURE BRIDGE WITH VLAN by OVS

if anyone want to configure bridge (here internal bridge1) with vlan tagging, following is the way to do that;

Apply the following commands in the host-machine (not inside the container)

Delete the previous plain bridge;

```
ovs-vsctl del-br bridge1
```

Now add vlan bridge

```
ovs-vsctl add-br vlink0
ovs-vsctl add-port vlink0 vlink0.100 tag=100 -- set interface vlink0.100 type=internal

ip link set vlink0 up
ip link set vlink0.100 up
```

Make changes so that this vlan-bridge up while booting/rebooting

```
vim /etc/network/if-up.d/netup-script.sh      ; add the following two lines
ip link set vlink0 up
ip link set vlink0.100 up
```

Now create vlan up/down script for containers

```
vim /etc/network/vlan100.up
#!/bin/bash
BRIDGE="vlink0"
ovs-vsctl --may-exist add-br $BRIDGE
ovs-vsctl --if-exists del-port $BRIDGE $5
ovs-vsctl --may-exist add-port $BRIDGE $5 tag=100

vim /etc/network/vlan100.dn
#!/bin/bash
ovsBr=vlink0
ovs-vsctl --if-exists del-port ${obsBr} $5

chmod +x /etc/network/vlan100.*
```

Then, change the lxc-config file to remove bridge1 and add vlan100

```
lxc-stop -n group1-node1-ct1

vim /var/lib/lxc/first-container/config ; change bridge1 to vlan100

lxc-stop -n group1-node1-ct2
lxc-stop -n group1-node1-ct3

vim /var/lib/lxc/second-internal-container/config ; change bridge1 to vlan100
vim /var/lib/lxc/third-internal-container/config  ; change bridge1 to vlan100


lxc-start -n <all-containers)


<Optional> if anyone want to add more vlan

ovs-vsctl add-br vlink1
ovs-vsctl add-port vlink1 vlink1.101 tag=101 -- set interface vlink0.101 type=internal
```