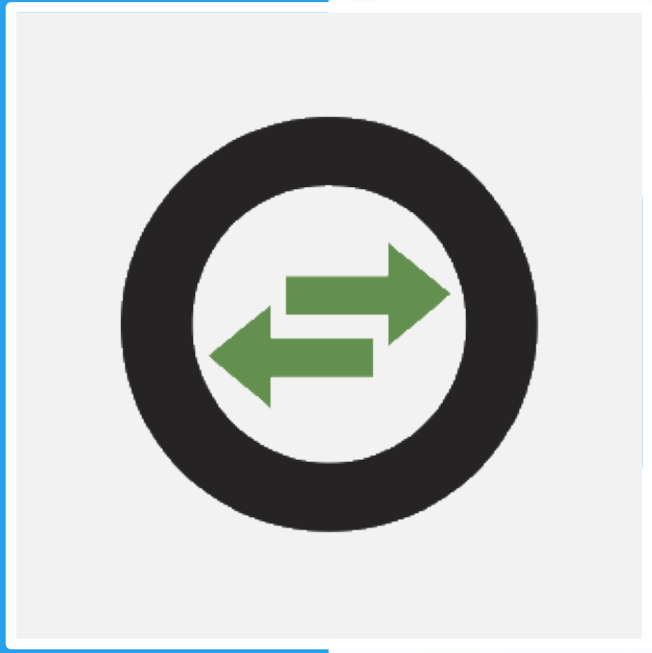


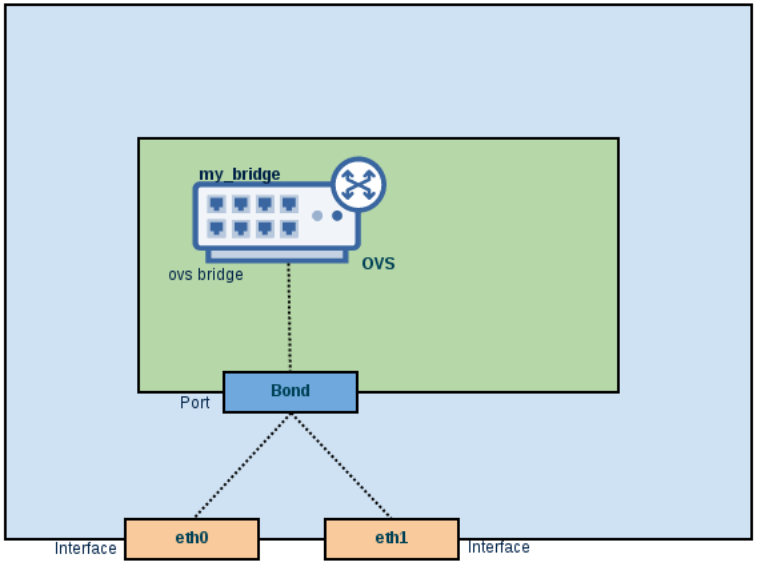
Open vSwitch

Day 02, Session 2.4

Open vSwitch

- Open vSwitch is a multilayer software/virtual switch used to inter connect virtual machines in the same host and between different hosts.
- OpenvSwitch supports many of the features you already familiar with, assuming you worked with switches before:
  - **VLAN tagging**
  - **LACP**
  - **STP**
  - **QOS**
  - **Tunneling protocols (GRE, VXLAN)**
  - **SPAN, RSPAN**

Virtual Switch Overview



We can see in the drawing,
there is one bridge named '**my_bridge**'
which was created using openvswitch.

Each bridge can have multiple ports and each port consists of one or more interfaces. In our example, there is one port named '**Bond**', which is an actual bond of two physical interfaces (eth0 and eth1)

Installing open vSwitch

> yum install -y openvswitch

- systemctl start openvswitch
- apt install openvswitch-switch

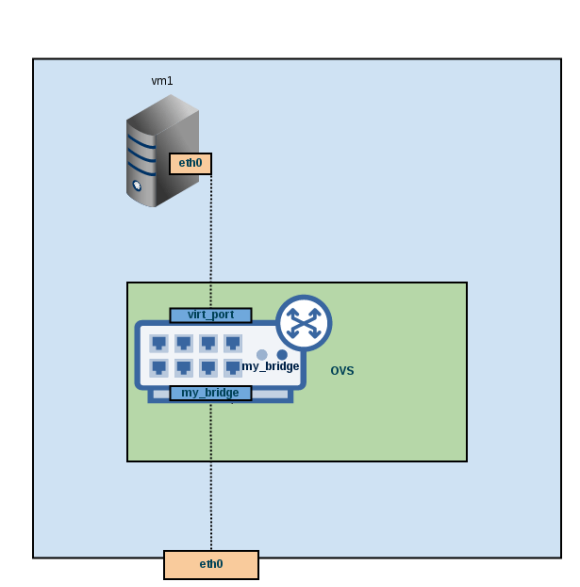
we will be able to use the '**ovs-vsctl show**' command

> sudo ovs-vsctl show

ovs_version: "2.5.0"

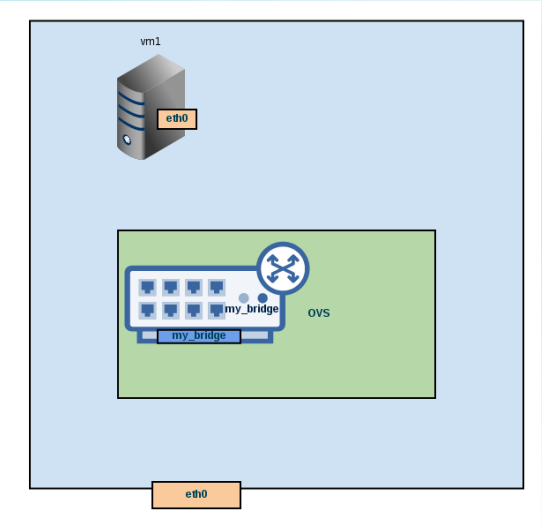
Connect a VM

Our goal is to connect a newly created VM to the internet but we want to connect it through an ovs bridge



Our goal is to achieve something similar to the above diagram

Connect a VM



To create an ovs bridge

> sudo ovs-vsctl add-br my_bridge

we added a new bridge, named '**my_bridge**'

To verify our bridge

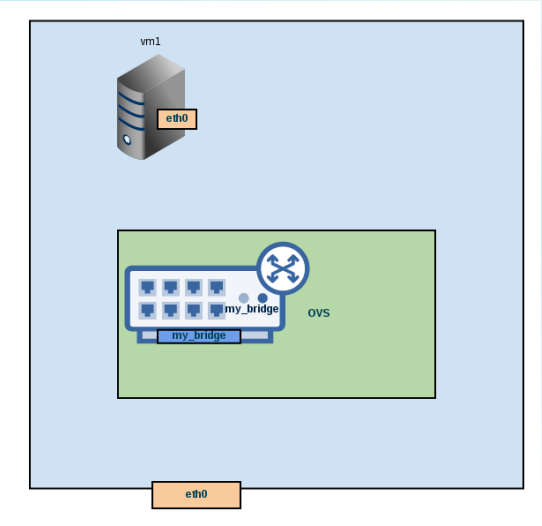
> ovs-vsctl show

```
9e72385f-ed0a-40fd-97f3-21d49cbf60f3
    Bridge my_bridge
        Port my_bridge
            Interface my_bridge
                type: internal
    ovs_version: "2.5.0"
```

Bring the 'my_bridge' interface up

> ip link set my_bridge up

Connect a VM



Note that our newly created bridge is not directly connected to our physical interface (eth0). Let's connect eth0 interface to '**my_bridge**'

> sudo ovs-vsctl add-port my_bridge eth0

We just lost connectivity to the external world.
(the most popular check today is probably pinging 8.8.8.8).

This is because eth0 is now connected to our bridge and not to the default IP stack of the system. Our system still trying to reach the external network directly through eth0. In order to recover our connectivity to the external network, we need to do two things:

- Remove eth0 address, since we no longer reach the internet directly through eth0
- Assign my_bridge with address so we can reach the internet through it

IP stack -> my_bridge -> eth0

Connect a VM

Removing eth0 current address

> ip addr del 192.168.121.52/24 dev eth0

Verify with '**ip a**' that eth0 indeed has no IP address

we will run **dhclient** to configure 'my_bridge', so it can be allocated with an IP address

> dhclient my_bridge

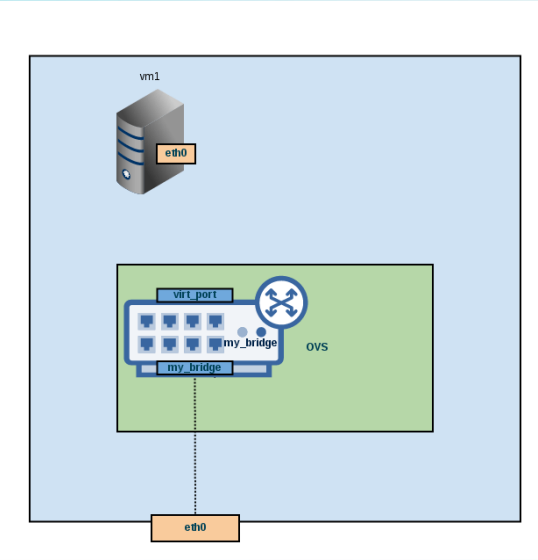
Now that '**my_bridge**' has an IP address, we should be able to reach the internet once again

Connect a VM

Bring the interface up

> ip **link** set virt_port up

Connect a VM



Add our newly created device to our ovs bridge

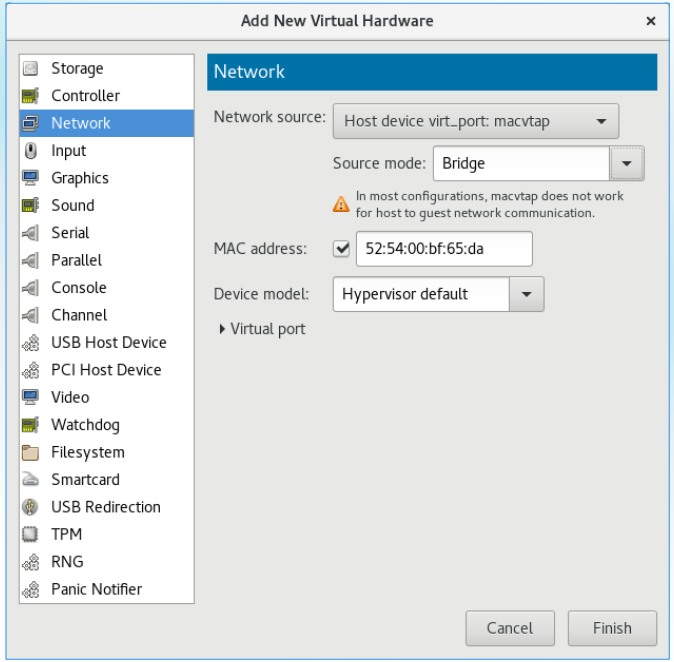
> sudo ovs-vsctl add-port my_bridge virt_port

Verify with '**ovs-vsctl show**' that our ports are connected to '**my_bridge**'

> sudo ovs-vsctl show

```
9e72385f-ed0a-40fd-97f3-21d49cbf60f3
    Bridge my_bridge
        Port my_bridge
            Interface my_bridge
                type: internal
        Port "virt_port"
            Interface "virt_port"
    ovs_version: "2.5.0"
```

Connect a VM



Connect ‘virt_port’ to the VM

- Go to the virtual machine properties
- Click at the bottom on ‘add hardware’ button
- Next, you choose ‘Network’ and in ‘Network source’ you choose the ‘virt _port’ device and ‘Bridge’ for ‘Source mode’

Congrats, we reached our goal. The virtual machine is now able to reach the internet, through our ovs bridge ‘**my_bridge**’

Some More

Commands	Descriptions
ovs-vsctl show	Print summary of the ovs database content
ovs-vsctl add-br <bridge_name>	Add a new bridge
ovs-vsctl del-br <bridge_name>	Delete existing bridge
ovs-vsctl add-port <bridge_name> <port_name>	Add a new port in the specified bridge
ip a (= 'ip addr' = 'ip address')	Displays addresses and their properties
ip addr del <IP address/CIDR> dev <device>	Remove the specified address from the specified device
ip set **link** <interface_name> up	Bring an interface up
ip tuntap add mode tap <device_name>	Add TAP device

Any Questions ?

Thank you