SERVICE FRACTAL

# Running OVS on Containers

Shivaram Mysore, Founder/CEO
@shivaram_mysore

# Why run OVS on Containers?

- Easy upgrade to run the latest version; no extra library dependencies (ex. `Python3`)
- Run on Fedora CoreOS
- Run multiple OVS on a single host
- Software based on demand deployment and programmability

# Linux Namespaces

- The entire OS shares the same routing table and the IP address.  This namespace forms a cluster of all global system resources which can only be used by the processes within the namespace, providing resource isolation.
- Docker containers use this technology to form their own cluster of resources which would be used only by that namespace, i.e. that container. Hence every container has its own IP address and work in isolation without facing resource sharing conflicts with other containers running on the same system.
- Linux's network namespaces are used to glue container processes and the host networking stack. Docker spawns a container in the containers own network namespace and later on runs a `veth` pair between the container namespace and the host network stack.

# Capabilities

With OVS being run as a container, it needs some privileges to access network and system resources. In Docker, this is controlled by providing "Capability" permissions to the running container. For OVS to work, Capabilities such as "`SYS_MODULE`", "`NET_ADMIN`" and "`SYS_NICE`" are required.

Reference: `http://man7.org/linux/man-pages/man7/capabilities.7.html`

# IPTables

- Docker extensively uses `iptables` to provide isolation amongst its services and filtering of traffic.

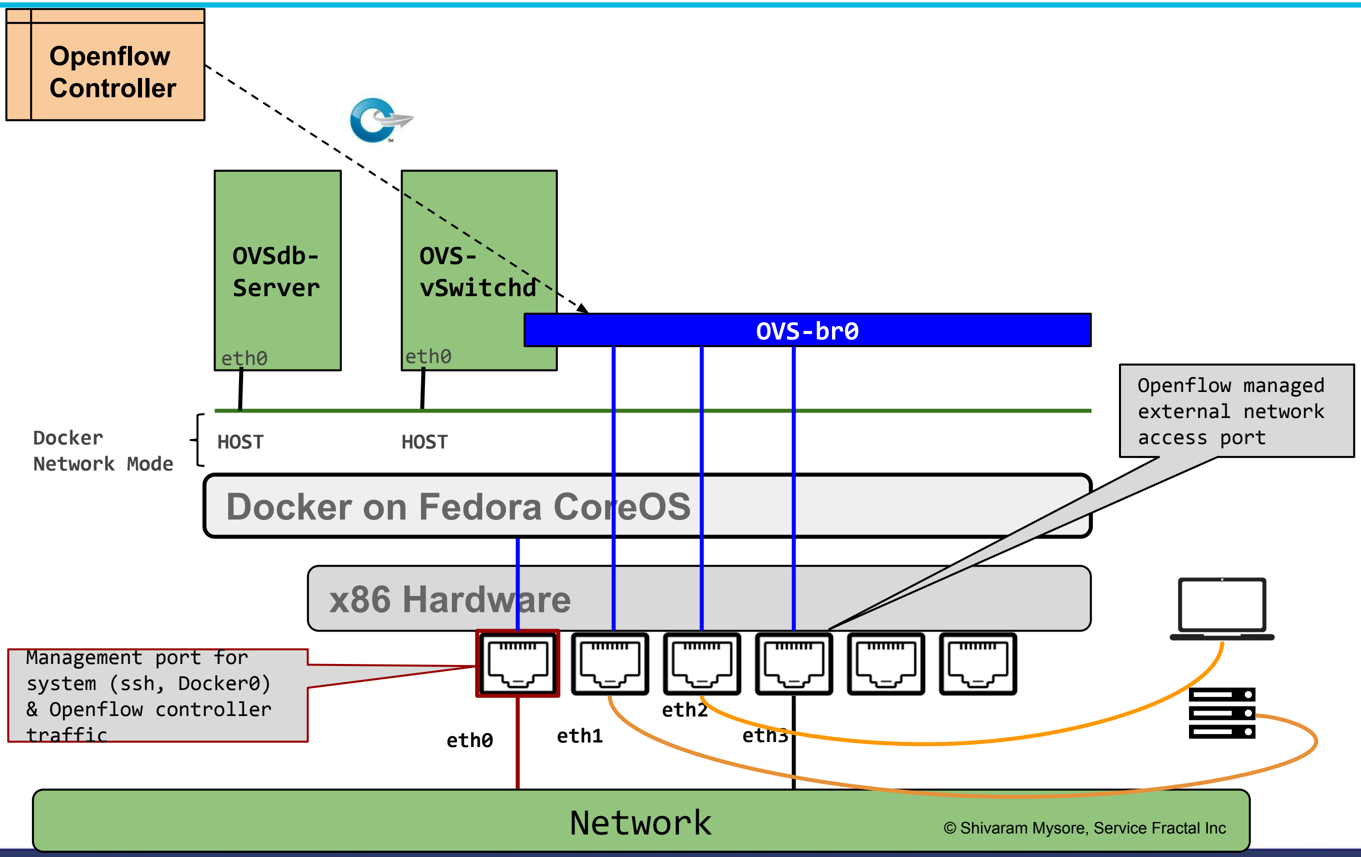
- Mostly, we may never have to touch this feature unless, the underlying system has a custom `iptables` rules.

# Docker Networking - cheat sheet

| Built-in network drivers | Container Interfaces | Connected to | IP Address | Outbound Traffic | Inbound Traffic | Container |
|---|---|---|---|---|---|---|
| **bridge** (default) | lo, eth0 (provided by veth pair) | docker0 (config via --bridge) | Default private IP space 172.17.0.0/16 | goes through an iptables MASQUERADE rule | goes through an iptables DNAT rule | Can have its own routes, iptable rules, etc |
| **null** (none) | Only lo interface | | | Can't send or receive network traffic | | |
| **host** | eth0 interface | host | Default - uses docker IP addresses; Can have additional IP if more interfaces are attached | Network traffic doesn't have to go through NAT, bridge, or veth | | Sees and able to access Host network interfaces (native performance) |
| **Container** | re-uses the network stack of another container | | shares with this other container the same interfaces, IP address(es), routes, iptables rules, etc. | | | Containers communicate over lo interface |

*Custom Networks - Weave, etc out of scope*

# Deployment



© Shivaram Mysore, Service Fractal Inc

# Running OVSDB-Server

```
$ docker run \
    --name=ovsdb-server \
    --cap-add=NET_ADMIN \
    --cap-add=SYS_MODULE \
    --cap-add=SYS_NICE \
    --network=host \
    --volume=/lib/modules:/lib/modules \ ← Needed to load Kernel Modules
    --volume=/home/core/ovs/log:/var/log/openvswitch \
    --volume=/home/core/ovs/var/lib/openvswitch:/var/lib/openvswitch \
    --volume=/home/core/ovs/var/run/openvswitch:/var/run/openvswitch \
    --volume=/home/core/ovs/etc/openvswitch:/etc/openvswitch \
    --security-opt label=disable \
    --privileged \ ← May be removed; Not tested without removal
    servicefractal/ovs:latest ovsdb-server
```

# Running OVS-vswitchd

```
$ docker run \
    --name=ovs-vswitchd \
    --cap-add=NET_ADMIN \
    --cap-add=SYS_MODULE \
    --cap-add=SYS_NICE \
    --network=host \
    --volumes-from=ovsdb-server \
    --security-opt label=disable \
    --privileged \
    servicefractal/ovs:latest ovs-vswitchd
```

← Consistency between ovsdb-server and this ovs-vswitchd container volumes

More info: https://github.com/servicefractal/ovs

# Open Questions

1. How to connect Container(s) (ex. `ngnix`) to this OVS bridge running on a Container?
   a. If someone has thoughts, please drop a note.  Thanks!
   b. Note: the model applied on standard linux install of OVS to move a container namespace does not work here.
2. DPDK enabled OVS

# Additional Info

1. Code, documentation, PRs, Issues & suggestions: https://github.com/servicefractal/ovs
2. Contact: **shivaram** dot **mysore** at **gmail.com** or **OVS-discuss** mailing list

# Thanks to our hardware partners