One Network To Rule Them All: Open, Scalable & Integrated Networking for Containers and VMs

Phil Estes <estesp@us.ibm.com>  @estesp
Kyle Mestery <mestery@us.ibm.com>  @mestery
**Container Introduction**

“Containers are a lie we tell a process.” - Mark Shuttleworth

**Why Containers?**
- Extremely lightweight (only a Linux process)
- Fast startup (process start + small overhead for containment setup)
- Container ecosystem has created simple and standard packaging model for applications
- Great fit with current development and cloud-era initiatives: a) CI/CD; b) microservice architectures

---

**Containment Process**

- Mount
- IPC
- Network
- User
- UTS
- PID

Linux Kernel
Container Introduction: Networking

> There is no such thing as (Linux) container networking!

You may create a new network namespace in Linux.

- Processes in this network namespace will have a unique list of network interfaces
- This namespace will have its own routing table
- Methods for creating, connecting and routing these virtual interfaces is up to the implementor of the container runtime.
- Many runtimes default to using a Linux bridge with virtual ethernet pairs assigned to the container network namespace; this is the original Docker default networking style
What Is Software Defined Networking?

Software-defined networking (SDN) is an umbrella term encompassing several kinds of network technology aimed at making the network as agile and flexible as the virtualized server and storage infrastructure of the modern data center.

Fundamentally, it’s about:

- Operational scale
- Agility and speed
- Moving complexity from HW to SW
Neutron Abstractions

- **Virtual Machine** (or container)
- **Virtual Interface (VIF)**
- **Virtual Port**
- **Virtual Network**
- **Virtual Subnet**

- **vm1**
  - IP: 10.10.10.100
- **vm2**
  - IP: 10.10.10.200

- **net1**
  - 10.10.10.0/24

- **vm1** and **vm2** are connected to **net1** through **Virtual Interface (VIF)**.
...You Can Then Build This:

**Public Network**
10.50.50.0/24

**Tenant A**
- **net1**
  - 192.168.1.0/0
- **net2**
  - 192.168.5.0/0

- **A-vm1**
  - IP: 192.168.1.5

- **A-vm2**
  - IP: 192.168.1.5
  - IP: 192.168.5.2

- **A-vm3**
  - IP: 192.168.5.9

**Tenant B**
- **net1**
  - 192.168.1.0/0
- **net2**
  - 192.168.9.0/0

- **B-vm1**
  - IP: 192.168.1.3

- **B-vm2**
  - IP: 192.168.1.5
  - IP: 192.168.9.3

- **B-vm3**
  - IP: 192.168.9.7
Open vSwitch 101

- Open vSwitch is a **virtual switch** which runs on a host or hypervisor.
- Open vSwitch is composed of:
  - Linux Kernel module
  - ovs-vswitchd daemon
  - ovsdb-server daemon
Open Virtual Networking 101

• OVN is a virtual networking system which:
  • manages Open vSwitch across a cluster of hosts
  • integrates with a cloud management system (CMS)

• OVN adds the following components to an OVS environment:
  • ovn-northd daemon
  • Central ovsdb-server with OVN NB and SB databases
  • ovn-controller daemon on each host in the cluster
There is more than one **model** for Linux container networking:

- **Container Network Interface (CNI)**
  - Developed via CoreOS appc project; used by K8s, rkt, others

- **Container Network Model (CNM)**
  - Developed by Socketplane team; acquired by Docker
  - **libnetwork** is an implementation of CNM
  - Project Kuryr supports CNM by way of implementing a libnetwork plugin
Ecosystem Players: Container Networking

Growing list of ecosystem players for container networking

- Project Calico
- Weave.works
- flannel
- OVN (Open Virtual Network)

Docker has enabled pluggability at several layers in the engine: storage, networking, authorization, layer (graph) store.

Several 3rd party networking plugins available for libnetwork.
Container Networking: libnetwork

Diagram showing network sandboxes with endpoints connected to front and backend networks.
Project Kuryr: Docker Networking for Neutron

- Our required **network plugin** for Docker’s **libnetwork** API translation to **Neutron** is found in **Project Kuryr**
- **Kuryr** is a Docker network plugin that utilizes the Neutron API to:
  - Provide network services to Docker containers and will provide containerized images for common Neutron network plugins

Kyle Mestery @mestery · Oct 10
Current status: Neutron + OVN + Kuryr #ohmy

https://github.com/openstack/kuryr
Kuryr: Docker to Neutron Mapping

Network
- Neutron Network

Endpoint
- Neutron Port

IPAM
- Neutron Subnet

Join/Leave
- `plug()` and `unplug()`
- requires code for different vif types: OVS, LB, ...

Sandbox

Endpoints

Network
Advantages of Kuryr

- Use your existing OpenStack Neutron networking layer!
- Tie together your VMs and containers (and bare metal with Ironic!) into the same virtual networking layer!
IBM Bluemix: Built on Open (Networking)

- **Bluemix container service** runs on OpenStack
  - Neutron provides networking layer to Docker containers
- **Next-generation container service implementation** using Kuryr
  - Will allow unified networking across containers, VMs, and bare metal
  - Continue to exploit underlayer of Neutron + OVS / OVN improvements
Demo

Demo Components:

- Docker (1.10.3)
- Kuryr (Newton)
- Neutron (Newton)
- OVN (from master)
Awesome! Questions?