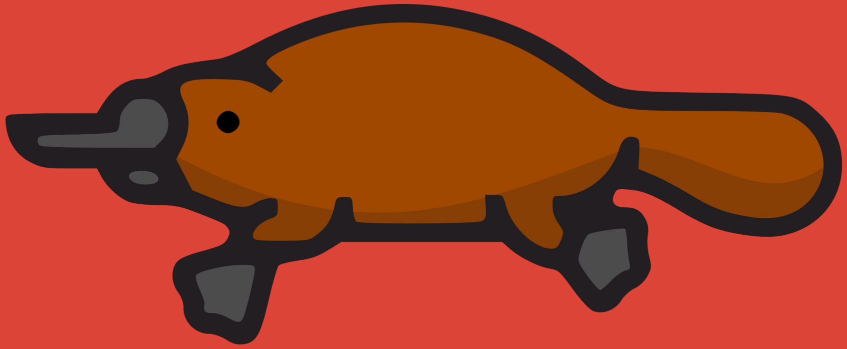


Kuryr & Fuxi

OpenStack networking and storage for Docker Swarm containers

Hongbin Lu
Antoni Segura Puimedon

A bit of background

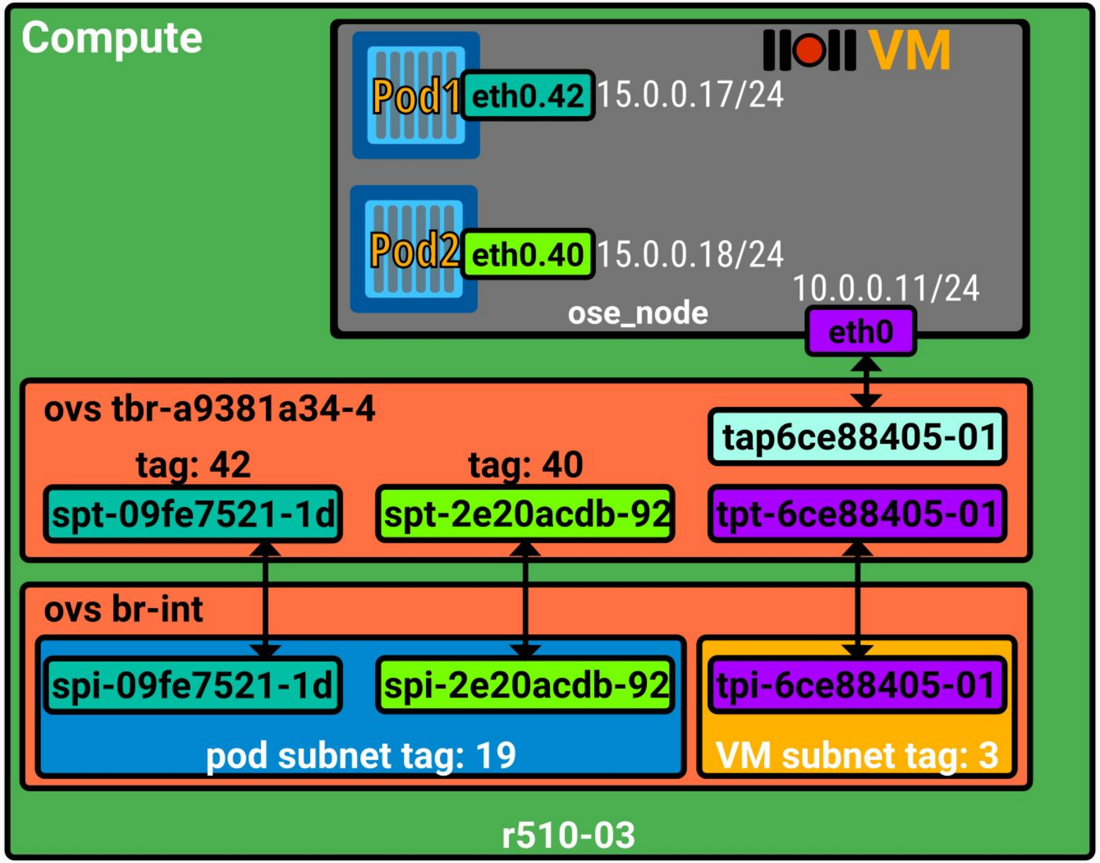
- Kuryr-libnetwork
    - Started during Liberty
    - Brings Neutron Networking to Docker containers
    - Targets bare-metal and container-in-VM
- Fuxi
    - Started during Mitaka
    - Brings Cinder and Manila volumes to bare-metal Docker containers

Kuryr-libnetwork: container-in-VM

- Three modes:
  - Neutron trunk using Vlan
  - Macvlan
  - Ipvlan
- Independent of which Neutron plugin you use
- Gets one Neutron port to each container running on Nova instances
- Security groups can target single containers
- Needs keystone credentials on the nova instances' /etc/kuryr/kuryr.conf
- Helps adoption of container workloads allowing you to leverage your VM SDN

Neutron trunk ports

- Available since Newton
- Implemented by most Neutron plugins
- Works with most guests since only vlan support is needed
- Security groups are still applied between subports



Macvlan/ipvlan

- Leverage Neutron allowed address pairs (Havana)
- No need to tag/untag as in vlan trunk
- Use the VM interface as link device (must be specified in conf)
- Both have several modes of operation

Kuryr-libnetwork: container-in-VM summary

Driver	Container limit	Performance	Security	Neutron Availability	Segmentation
Trunk ports	~4094/VM	3	SG	Ocata+	Vlan tags
ipvlan	≤ subnet size	1	/	Mitaka	L3
macvlan	≤ subnet size	2	/	Mitaka	L2

Getting started with Kuryr libnetwork

- Requirements
    - Bare-metal:
        - Neutron l2 agent
        - Docker
    - On nova instance:
        - Guest OS must support ipvlan/macvlan or have trunk parent port
        - Neutron firewall must be ovs (ref impl)
- Installation
    - As in the snippet

```
Neutron.conf snippet

[securitygroup]
firewall_driver=openvswitch

[DEFAULT]
serviceplugins=trunk #other plugins can be enabled
```

```
celebdor@calcifer ~/ $ docker plugin install
kuryr/libnetwork2
Plugin "kuryr/libnetwork2" is requesting the following
privileges:
 - network: [host]
 - mount: [/var/run/openvswitch]
 - mount: [/var/log/kuryr]
 - mount: [/etc/kuryr]
 - capabilities: [CAP_NET_ADMIN]
Do you grant the above permissions? [y/N]
```

Getting started with Kuryr libnetwork

- Configuration
  - /etc/kuryr/kuryr.conf on the host as always
  - TLS available
  - http://tech.paulcz.net/2016/01/secure-docker-with-tls/
  - Ovs native firewall support

kuryr.conf

```
[binding]

driver = kuryr.lib.binding.drivers.vlan
link_iface = eth0

[default]
ssl_cert_file=/etc/kuryr/kuryr.crt
ssl_key_file=/etc/kuryr/kuryr.key
enable_ssl=True
```

Kuryr.json (plugin config)

```
        {
          "Name": "kuryr",
          "Addr": "https://127.0.0.1:23750",
          "TLSConfig": {
            "InsecureSkipVerify": false,
            "CAFile": "/var/lib/kuryr/certs/ca.pem",
            "CertFile": "/var/lib/kuryr/certs/cert.pem",
            "KeyFile": "/var/lib/kuryr/certs/key.pem"
          }
        }
```

Getting started with Kuryr libnetwork

- Running
  - Standard docker api
  - Special kuryr specific opts
  - Allows connecting to pre-existing Neutron resources

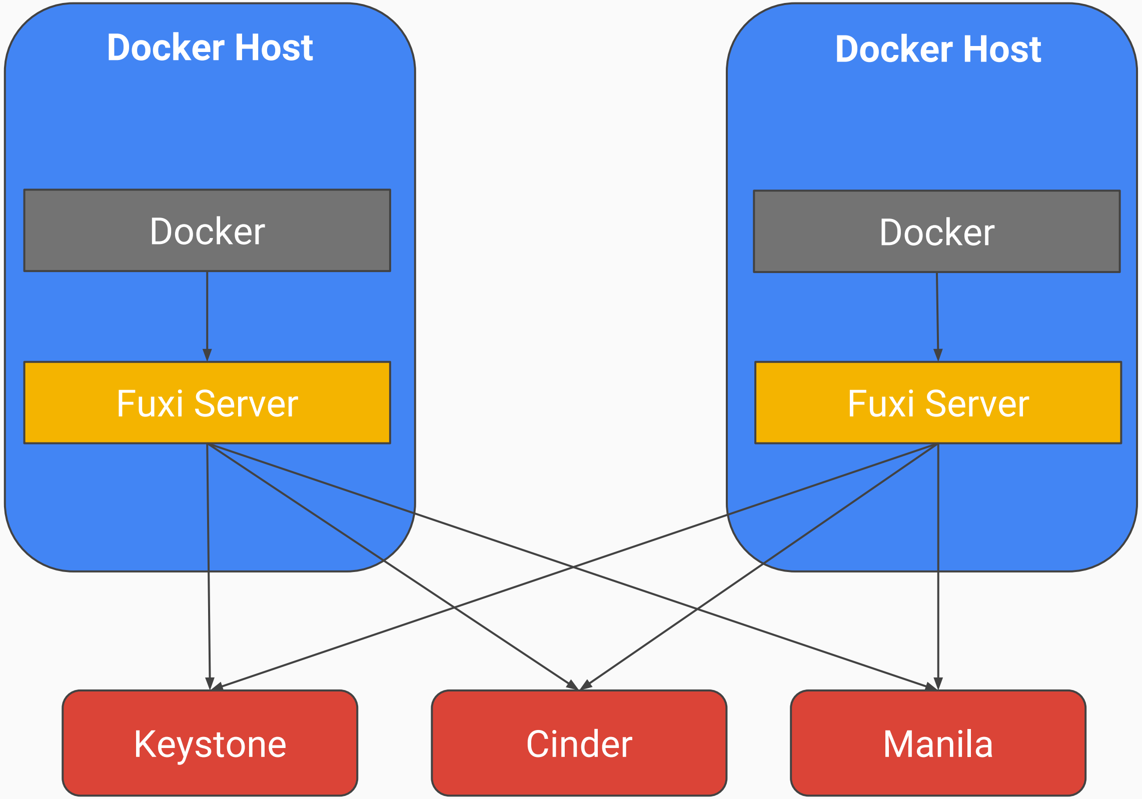
```
$ # Create network
$ docker network create --driver=kuryr --ipam-driver=kuryr \
    --subnet 10.10.0.0/16 --gateway 10.10.0.1 foo

$ # Create network mapping to existing neutron net
$ docker network create -driver=kuryr --ipam-driver=kuryr \
    --subnet=10.10.0.0/24 --gateway=10.10.0.1 \
    -o neutron.net.uuid=d98d1259-03d1-4b45-9b86-b039cba1d90d \
    my_reused_net

$ # Run container
$ docker run -it --net=foo --ip=10.0.0.5 alpine

$ # Reuse port (must be unbound)
$ docker run -it --net=my_resused_net --ip=10.10.0.2 alpine

$ # Run container with open ports (sgs)
$  docker run --net=foo --expose=1234-1238/udp -it alpine
```

Fuxi volumes

- Cinder
    - Allow provisioning Cinder volume in Docker
    - Or create with existing Cinder volume
    - Cluster node
- Manila
    - Similar to Cinder but using Manila shares instead

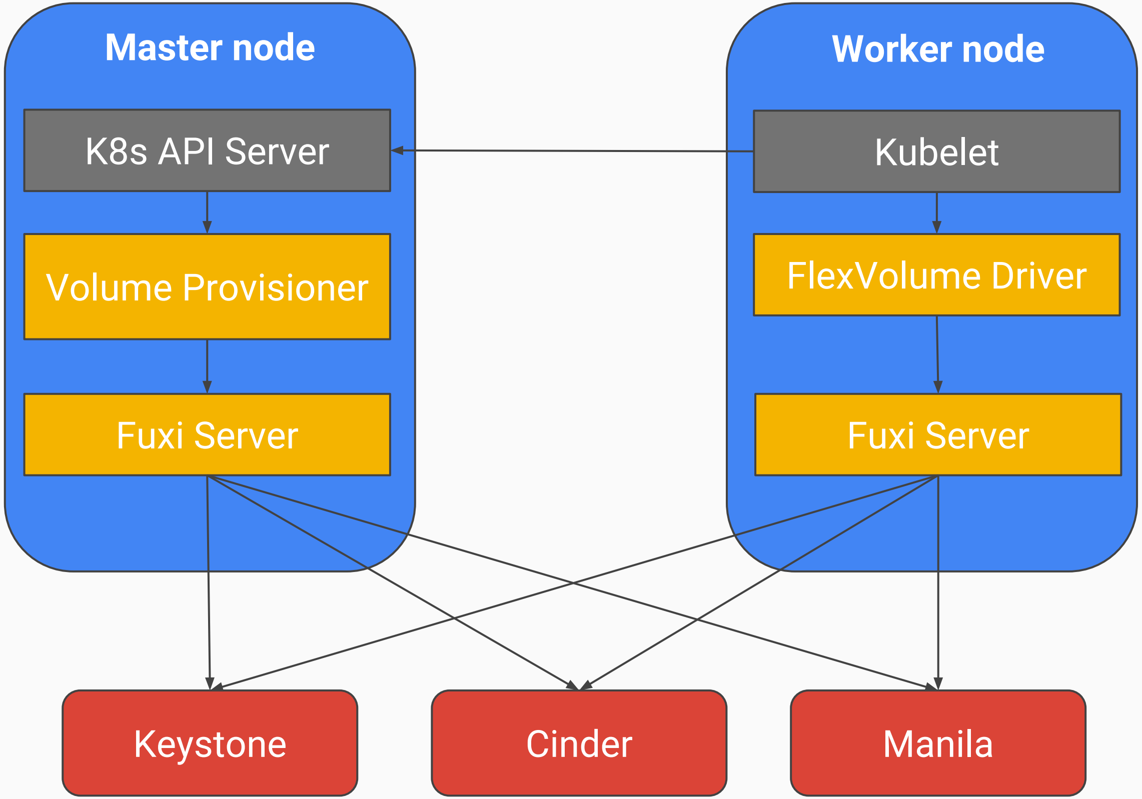
Fuxi - Docker

- Fuxi Server is a docker remote volume plugin
- Fuxi Server running in each host that runs Docker
- Docker will calls Fuxi Server for creating Docker volumes
- Fuxi Server translates the request into API calls to Cinder/Manila.



Fuxi - Kubernetes

- Volume provisioner running in the Kuryr-Kubernetes Controller
- Fuxi Server is the same as in Docker Swarm
- Spec merged in Pike
- Stay tuned for Queens!



Getting started with fuxi

- Requirements
    - Bare-metal:
        - Docker
        - Storage client depending on cinder/manila backend
- Installation
    - From pip
- Configuration

```
## fuxi.conf snippet
[DEFAULT]
volume_providers=cinder,manila
my_ip=ip_of_the_docker_worker_node

[cinder]
…
volume_connector = osbrick
fstype = ext4

[manila]
…
volume_connector = osbrick
```

Getting started with fuxi

- Running
  - Standard docker API
  - Specific option depending on volume type and provider

```
$ # Create cinder volume
$ docker volume create --driver fuxi --name my_vol \
      --opt size=1 \
      --opt fstype=ext4 \
      --opt multiattach=true \
      --opt volume_provider=cinder

$ # Reuse existing cinder volume
$ docker volume create --driver fuxi --name existing_vol \
      --opt size=1 \
      --opt volume_id=125da087-8b89-46de-97e4-c275c9a5bd1a\
      --opt volume_provider=cinder

$ # Create generic manila volume
$ docker volume create --driver fuxi --name my_vol \
      --opt volume_provider=manila

$ # Run container with volume
$ docker run -v my_vol:/var/www httpd
```

Roadmap

- Fuxi
  - Docker plugin installation
  - Docker TLS support
  - Flexvolume driver
  - Kuryr-kubernetes integration
  - Kubernetes CI
- Kuryr-libnetwork
  - Global scope support in swarm mode https://bugs.launchpad.net/kuryr-libnetwork/+bug/1668486
  - Docker swarm multinode CI

Demo

Q&A