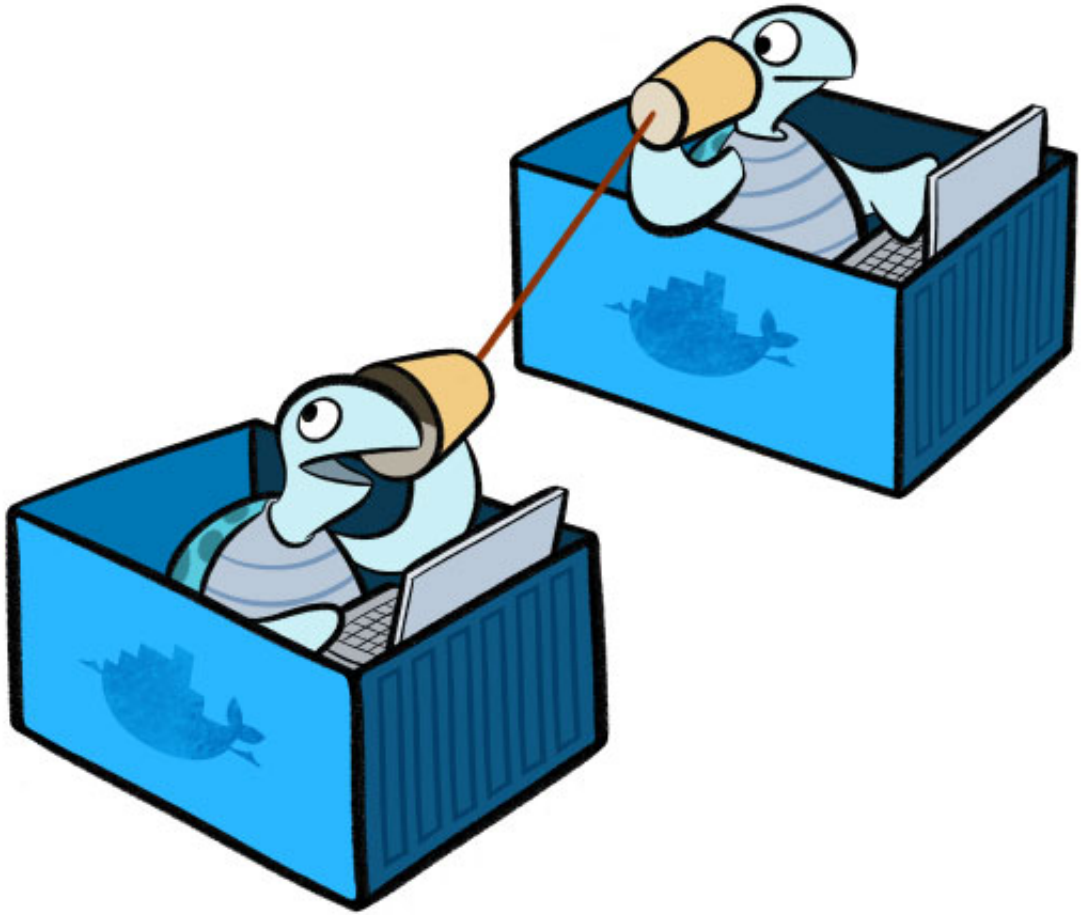
Multi-Host networking with Docker

OpenContainer Korea 2015
5 September 2015

anarcher

Multi-Host networking with docker



Demo

- It's only available with the experimental build of docker
- It was unveiled with docker engine 1.7 in dockercon 2015
- Release with docker engine 1.9 release?

Docker network UI

```
Usage: docker network [OPTIONS] COMMAND [OPTIONS] [arg...]

Commands:
    create                  Create a network
    rm                      Remove a network
    ls                      List all networks
    info                    Display information of a network

Run 'docker network COMMAND --help' for more information on a command.

  --help=false       Print usage
```

Create networks,removing or listing

```
docker network create -d <plugin_name> foo

$ docker network create foo
aae601f43744bc1f57c515a16c8c7c4989a2cad577978a32e6910b799a6bccf6
$ docker network create -d overlay bar
d9989793e2f5fe400a58ef77f706d03f668219688ee989ea68ea78b990fa2406
```

Docker service UI

```
Usage: docker service [OPTIONS] COMMAND [OPTIONS] [arg...]

Commands:
    publish   Publish a service
    unpublish Remove a service
    attach    Attach a backend (container) to the service
    detach    Detach the backend from the service
    ls        Lists all services
    info      Display information about a service

Run 'docker service COMMAND --help' for more information on a command.
```

When `docker run`, Can use `--publish-service` option

```
docker run -itd --publish-service db.foo postgres
```

Or can use `docker service publish` and `docker service attach` for linking a container with a service.

```
$ docker service publish my-service.foo
ec56fd74717d00f968c26675c9a77707e49ae64b8e54832ebf78888eb116e428
$ docker service attach a0ebc12d3e48 my-service.foo
```

Docker daemon & libkv

- docker-machine create

```
docker-machine --debug create \
    -d virtualbox \
    --virtualbox-boot2docker-url=https://github.com/anarcher/boot2docker-experimental/releases/do
    --engine-opt="kv-store=consul:$(docker-machine ip infra):8500" \
    --engine-label="com.docker.network.driver.overlay.bind_interface=eth1" \
    --engine-label="com.docker.network.driver.overlay.neighbor_ip=$(docker-machine ip demo0)" dem
```

- docker daemon

```
docker daemon
    --kv-store=consul:consul-host:8500 \
    --label=com.docker.network.driver.overlay.bind_interface=eth1 \
    --label=com.docker.network.driver.overlay.neighbor_ip=10.254.101.21
```

- Serf(serfdom.io (http://serfdom.io)) is used to cluster membership
- docker/libkv(--kv-store) is used to the metadata for docker network,docker service

Overlay network driver

- Uses VXLAN(Virtual eXtensible Local Area Network)

```
- Support from Linux(3.7)
- Configure (Logical) L2 Network on L3 Networks
- Packet encapsulation via IP/UDP
```

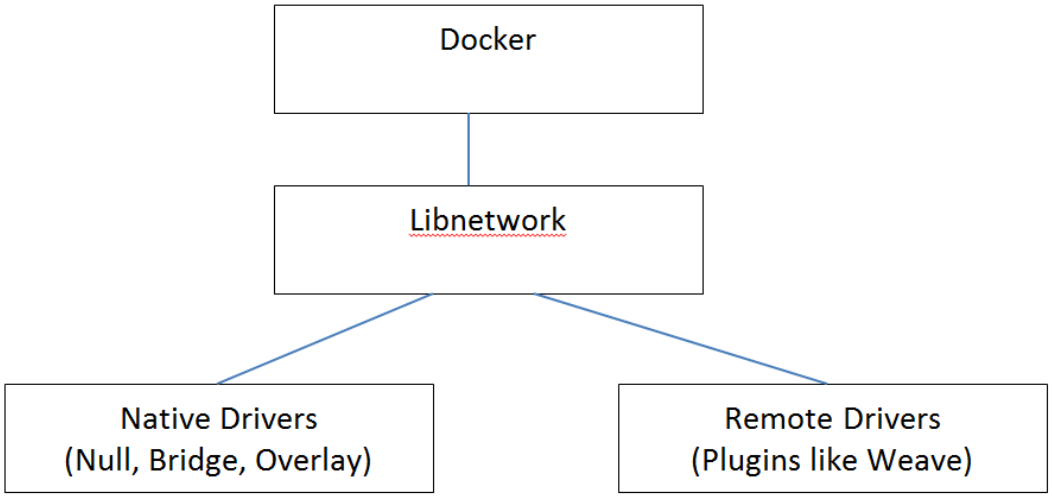
- Uses serf(serfdom.io (http://serfdom.io)) for searching other network node(docker engine)

- Uses Linux netns(network namespace)

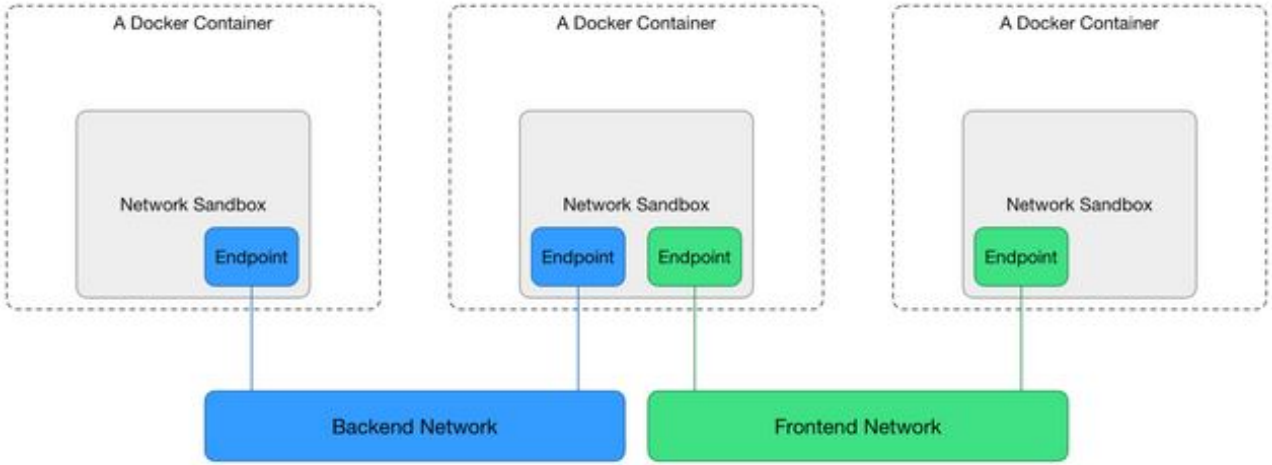
```
- Each namespaces are possible to have an indepenent network environment
- /var/run/docker/netns/
```

docker/libnetwork

- Separates the network part from Docker engine (Docker engine is also one of libnetwork users too)
- An Implementation of CMD(Container Network Model)
- docker/libnetwork has also implementations absout host,bridge which are docker engine's default network features)
- Driver based networking (These are two type drivers. builtin driver and remote drvier)


Docker
Libnetwork
Native Drivers
(Null, Bridge, Overlay)
Remote Drivers
(Plugins like Weave)

CNM : Container Network Model



- Sandbox : Isolation environment for containers' networks

```
- IP,MAC,Route,DNS ; Linux netns,FreeBSD Jail
```

- Endpoint : A network interface to communicate in a specific network

```
- The endpoint’s lifetime is completely separate from the container’s lifetime
```

- Network : A network is a collection of endpoints that are allowed to communicate with each other.

- An Endpoint can belong to only one network

Remote network driver plugin

- JSON/RPC/HTTP
- Plugin discovery

```
- /etc/docker/plugin/[name].spec|.sock|.json
```

- API

```
- /Plugin.Activate
- /NetworkDriver.CreateNetwork
- /NetworkDriver.DeleteNetwork
- /NetworkDriver.CreateEndpoint
- /NetworkDriver.EndpointOperInfo
- /NetworkDriver.DeleteEndpoint
- /NetworkDriver.Join
- /NetworkDriver.Leave
```

Thank you

anarcher
anarcher@gmail.com (mailto:anarcher@gmail.com)