Docker provides multiple implementations of the Container Networking Model in the form of network drivers. In this lesson, we will explore the native network drivers that come packaged with Docker by default. We will provide a high-level overview of how these drivers work and discuss the use cases for each. We will also demonstrate how to run containers that use each driver.

Relevant Documentation

- https://success.docker.com/article/networking/
- https://blog.docker.com/2016/12/understanding-docker-networking-drivers-use-cases/

Lesson Reference

Host

Create two containers and demonstrate communication between them using the host network driver. The `ip add` commands also demonstrate that the container is using the host's `eth0` network interface directly.

```
docker run -d --net host --name host_busybox radial/busyboxplus:curl sleep 3600
docker run -d --net host --name host_nginx nginx
ip add | grep eth0
docker exec host_busybox ip add | grep eth0
docker exec host_busybox curl localhost:80
curl localhost:80
```

Bridge

Create two containers and demonstrate that they can communicate using a custom bridge network. The `ip link` command can be used to explore the Linux bridge interfaces created by the bridge network driver.

```
ip link
docker network create --driver bridge my-bridge-net
ip link
docker run -d --name bridge_nginx --network my-bridge-net nginx
docker run --rm --name bridge_busybox --network my-bridge-net radial/busyboxplus:curl curl bridge_nginx:80
```

Overlay

Create two services in the Docker Swarm cluster and demonstrate that they are able to communicate using a custom overlay network.

```
docker network create --driver overlay my-overlay-net
docker service create --name overlay_nginx --network my-overlay-net nginx
docker service create --name overlay_busybox --network my-overlay-net radial/busyboxplus:curl sh -c 'curl ove
docker service logs overlay_busybox
```

MACVLAN

Create a MACVLAN network. Then run two containers that are able to communicate using the MACVLAN network.

```
docker network create -d macvlan --subnet 192.168.0.0/24 --gateway 192.168.0.1 -o parent=eth0 my-macvlan-net
docker run -d --name macvlan_nginx --net my-macvlan-net nginx
docker run --rm --name macvlan_busybox --net my-macvlan-net radial/busyboxplus:curl curl 192.168.0.2:80
```

None

Create a container that uses the `none` network and demonstrate that a normal container cannot reach it.

```
docker run --net none -d --name none_nginx nginx
docker run --rm radial/busyboxplus:curl curl none_nginx:80
```