

**Docker Certified Associate 2020**

**Networking**

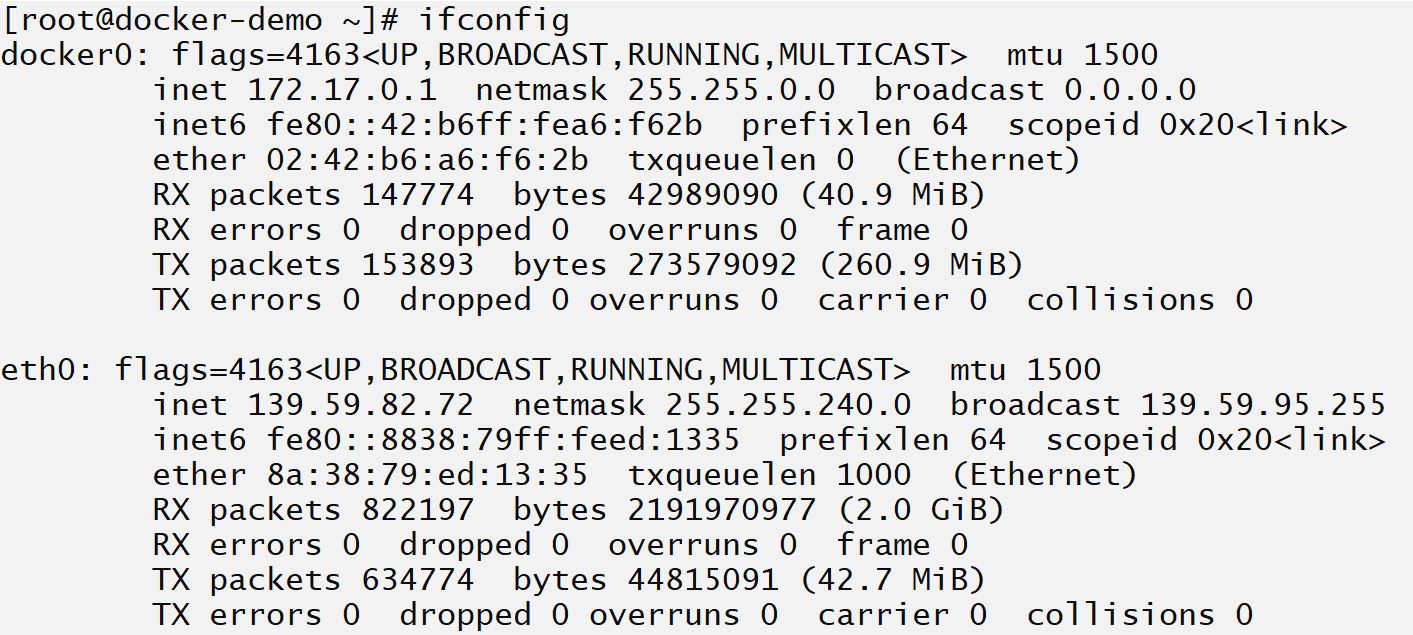
## **ISSUED BY**

Abhishek Muduli

# 

## **Module 1: Overview of Docker Networking**

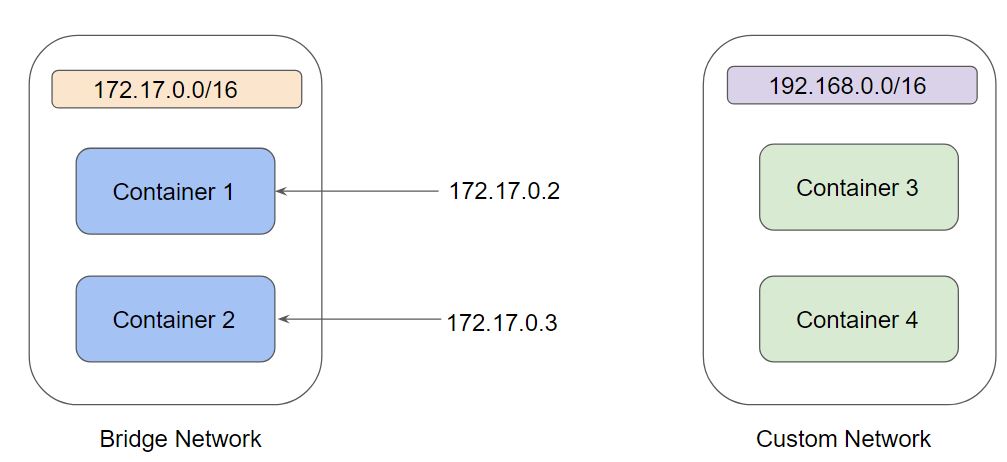
Docker takes care of the networking aspects so that containers can communicate with other containers and also with the Docker Host.



Docker networking subsystem is pluggable, using drivers.

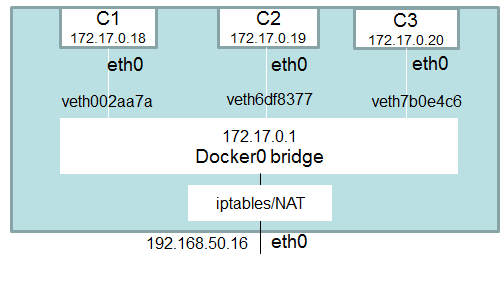
There are several drivers available by default and provides core networking functionality.

* bridge
* host
* overlay
* macvlan
* none



## **Module 2: Bridge Networks**

A bridge network uses a software bridge that allows containers connected to the same bridge network to communicate while providing isolation from containers which are not connected to that bridge network.



Bridge is the default network driver for Docker.

If we do not specify a driver, this is the type of network you are creating.

When you start Docker, a default bridge network (also called a bridge) is created automatically, and newly-started containers connect to it unless otherwise specified.

We also can create a User-Defined Bridge Network which is superior to the default bridge.

## **Module 3: Host Network**

This driver removes the network isolation between the docker host and the docker containers to use the host’s networking directly.

For instance, if you run a container that binds to port 80 and you use host networking, the container’s application will be available on port 80 on the host’s IP address.

## **Module 4: None Network**

If you want to completely disable the networking stack on a container, you can use the none network.

This mode will not configure any IP for the container and doesn’t have any access to the external network as well as for other containers.

## **Module 5: Publishing Exposed Ports of Container**

We were discussing an approach to publishing container port to host.

docker container run -dt --name webserver -p 80:80 nginx

This is also referred to as a publish list as it publishes the only a list of the port specified.

There is also a second approach to publish all the exposed ports of the container.

docker container run -dt --name webserver -P nginx

This is also referred to as a publish all.

In this approach, all exposed ports are published to random ports of the host.