BATCH : BATCH 48

LESSON : **Docker**

DATE : 11.04.2022

SUBJECT : **Docker Networking & Images**

techproeducation

techproeducation

techproeducation

techproeducation

techproedu

techproeducation.com info@techproeducation.com +1 (917) 768-7466

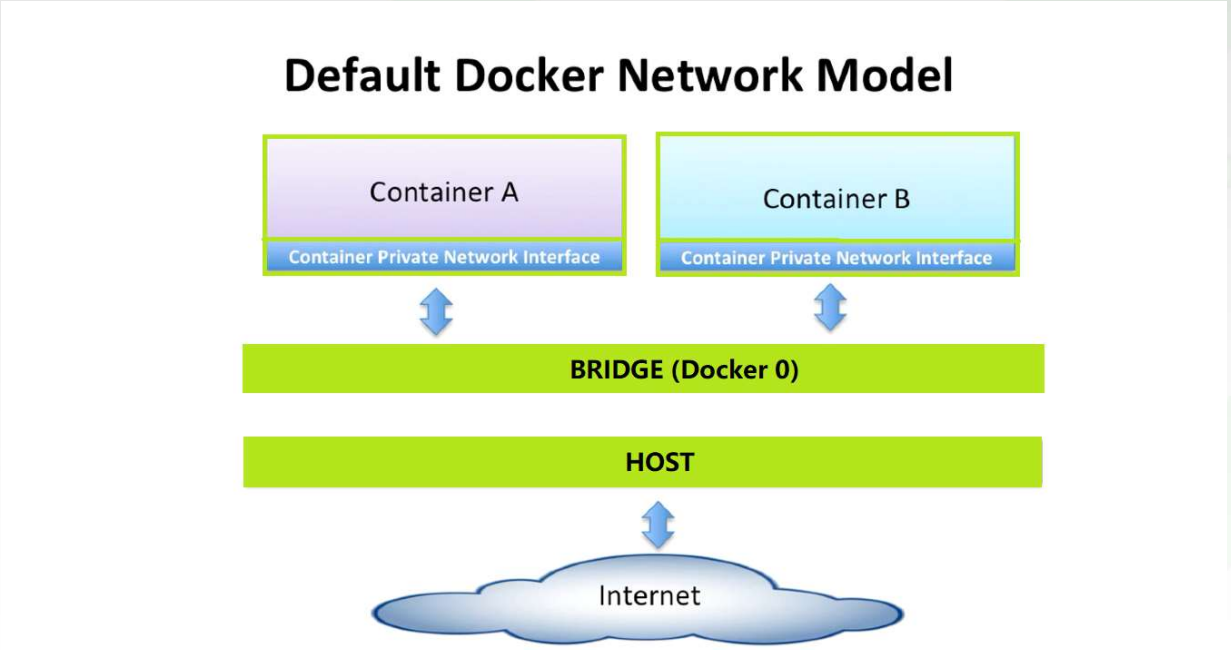
Docker Networking

Table of Contents

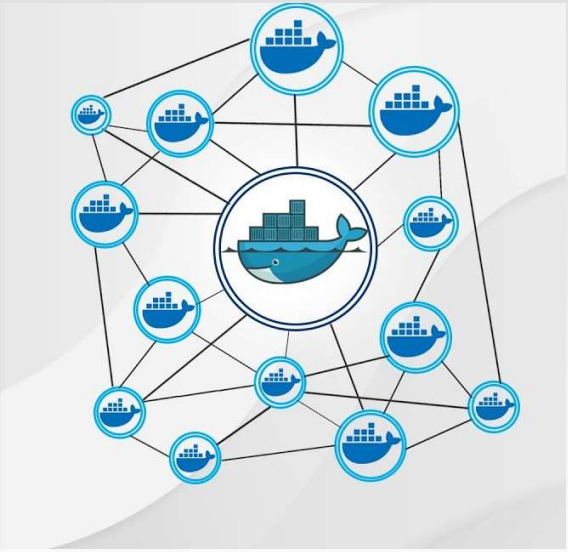
Networking Overview

Networking Overview

A **network** is two or more computer systems linked together by some form of the transmission medium.



Networking Overview



- One of the reasons Docker containers and services are so powerful is that you can connect them together, or connect them to non-Docker workloads.
- Whether your Docker hosts run linux, Windows, or a mix of the two, you can use Docker to manage them in a platform-agnostic way.

Network Drivers

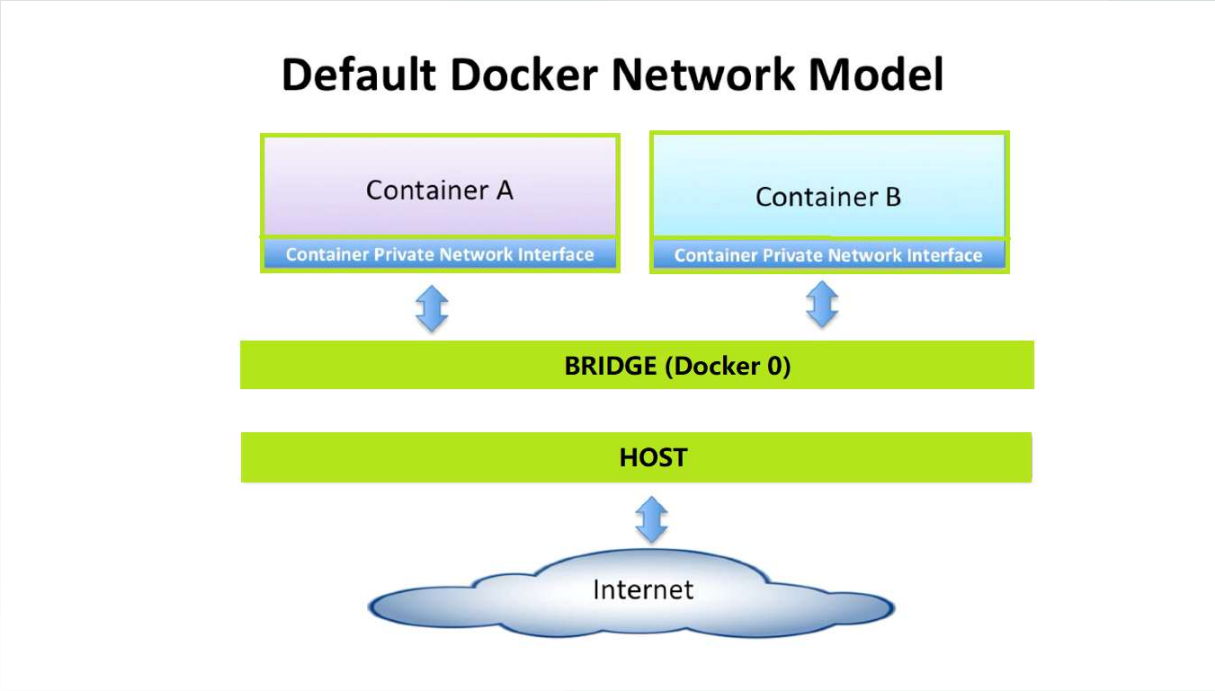
Network Drivers

As default, docker has three network drivers.

- Bridge
- Host
- None

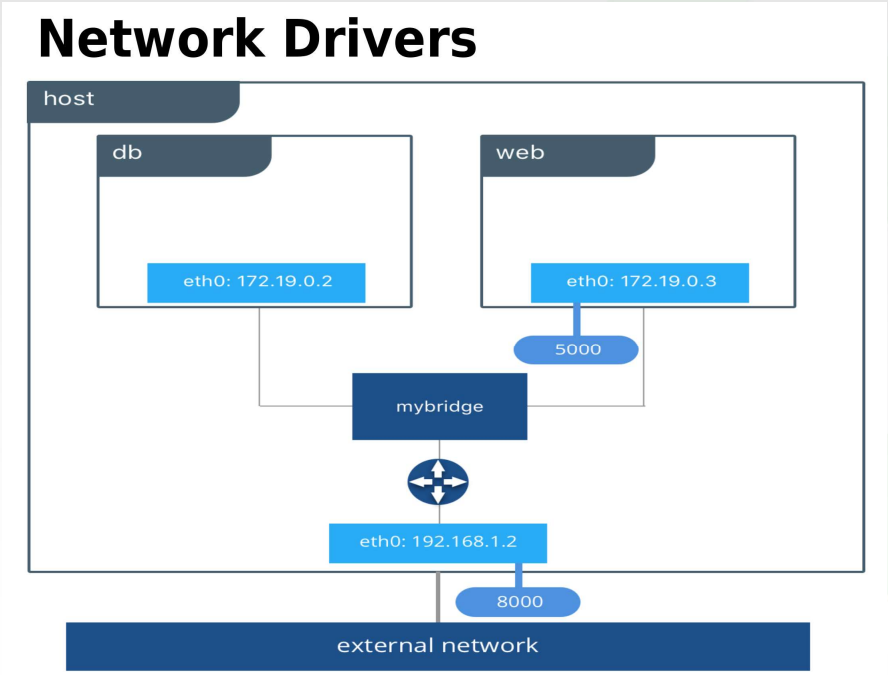
Network Drivers

- **Bridge** is the private default network driver. If we don't specify a driver, this is the type of network we are creating.



Network Drivers

- When we create containers,it will automatically attach to the bridge driver.



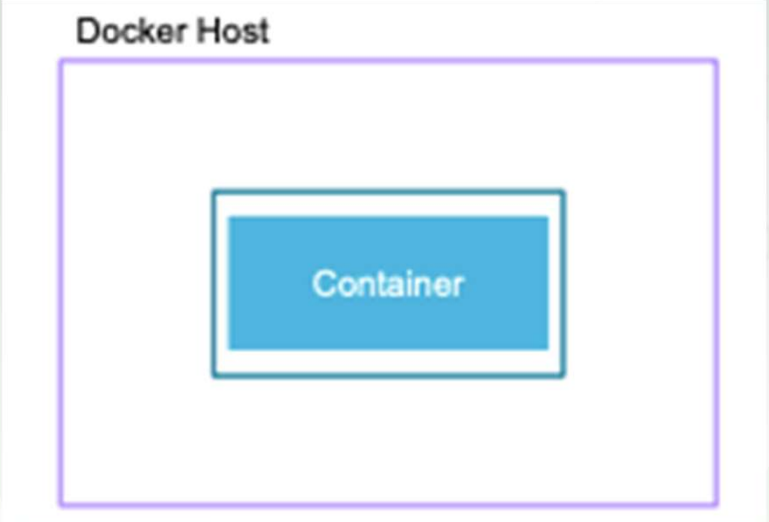
Network Drivers

- **Host** removes network isolation between the docker host and docker containers. It uses the host's networking directly.

- Host networks are best when the network stack should not be isolated from the Docker host, but we want other aspects of the container to be isolated.

Network Drivers

- **None** network driver disable all networking of containers.
- **None** network driver will not configure any IP for the container and doesn't have any access to the extemal network as wel as for other containers.
- It is used when a user wants to block the networking access to a container.

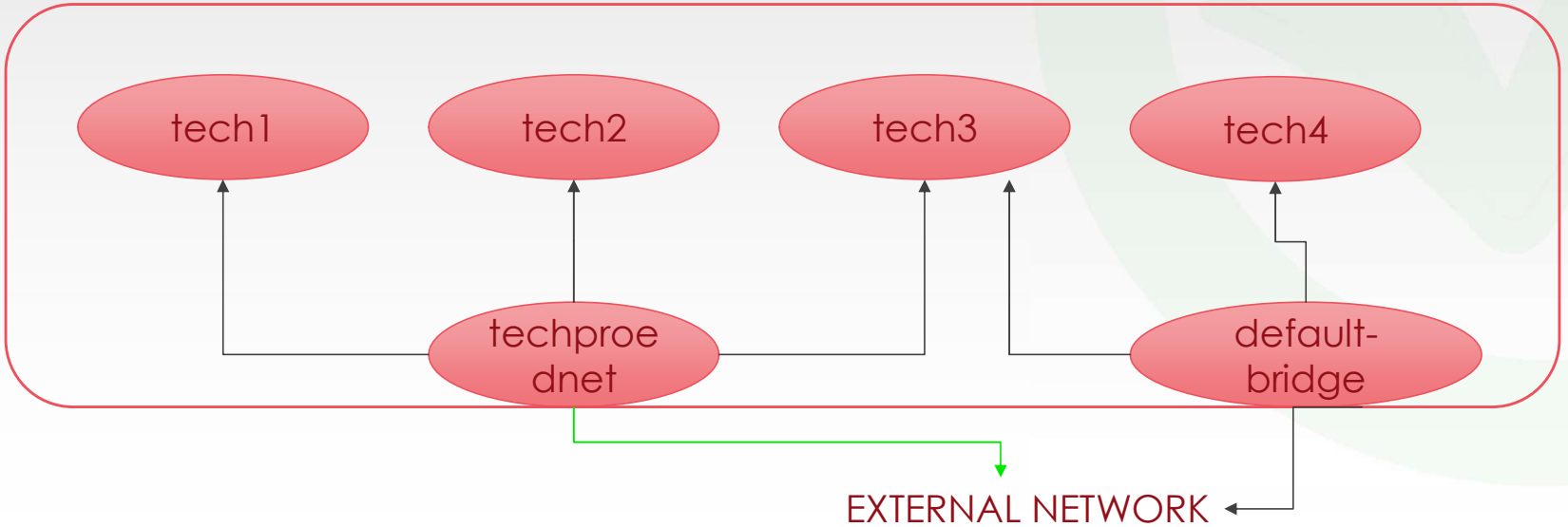


User-defined bridge networks

User-defined Bridge Networks

- In addition to the default networks, users can create their own networks called user-defined networks of any network driver type.

$ docker network create--driver bridge techproednet



Run – Port Mappings

Run – Port Mappings

- By default, when you create a container, it does not publish any of its ports to the outside world. To make a port available to services outside of Docker, or to Docker containers which are not connected to the container's network, use the **--publish** or **-p flag.**

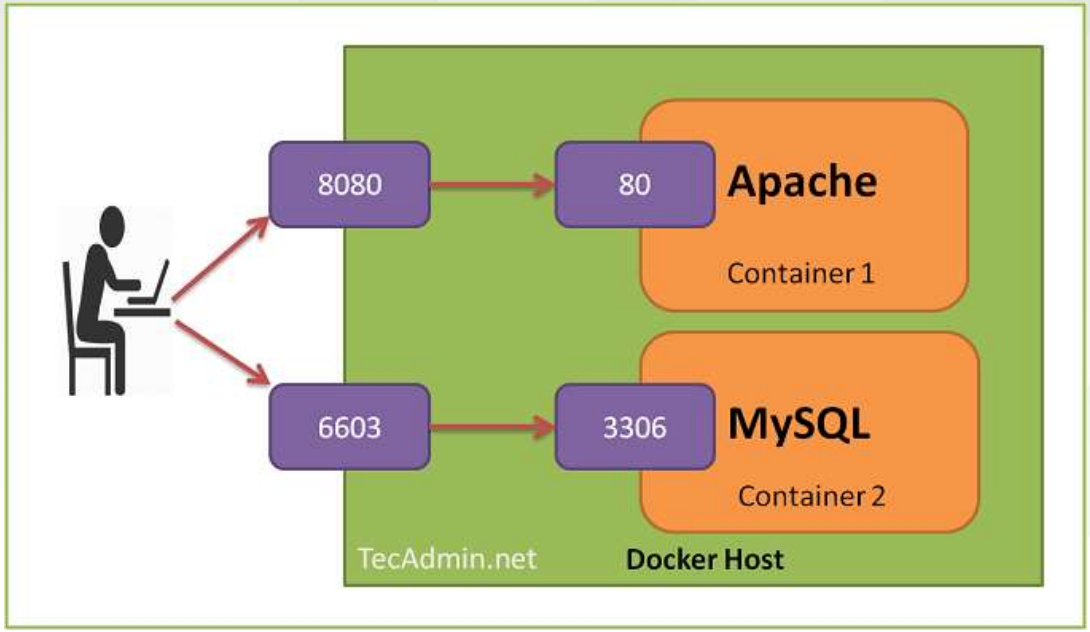
```
-p host_port : container_port
```

-P -> Random Ports

Run – Port Mappings

$ docker run-d-p 8080:80 apache_image

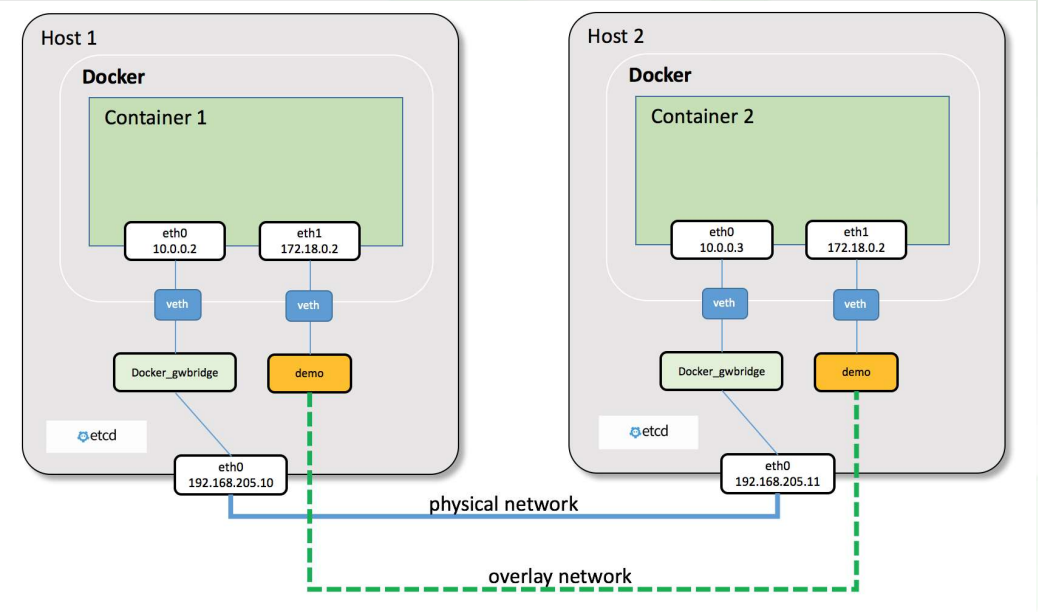
$ docker run d -p 6603:3306 mysql_image



Other Network Drivers

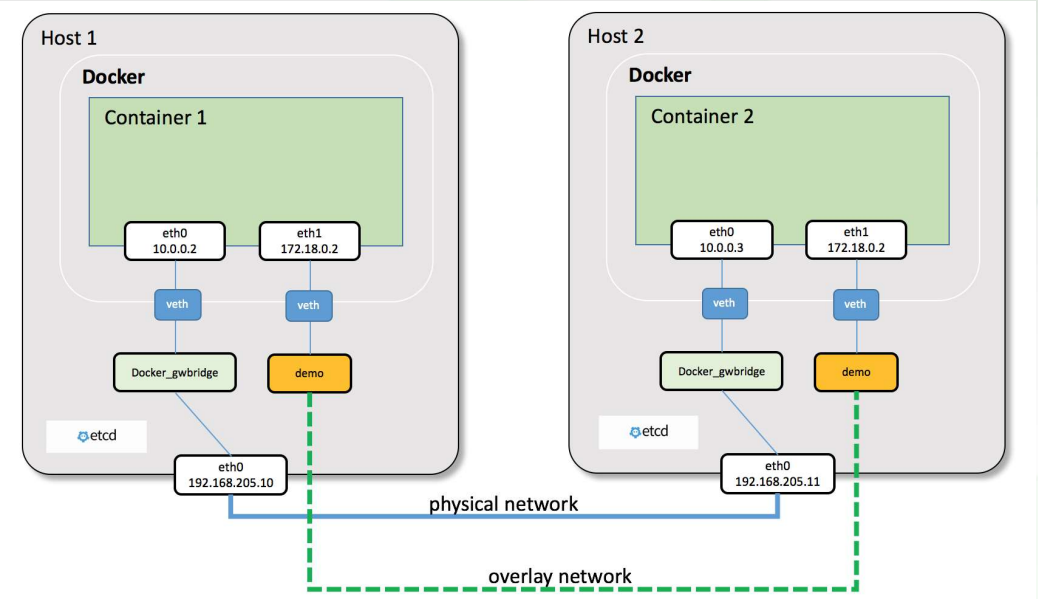
Network Drivers

- **The Overlay network driver** creates a distributed network among **multiple Docker deamon hosts.**



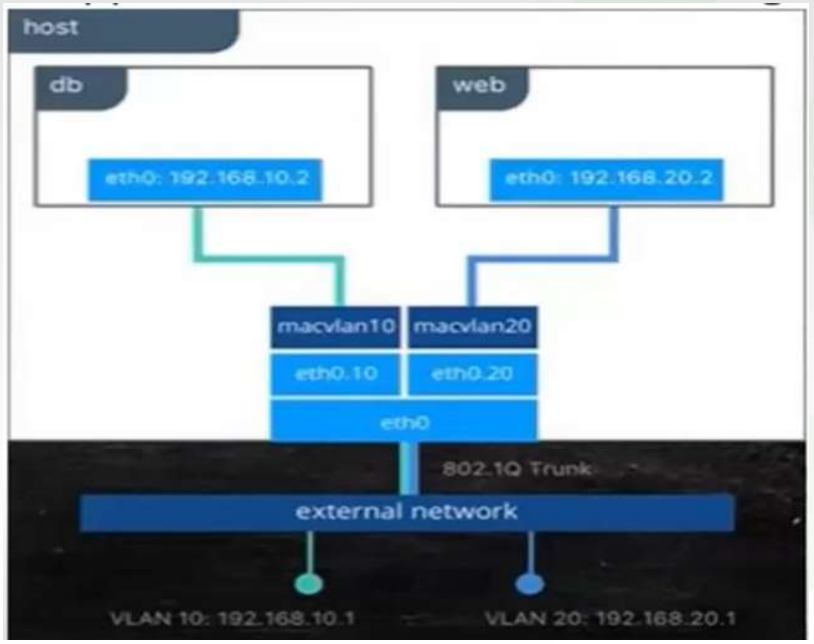
Network Drivers

- Overlay networks connect multiple Docker daemons together and enable swam services to communicate with each other.



Network Drivers

- **MacVlan network** driver supplies the containers networking as has a physical NIC.



Network Drivers

- Macvlan networks allow you to assign a MAC address to a container, making it appear as a physical device on your network.
- Using the macvlan driver is sometimes the best choice when dealing with legacy applications that expect to be directly connected to the physical network, rather than routed through the Docker host's network stack.

Docker Network Commands

Docker Network Commands

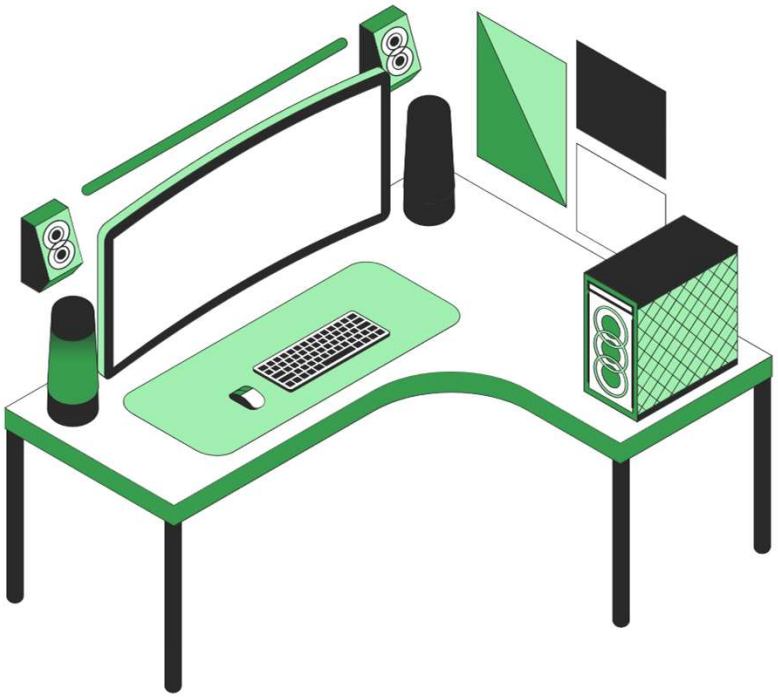
```
root@CPDockerTEST:/home/ubuntu# docker network

Usage:  docker network COMMAND

Manage networks

Commands:
  connect     Connect a container to a network
  create      Create a network
  disconnect  Disconnect a container from a network
  inspect     Display detailed information on one or more networks
  ls          List networks
  prune       Remove all unused networks
  rm          Remove one or more networks

Run 'docker network COMMAND --help' for more information on a command.
root@CPDockerTEST:/home/ubuntu#
```



Do you have any questions?

Send it to us! We hope you learned something new.

TECHPROED