

DATE : 27.03.2025

DT/NT : DT

LESSON : DOCKER

SUBJECT: NETWORK

BATCH : B 303

AWS-DEVOPS



TECHPRO
EDUCATION



techproeducation.com



+1 (585) 304 29 59





Table of Contents

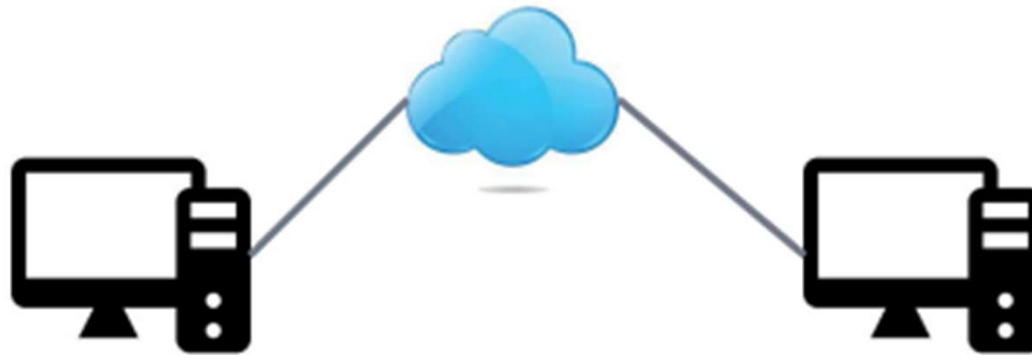
- ▶ Networking overview
- ▶ Network drivers
- ▶ User-defined bridge networks
- ▶ Run - Port mappings
- ▶ Other Network drivers
- ▶ docker network Commands



Networking overview

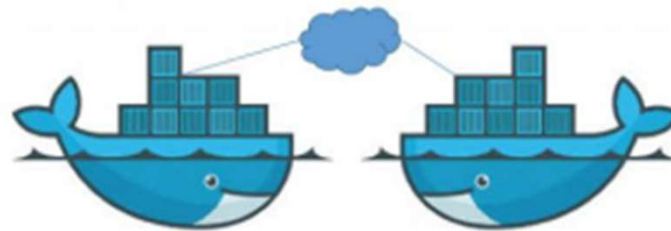
Networking overview

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.



Docker networks

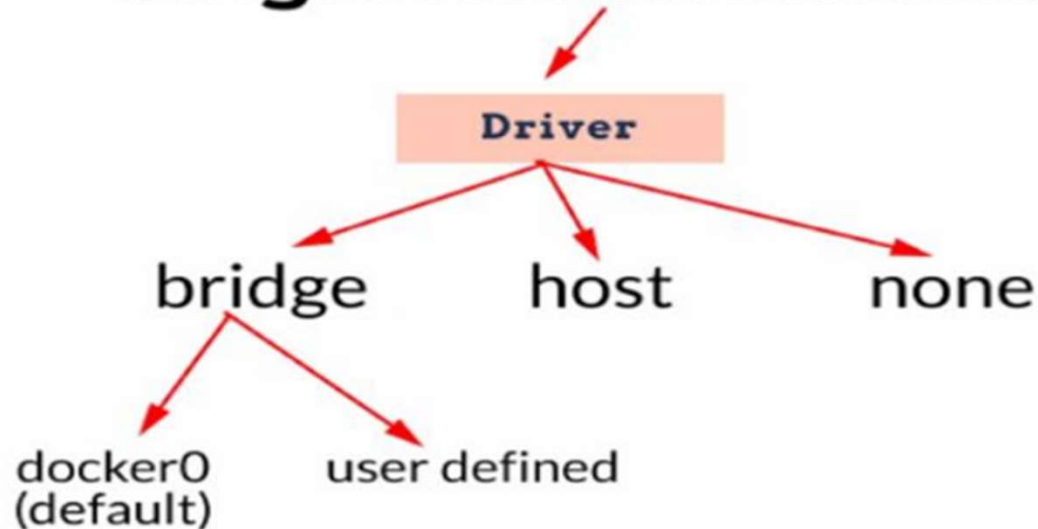


Network drivers

Network drivers

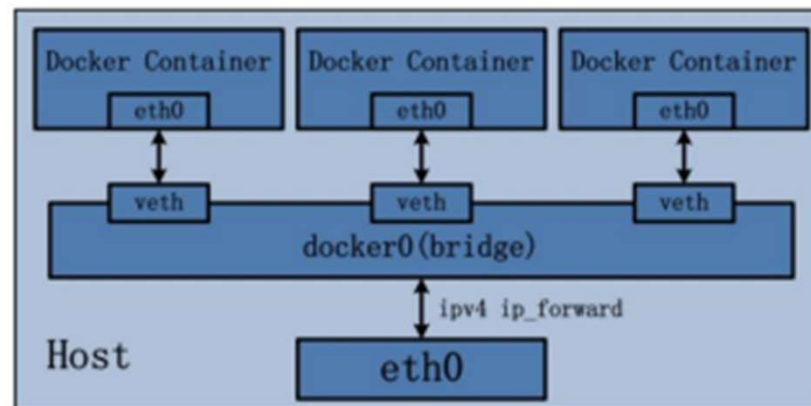
- As default, docker has three network drivers.

Single Host Networking



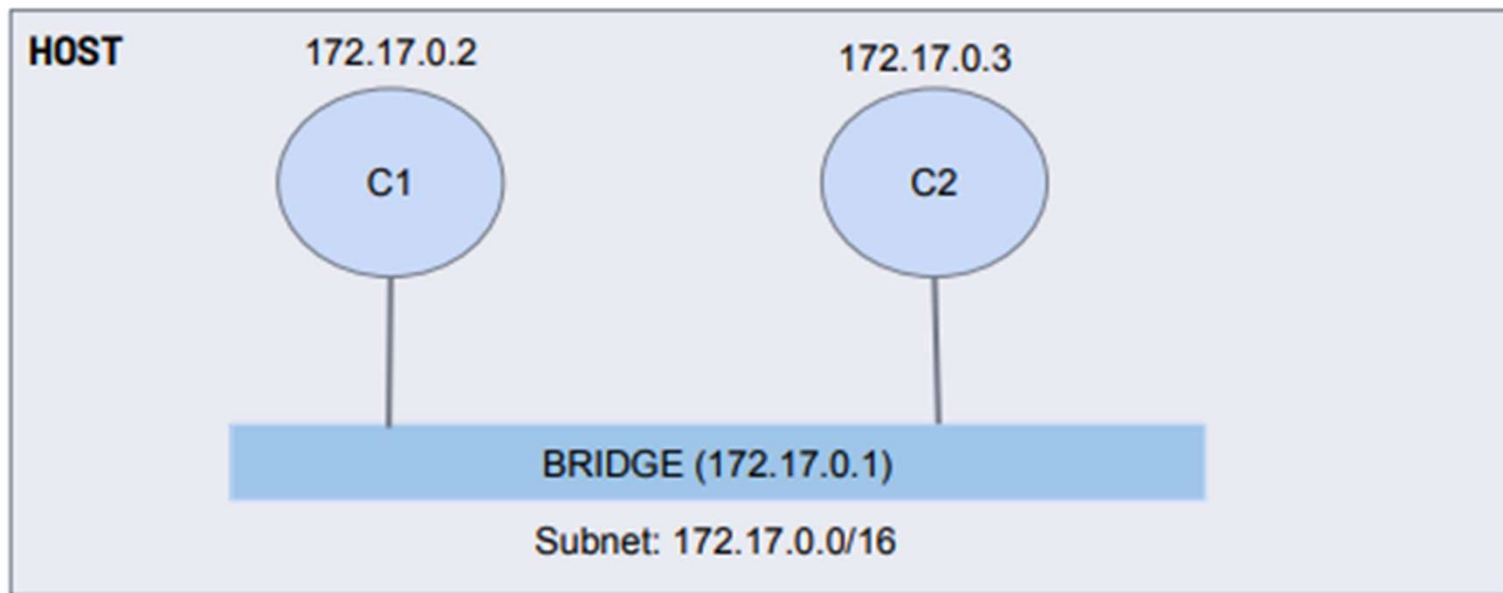
Network drivers

- bridge is the private default network driver. If we don't specify a driver, this is the type of network we are creating.
- When we install the docker, the Docker daemon creates virtual ethernet bridge docker0 that performs the operation by automatically delivering packets among various network interfaces.



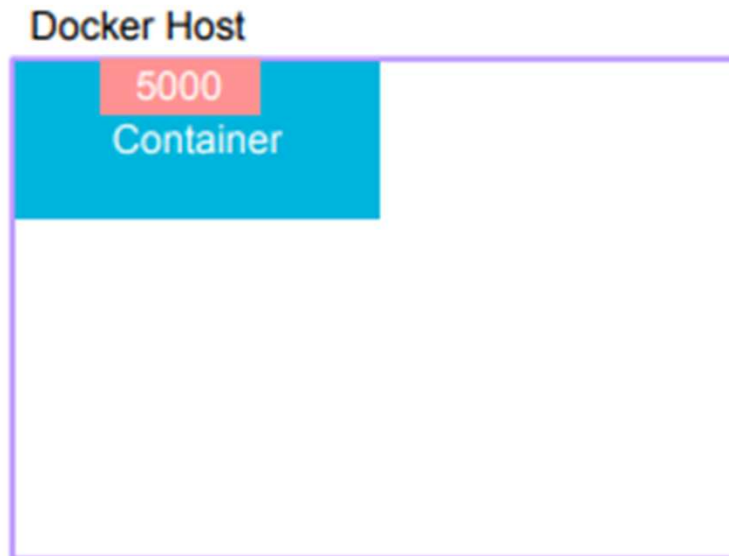
Network drivers

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



Network drivers

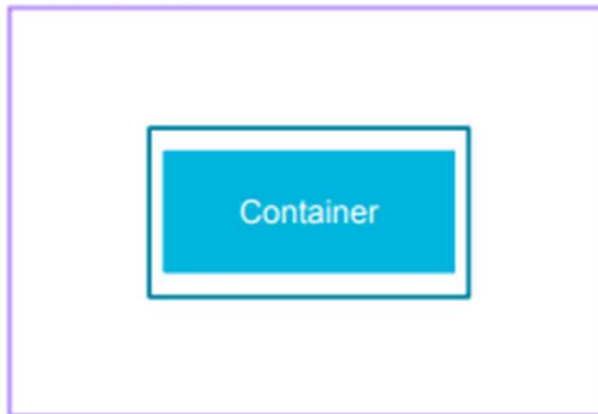
- Host removes network isolation between the docker host and docker containers. It uses the host's networking directly.
- Host networks are best when the network stack should not be isolated from the Docker host, but we want other aspects of the container to be isolated.



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 external network as well as for other containers.
- It is used when a user wants to block the networking access to a container.

Docker Host





User-defined bridge networks

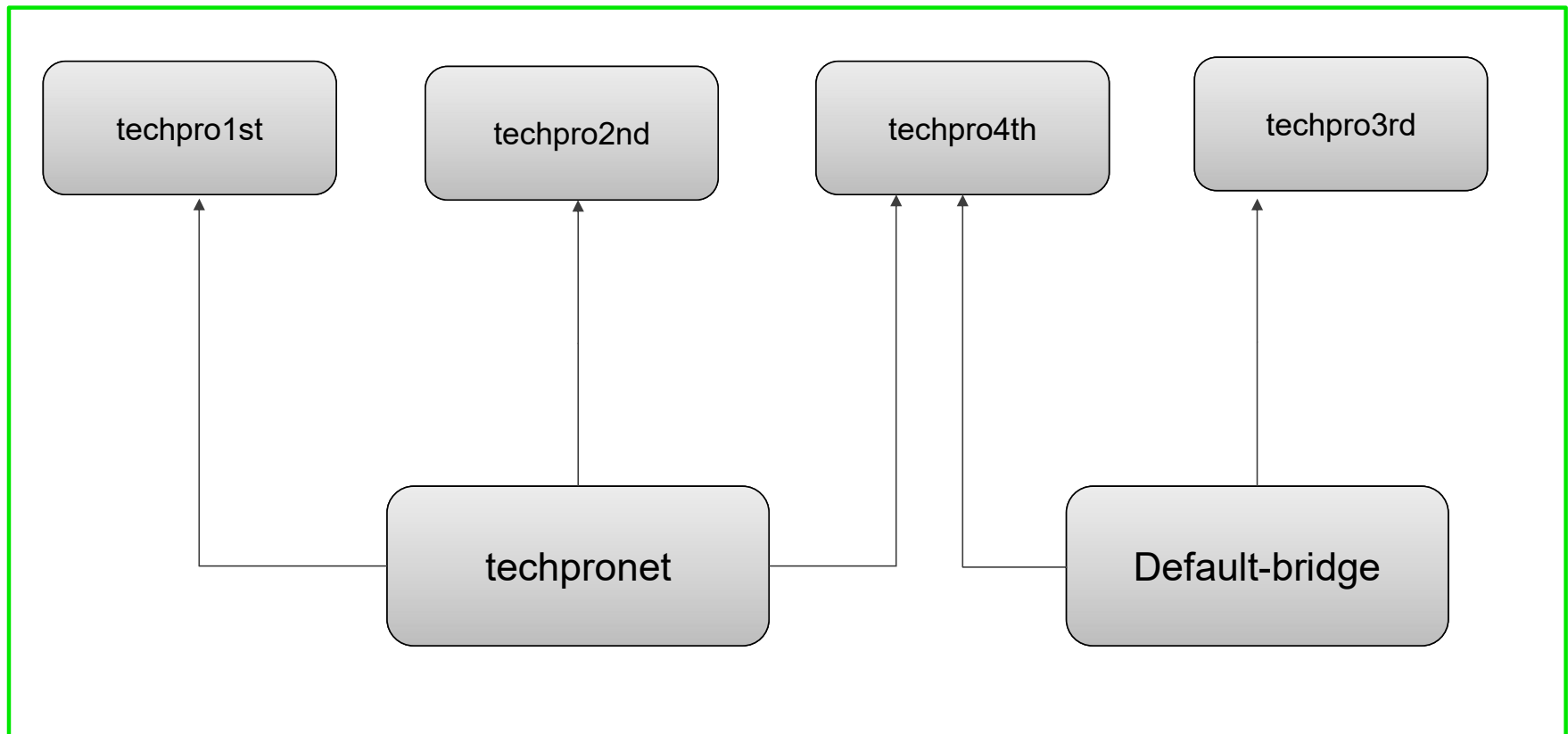


User-defined bridge networks

- 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 techpronet
```

User-defined bridge networks





Run - Port mappings



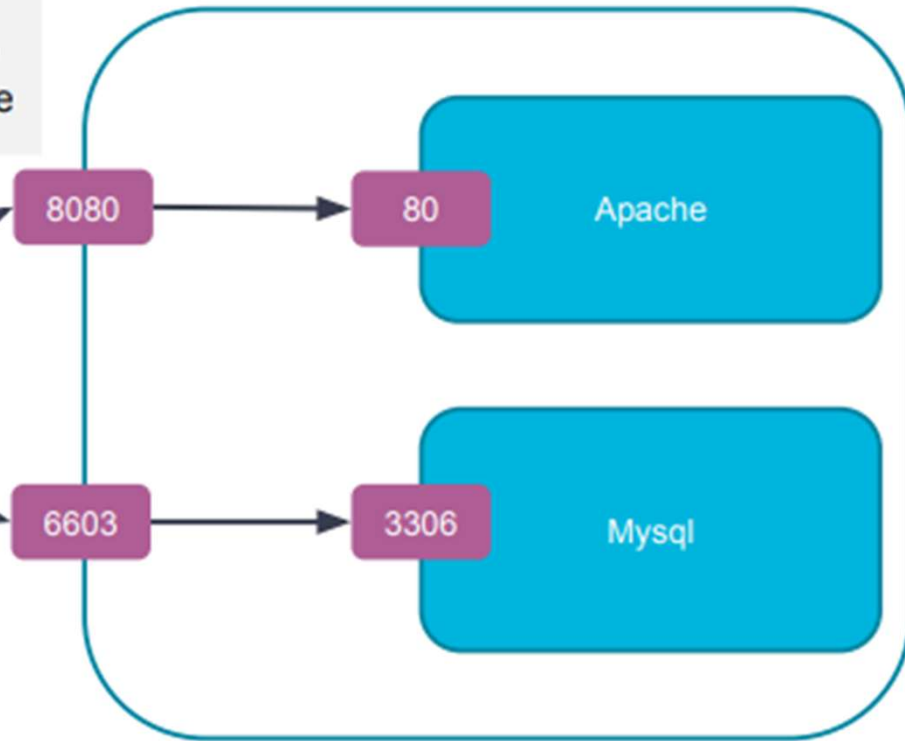
Run - Port mappings

- 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
```


Run - Port mappings

```
$ docker run -d -p 8080:80 apache_image  
$ docker run -d -p 6603:3306 mysql_image
```

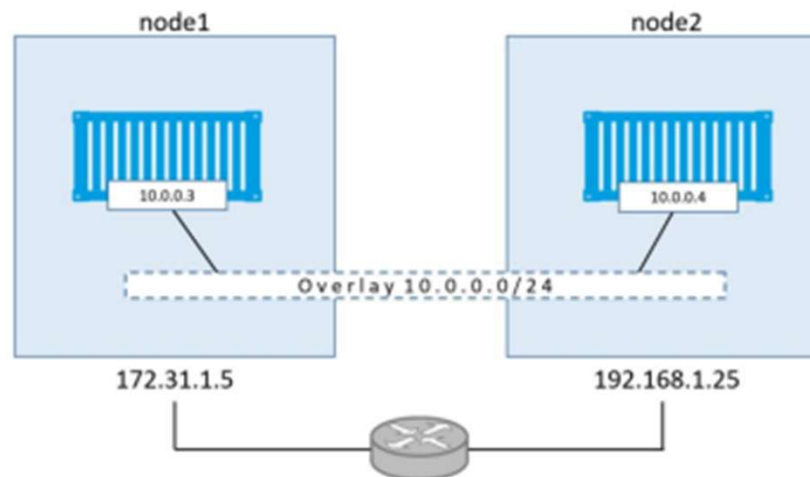




Other Network drivers

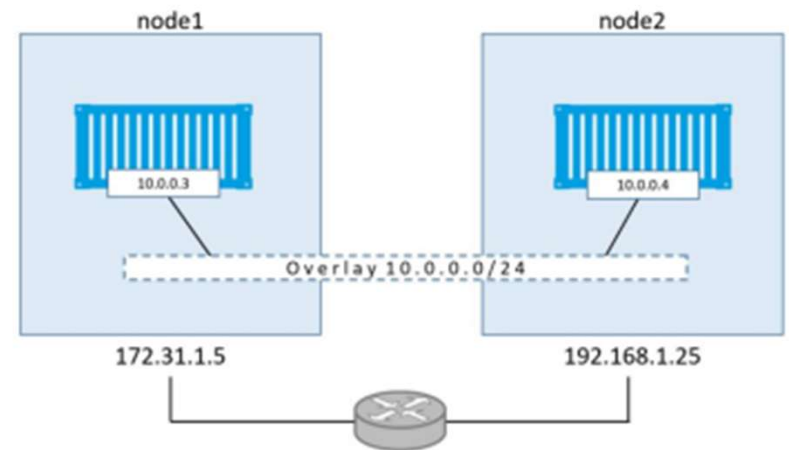
Network drivers

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



Network drivers

overlay: Overlay networks connect multiple Docker daemons together and enable swarm services to communicate with each other. You can also use overlay networks to facilitate communication between a swarm service and a standalone container, or between two standalone containers on different Docker daemons. This strategy removes the need to do OS-level routing between these containers





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.



Network drivers

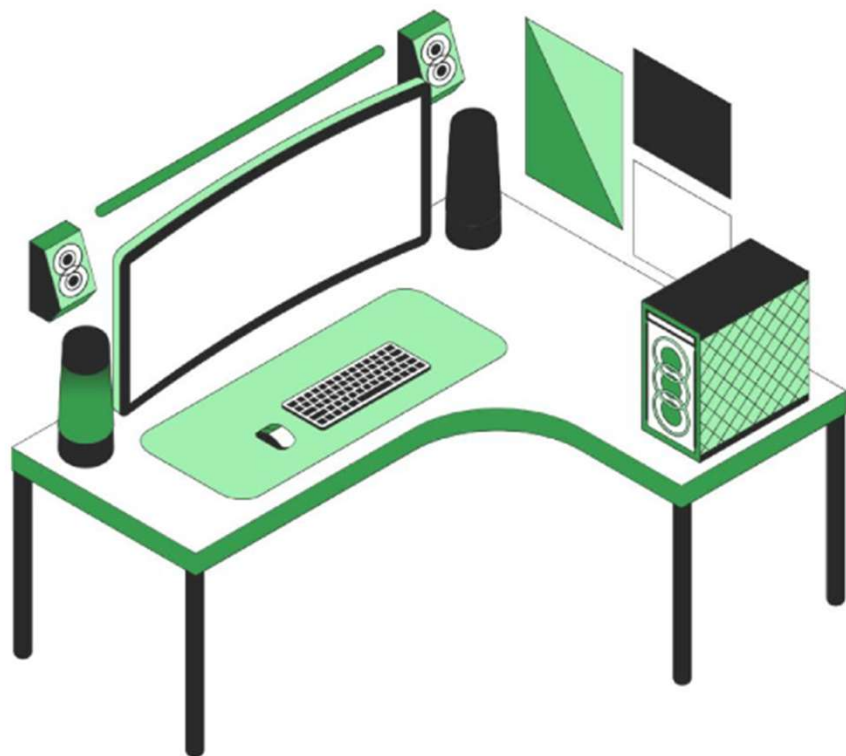
- **Network plugins:** We can install and use third-party network plugins with Docker. These plugins are available from Docker Hub or from third-party vendors.
- Third-party network plugins allow us to integrate Docker with specialized network stacks.



Docker Network Commands

Docker Network Commands

Command	Description
<u>docker network connect</u>	Connect a container to a network
<u>docker network create</u>	Create a network
<u>docker network disconnect</u>	Disconnect a container from a network
<u>docker network inspect</u>	Display detailed information on one or more networks
<u>docker network ls</u>	List networks
<u>docker network prune</u>	Remove all unused networks
<u>docker network rm</u>	Remove one or more networks



Do you have any questions?

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