

# Docker Networking



# Table of Contents



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





1

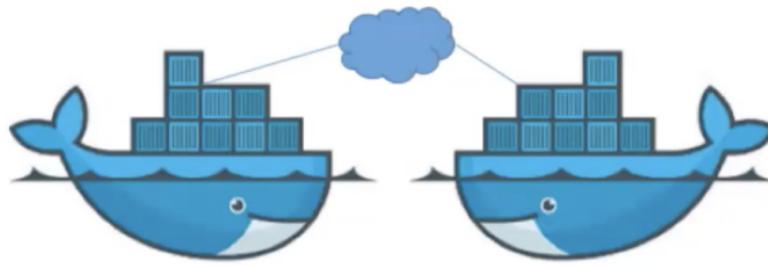
# Networking overview





# 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.
- Docker containers and services do not even need to be aware that they are deployed on Docker, or whether their peers are also Docker workloads or not.
- 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.



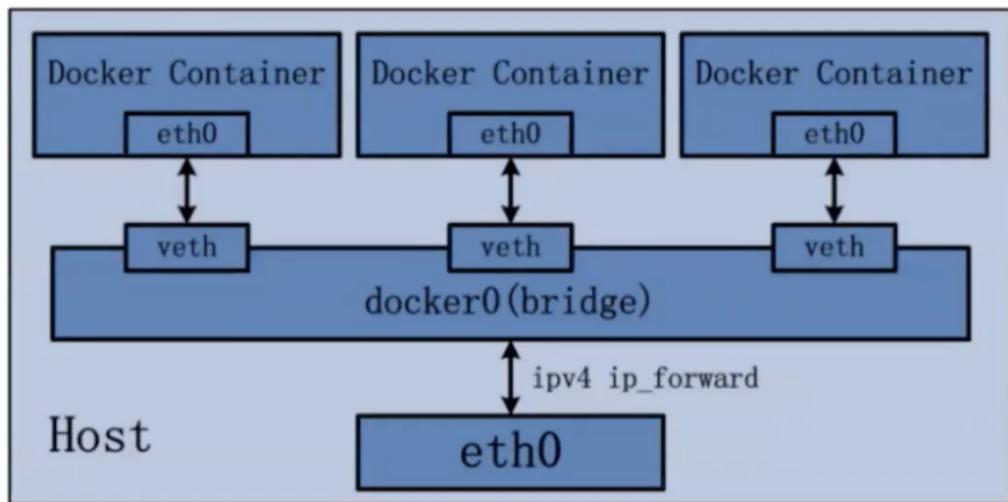
Docker networks





# Network drivers

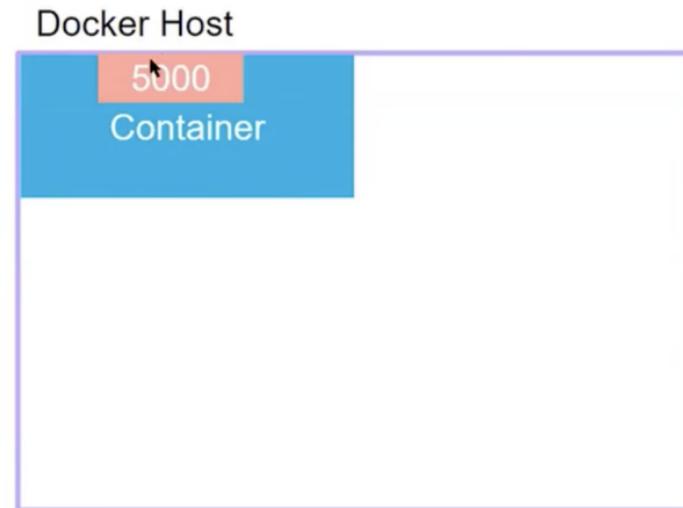
- **bridge:** The default network driver. If you don't specify a driver, this is the type of network you are creating. Bridge networks are usually used when your applications run in standalone containers that need to communicate.





# Network drivers

- **host:** For standalone containers, remove network isolation between the container and the Docker host, and use the host's networking directly.

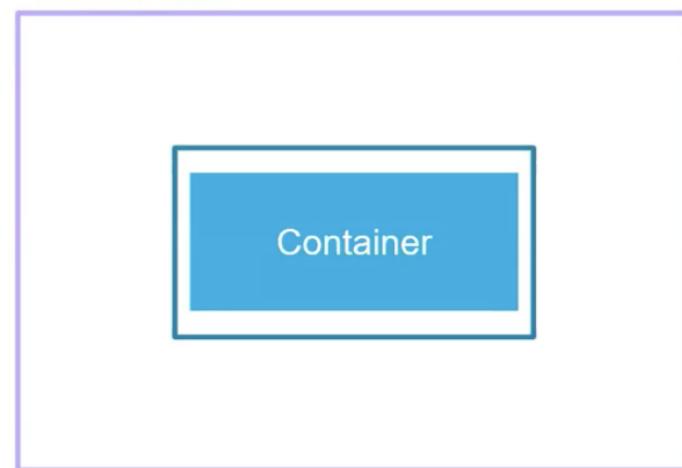


# ► Network drivers



- **none:** For this container, disable all networking. Usually used in conjunction with a custom network driver.

Docker Host



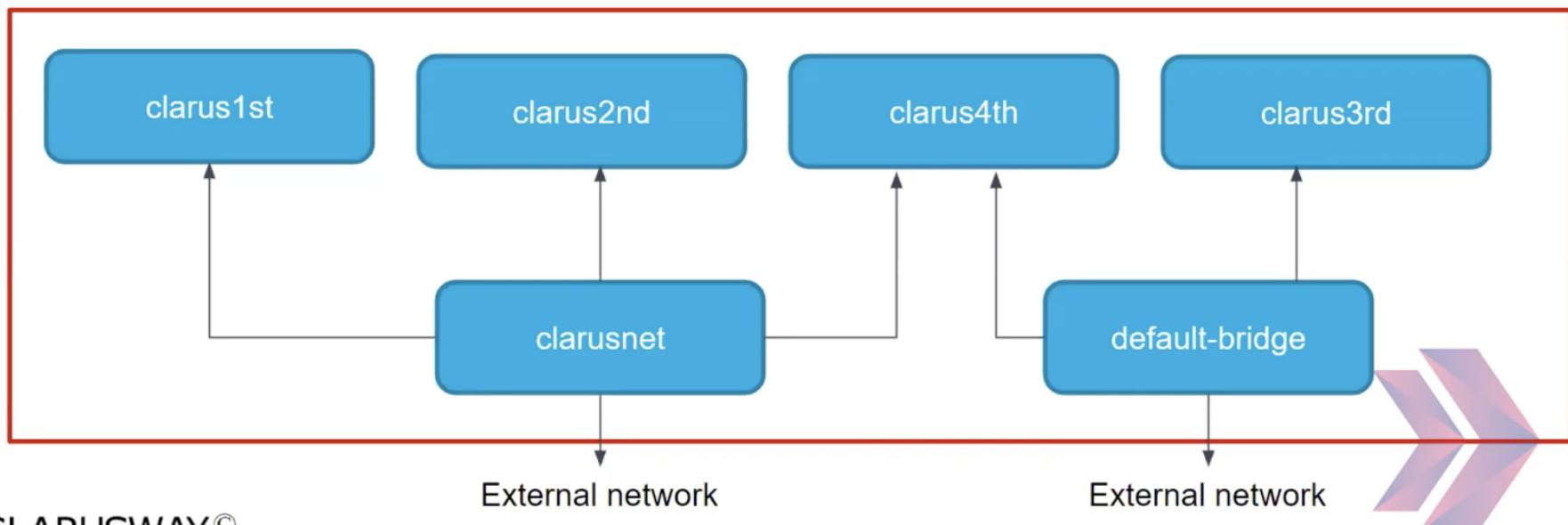
# User-defined bridge networks



Joe Blue-Instructor

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 clarusnet
```





# ► 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.

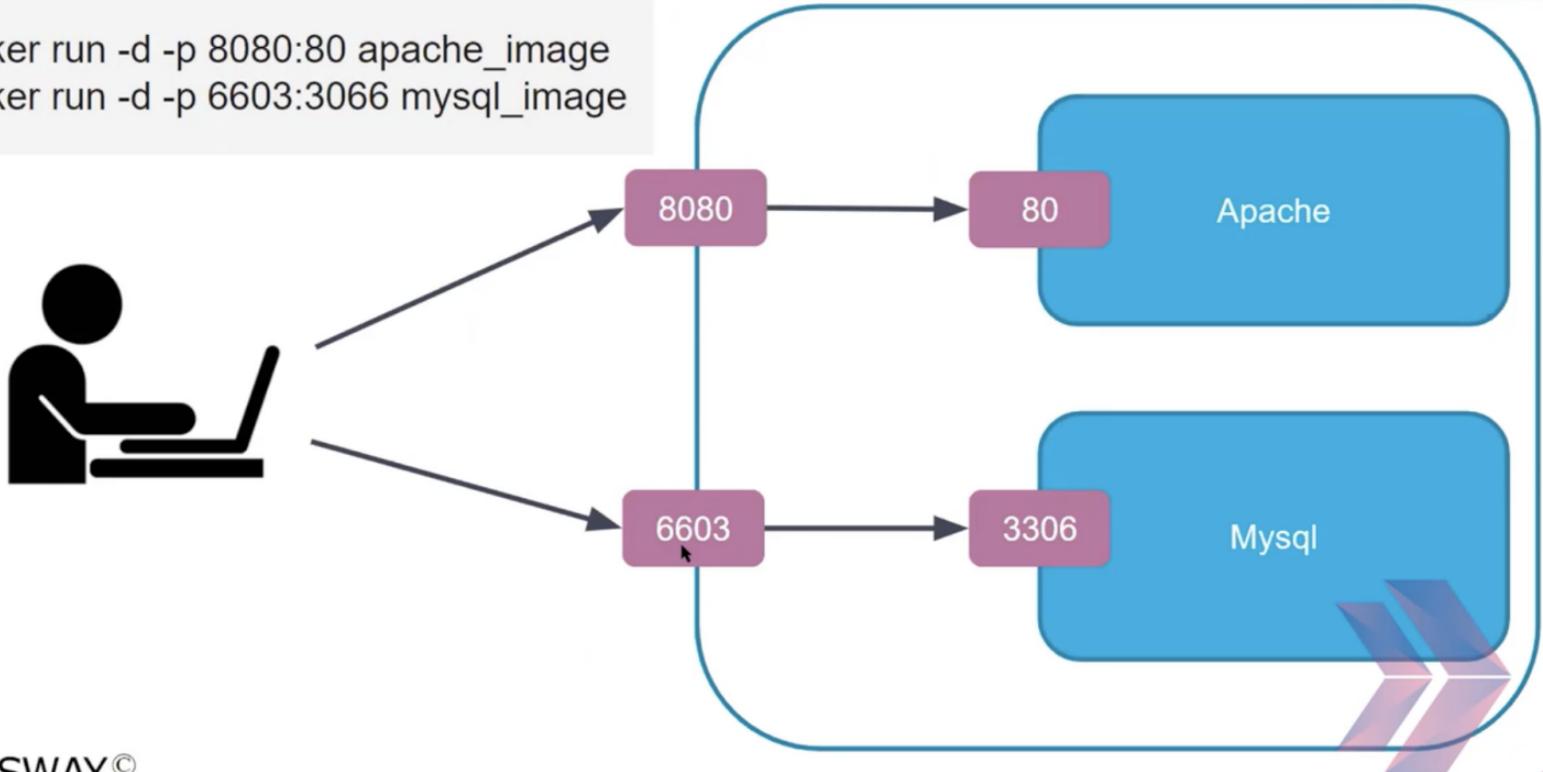
```
-p host_port : container_port
```





# ► Run - Port mappings

```
$ docker run -d -p 8080:80 apache_image  
$ docker run -d -p 6603:3066 mysql_image
```





# ► docker network Commands

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

