

Introduction à Docker



docker®

Planning

“A goal without a plan is just a wish.” - Me



Run a container with a simple Dockerfile



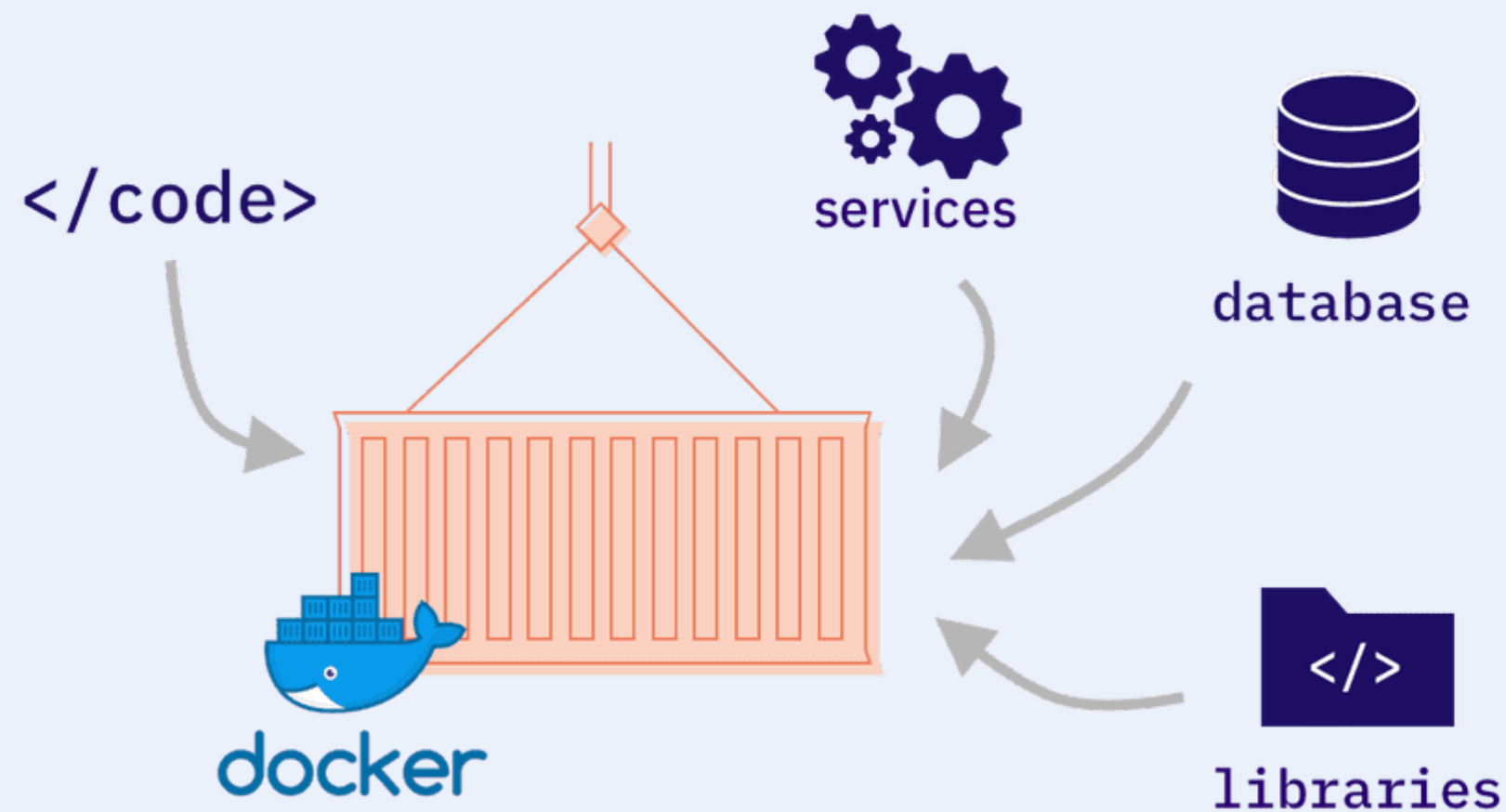
Deal with communication between
containers



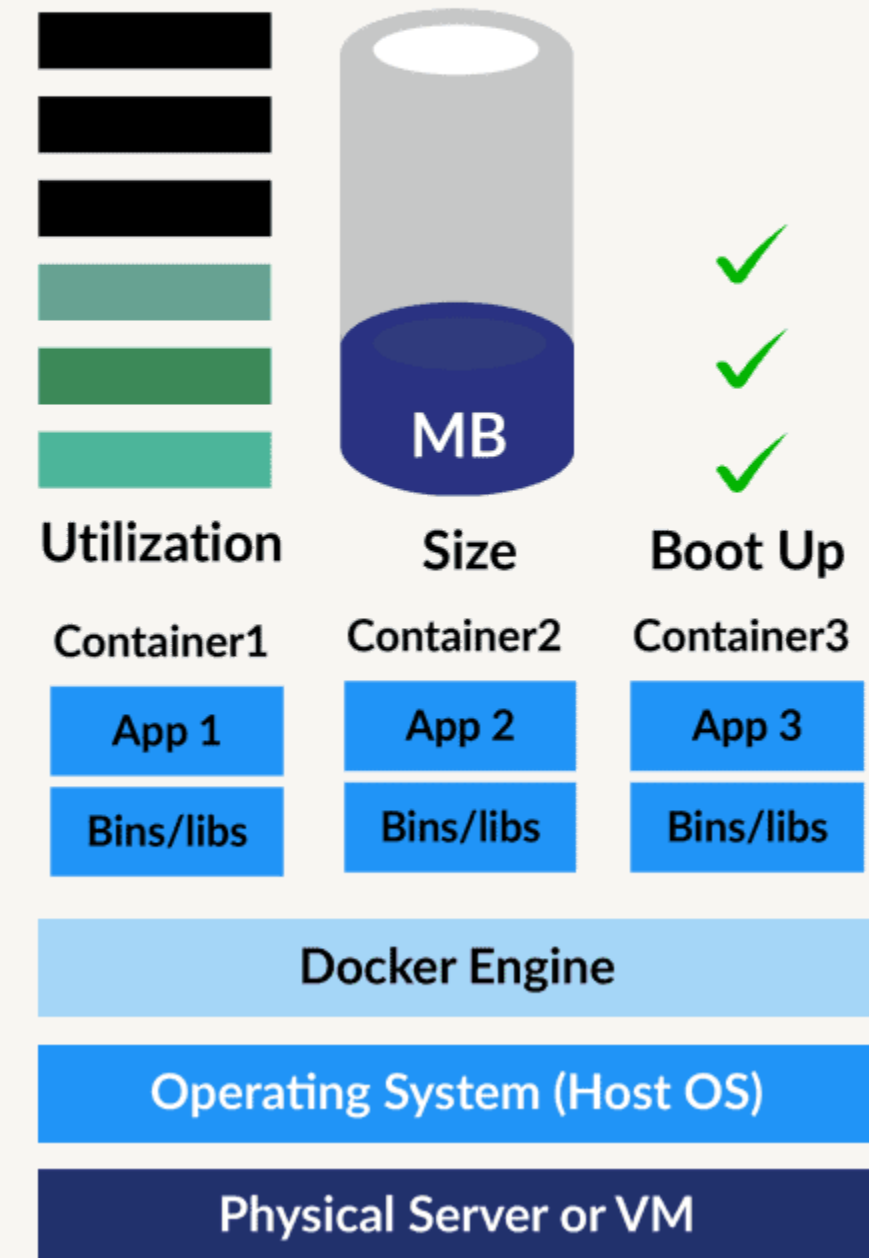
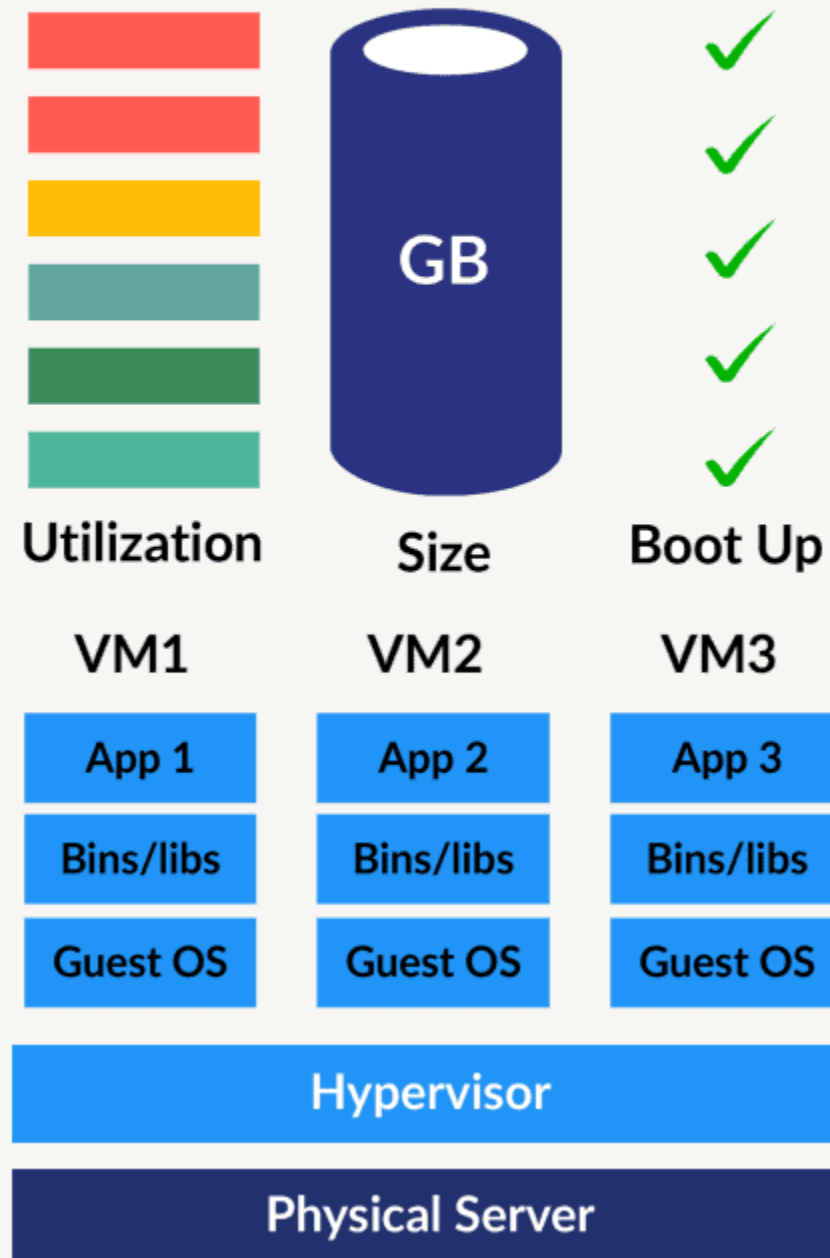
Configure your whole project with Docker
Compose

What is Docker ?

“Docker is a platform and tool that allows you to develop, deploy, and run applications in containers.” - ChatGPT

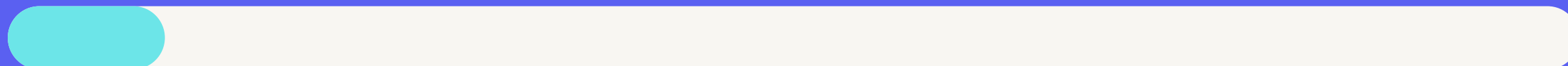


Docker vs Virtual Machine



Part 0

Installation of Docker



How to install

Windows + <https://docs.docker.com/desktop/install/windows-install/>

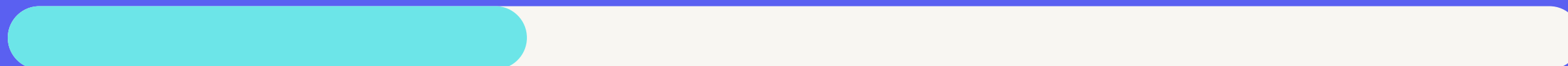
MacOS + <https://docs.docker.com/desktop/install/mac-install/>

Ubuntu + <https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository>

“If you have WSL on Windows, you will install on WSL (Ubuntu)” - Me

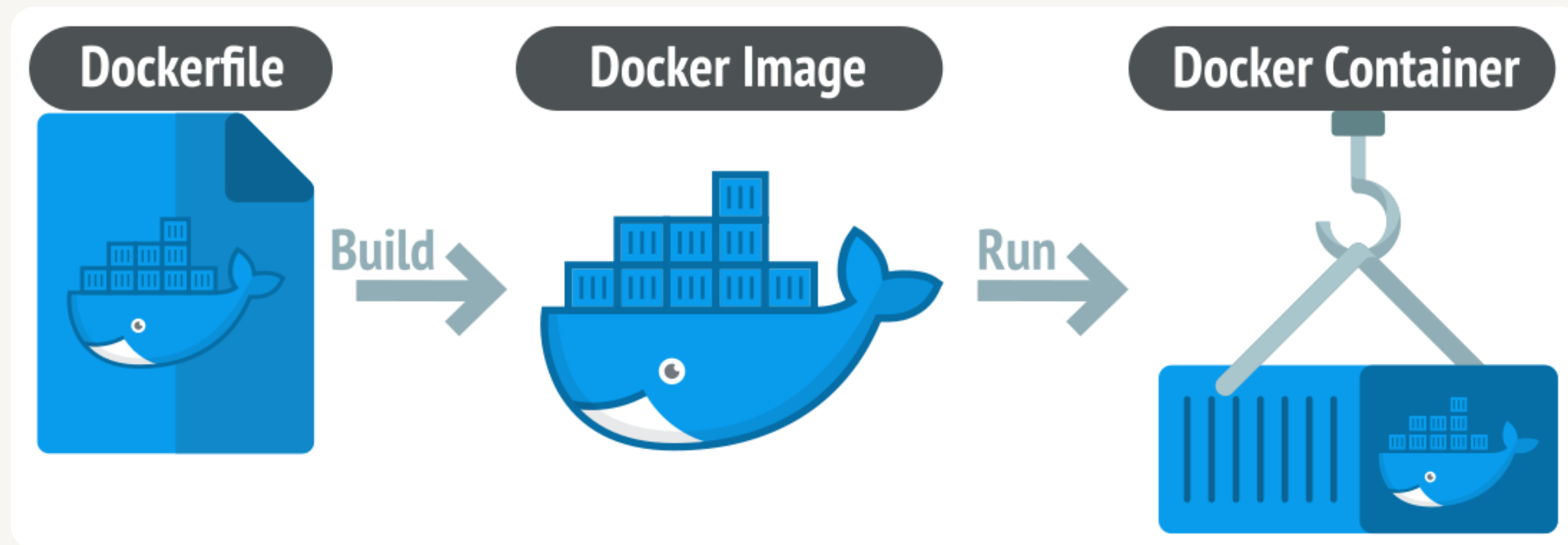
Part I

Creating your first Dockerfile



What is a Dockerfile ?

“A Dockerfile is a plain text configuration file used in Docker to define and create a Docker image.” - ChatGPT



Your first Dockerfile

Let's learn the basic keywords of Dockerfile

FROM: The FROM command specifies the base image from which your Docker image will be built.

RUN: The RUN command executes a command in the image during the image-building process.

COPY: The COPY command is used to copy files and directories from your local system into the image.

WORKDIR: The WORKDIR command sets the working directory within the container.

EXPOSE: The EXPOSE command specifies which network ports the container should listen on.

CMD: The CMD command defines the default command that should be executed when a container is started from the image.

Here is an exemple of a Dockerfile



```
1  # fetch node v4 LTS codename argon
2  FROM node:argon
3
4  # Request samplename build argument
5  ARG samplename
6
7  # Create app directory
8  RUN mkdir -p /usr/src/spfx-samples
9  WORKDIR /usr/src/spfx-samples
10
11 #Install app dependencies
12 RUN git clone https://github.com/SharePoint/sp-dev-fx-webparts.git .
13 WORKDIR /usr/src/spfx-samples/samples/$samplename
14
15 # install gulp on a global scope
16 RUN npm install gulp -g
17
18 # RUN ["npm", "install", "gulp"]
19 RUN npm install
20 RUN npm cache clean
21
22 # Expose required ports
23 EXPOSE 4321 35729 5432
24
25 # Run sample
26 CMD ["gulp", "serve"]
27
```

Docker client commands

Let's learn the basic commands of the client

docker version: This command displays the Docker client and server version information

docker images: This command lists all the Docker images that are currently available on your system.

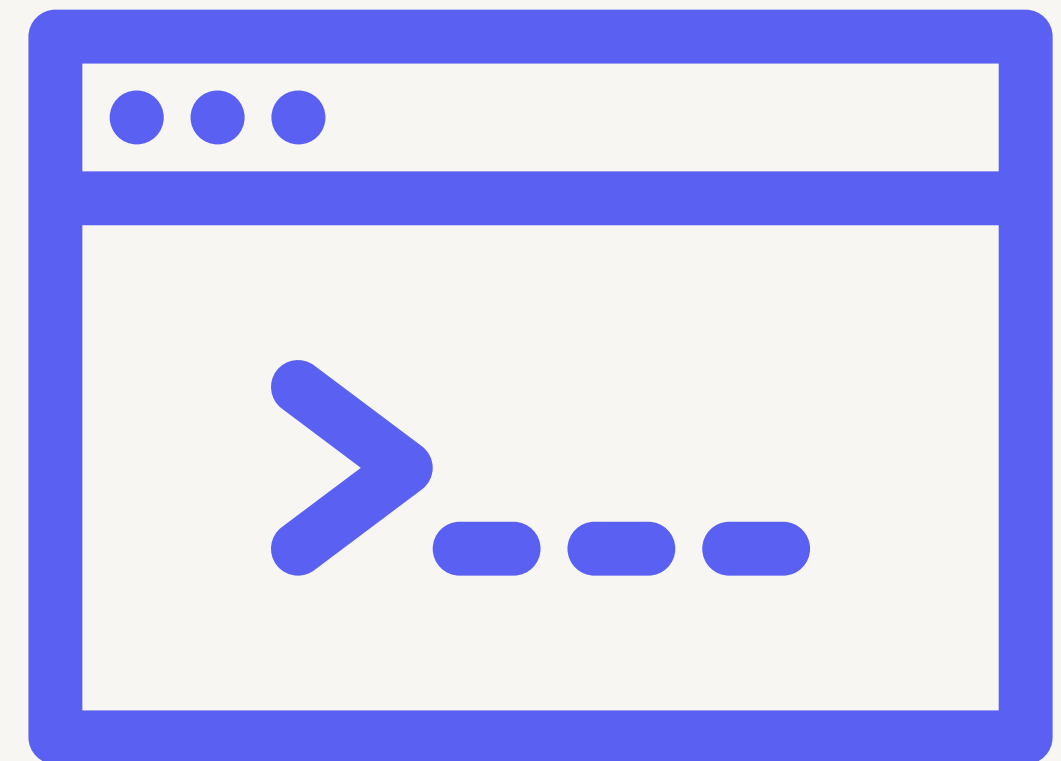
docker build: This command create an image from the specified Dockerfile

docker ps: This command shows a list of running containers

docker run: This command is used to create and start a new container from a specific image

docker stop: This command stop to halt a running container

docker start: This command restarts a stopped container



A MySQL Dockerfile

Now follow those instructions to create a Dockerfile for your MySQL database



1. You must start from the **mysql:latest** image
2. Set up environment variables for the **root password** and the **database**
3. Copy the **init.sql** file into the **/docker-entrypoint-initdb.d/** folder.
4. Expose the **3306** port

“If you want to try your MySQL database, just connect to it.” - Edouard Chhang

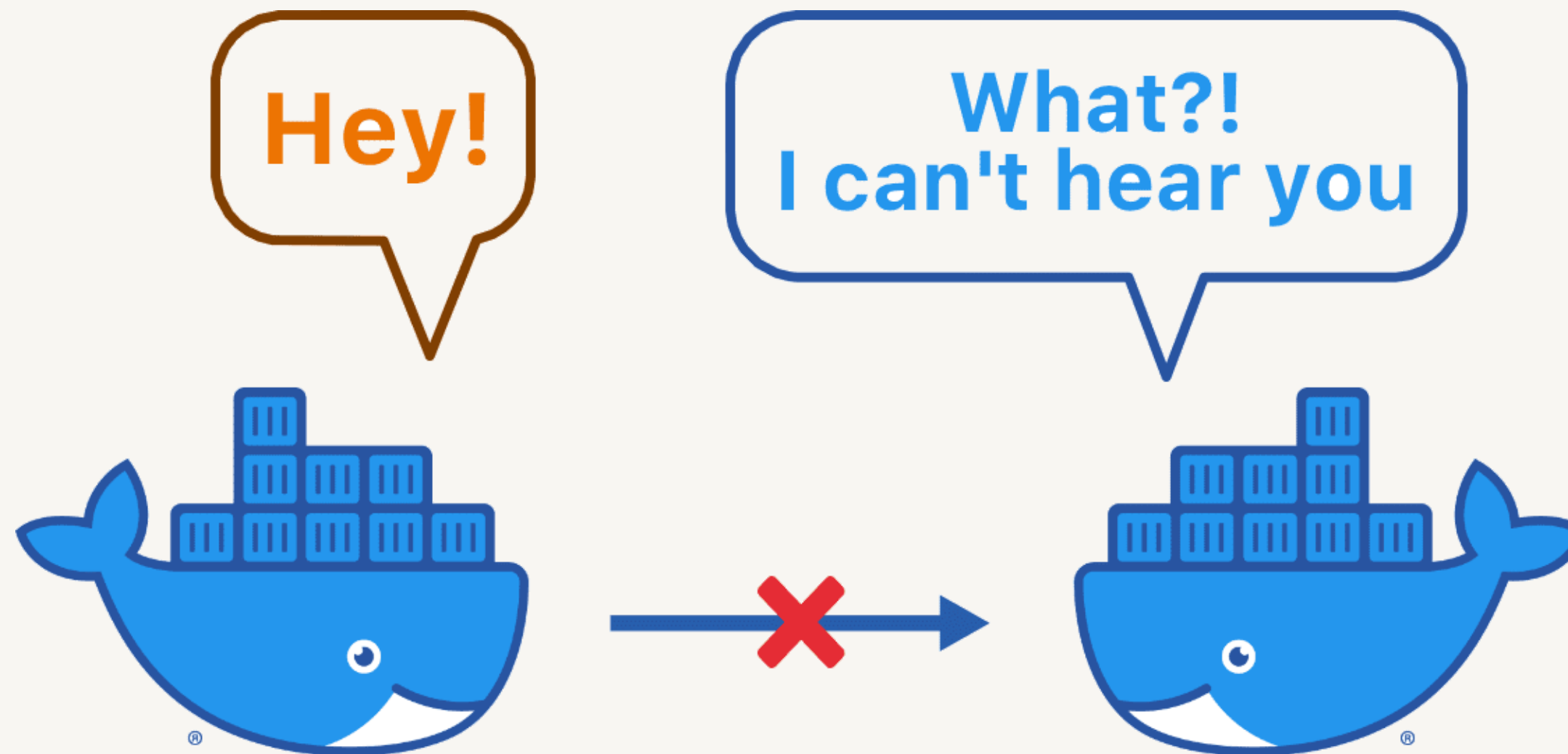
Part II

Connecting multiple docker containers



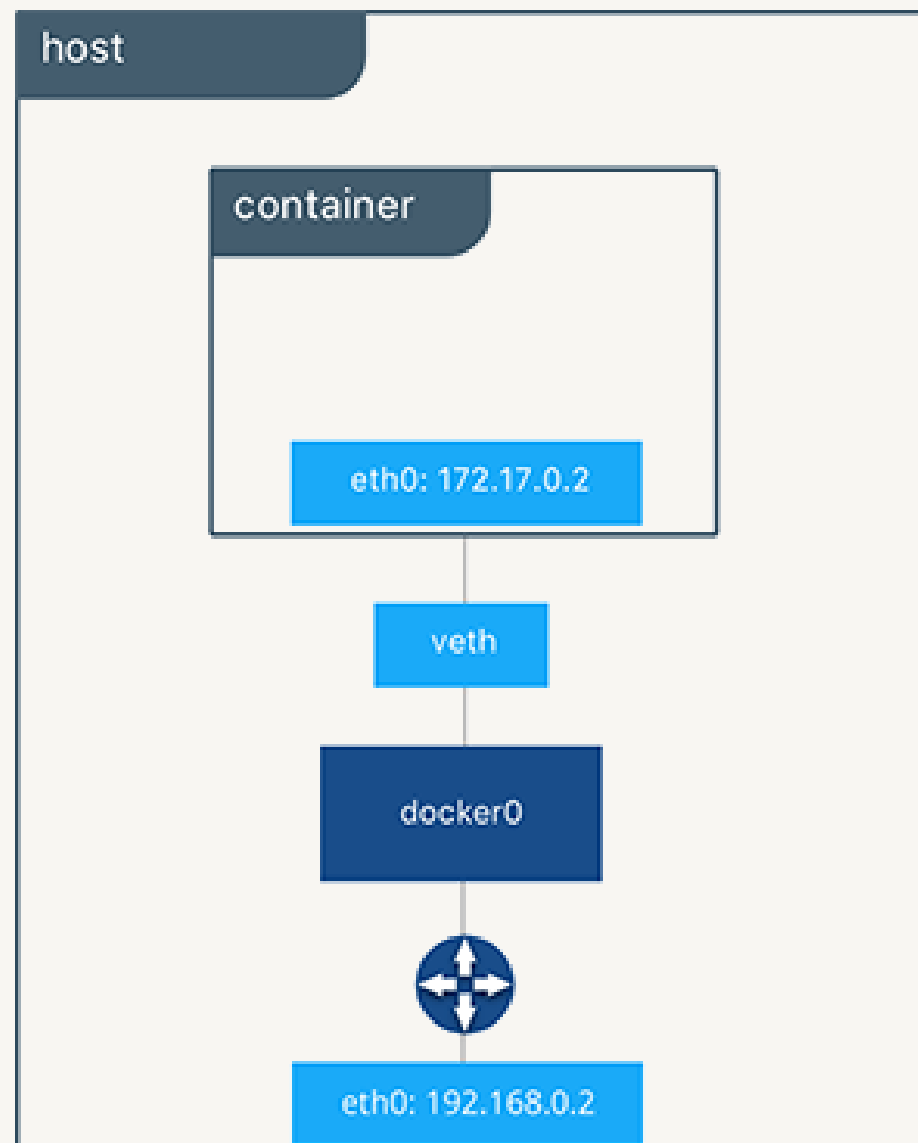
What is a Docker network ?

“A Docker network is a virtual network that allows Docker containers to communicate with each other and with external networks in a controlled and isolated manner.” - ChatGPT



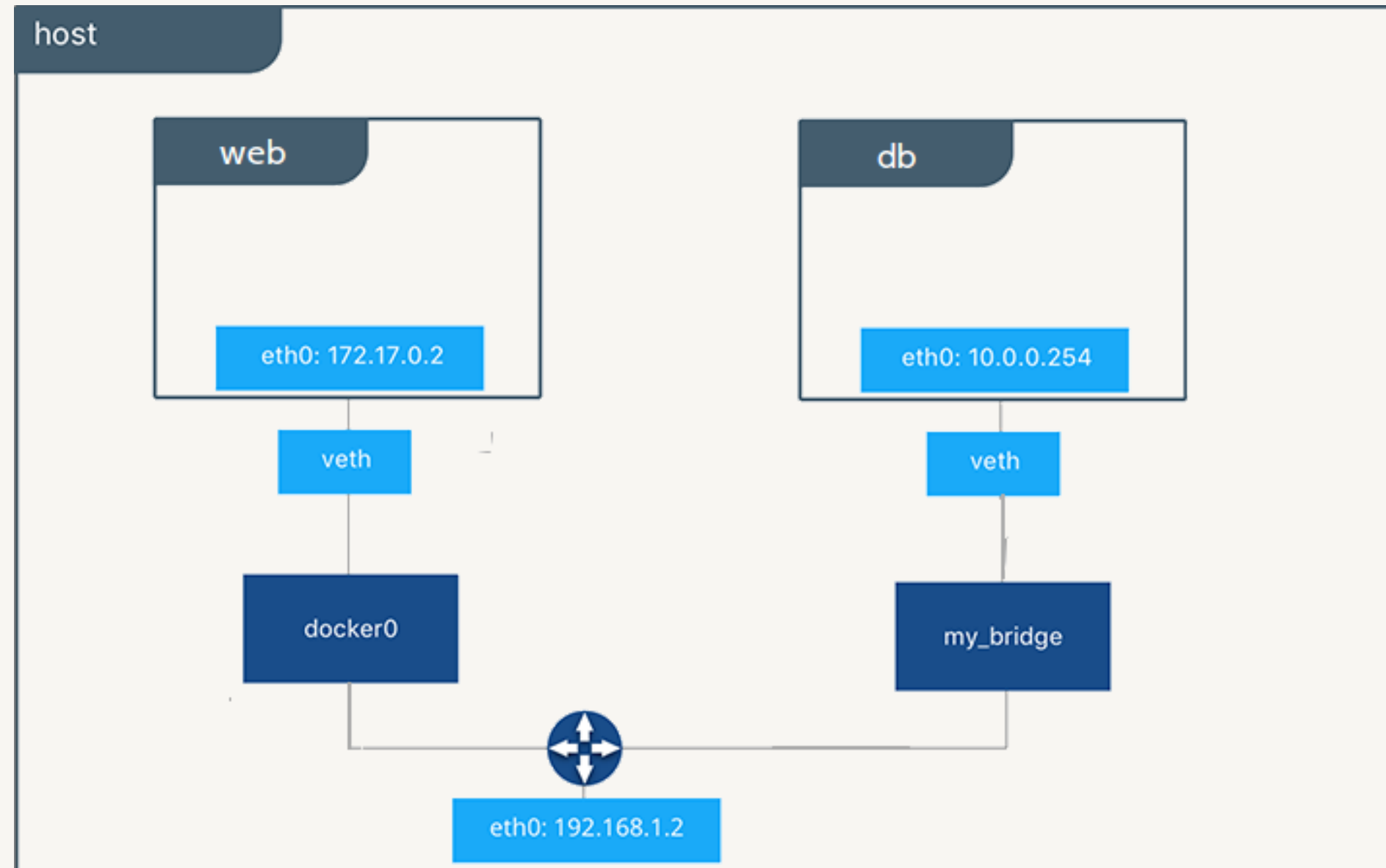
Communication between containers

Initial state of a container in networks



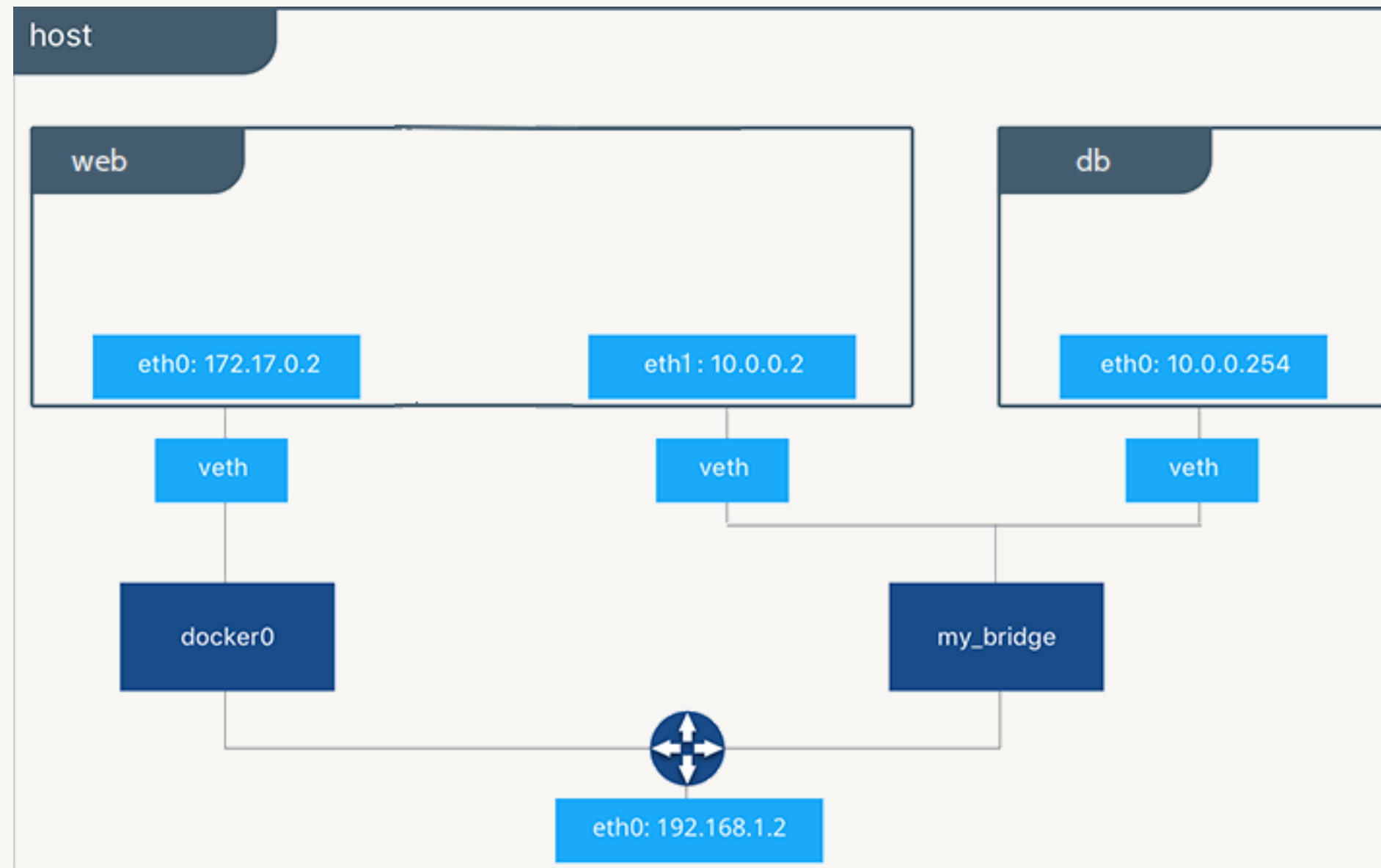
Communication between containers

Two containers without any network configuration



Communication between containers

Two containers with network configuration



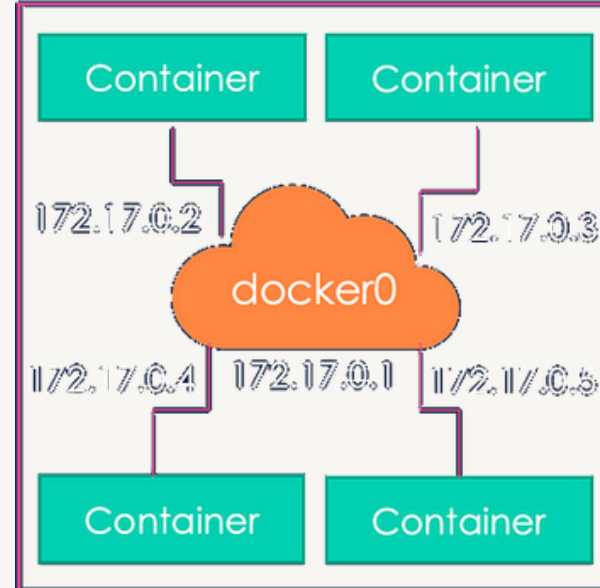
Different types of networks driver

The three basic network drivers



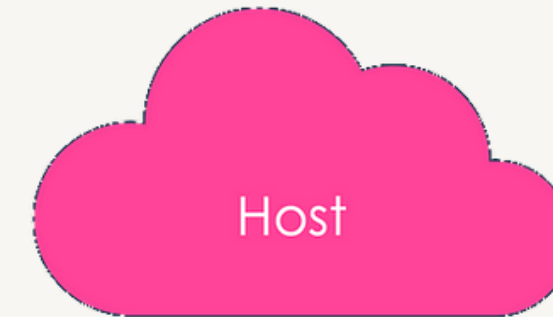
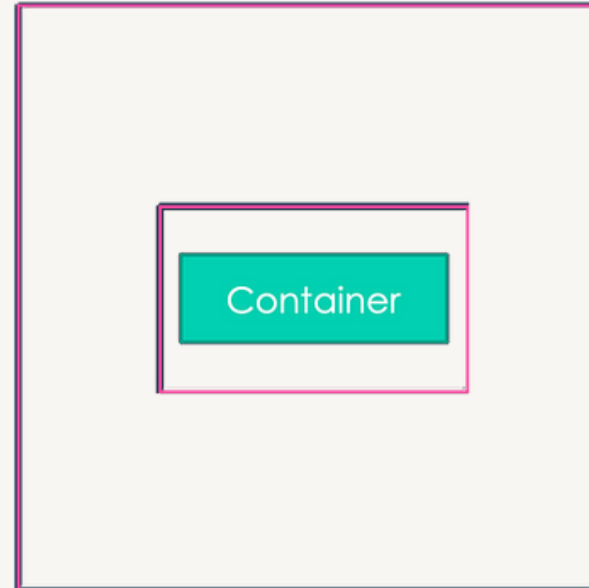
`docker run ubuntu`

Docker Host



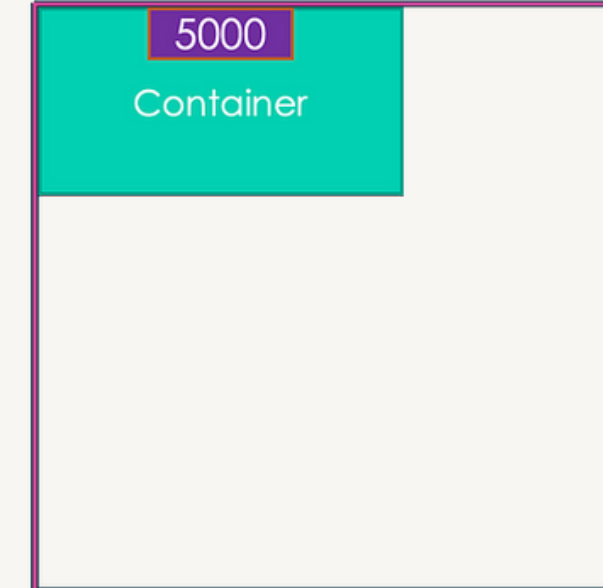
`docker run \`
`--network=none`
`ubuntu`

Docker Host



`docker run \`
`--network=host`
`ubuntu`

Docker Host



A Node.js Dockerfile

Now follow those instructions to create a Dockerfile for your Node.js API



1. You must start from the **node:18-alpine3.18** image
2. Set the working directory to **/app**
3. Copy the **application code** to the working directory
4. Install dependencies by running **npm install**
5. Expose the **3000** port
6. Finally start the application by running **npm start**

“If you want to try your Node.js API, just send a request to it.” - Edouard Chhang

Run containers with the right network

Let's take a look at the docker run command documentation to choose the right network !

Find by yourself on the [documentation](#) what option to specify with the docker run command so you can select a network for your container.

docker run

Create and run a new container from an image

Usage

```
docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

Description

The `docker run` command runs a command in a new container, pulling the image if needed and starting the container.

You can restart a stopped container with all its previous changes intact using `docker start`. Use `docker ps -a` to view a list of all containers, including those that are stopped.

“If you want to find something, just CTRL+F it.” - Edouard Chhang

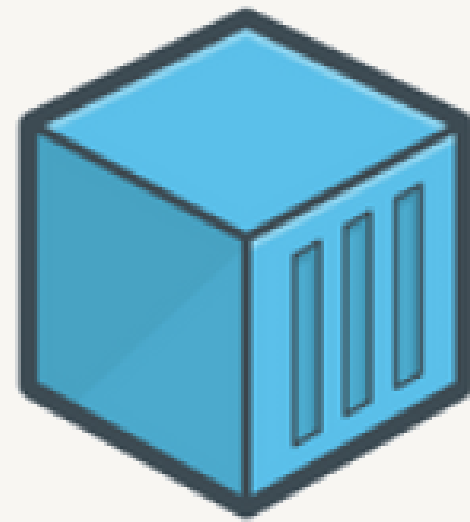
Part III

Configure the whole project all together

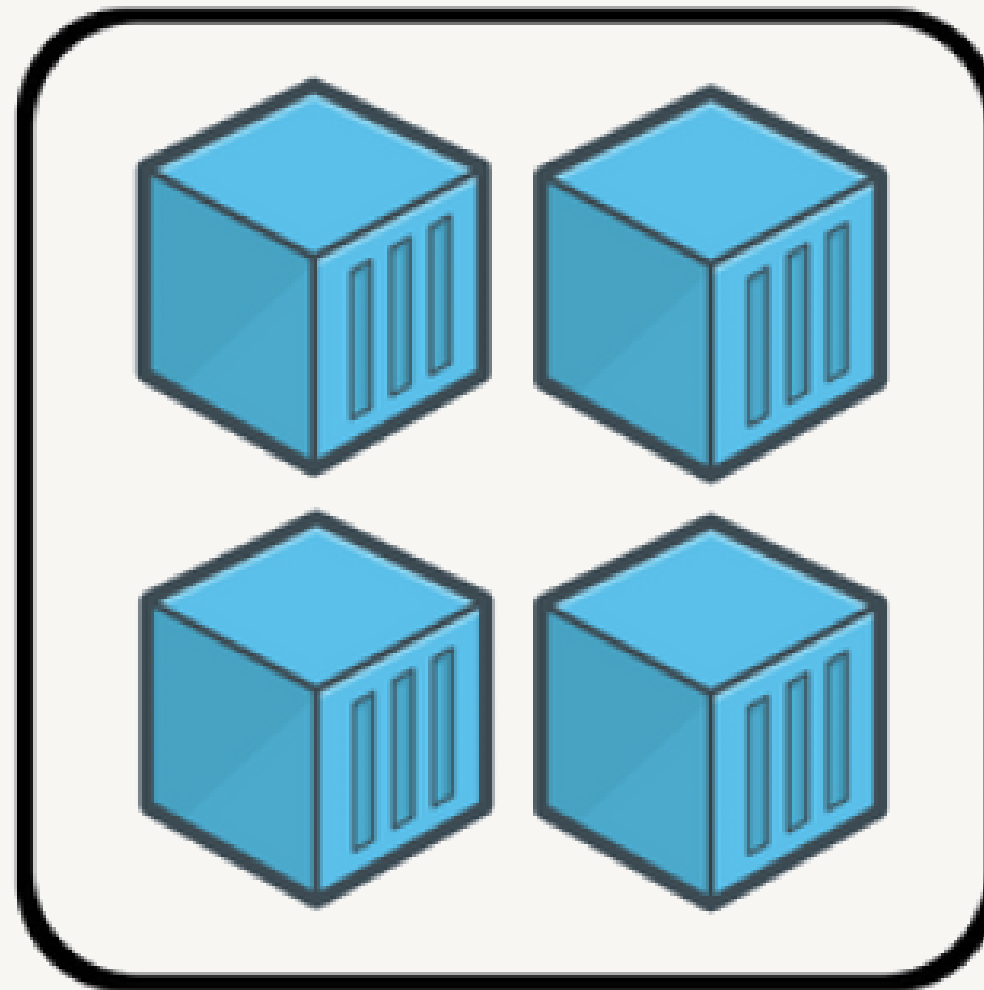


What is a Docker Compose ?

“Docker Compose is a tool for defining and running multi-container Docker applications.” - ChatGPT



Docker



Docker Compose

Your first Docker Compose

Let's learn the basic keywords of Docker Compose

version: Specifies the Compose file version.

services: Defines the services that make up your application.

image: Specifies the Docker image to use for a service.

build: Instead of specifying an image, you can use the build directive to build an image from a Dockerfile

ports: Maps container ports to host ports.

networks: Specifies custom networks for services and connects services to these networks.

environment: Sets environment variables within the container.

Here is an exemple of a Docker Compose



```
docker-compose.yml X
docker-compose.yml
1  version: "3.8"
2  services:
3      nodeserver:
4          build:
5              context: .
6          volumes:
7              - ./src:/app/src
8          environment:
9              NODE_ENV: production
10         ports:
11             - "3000:3000"
12     nginx:
13         restart: always
14         build:
15             context: ./nginx
16         ports:
17             - "80:80"
```

Docker Compose commands

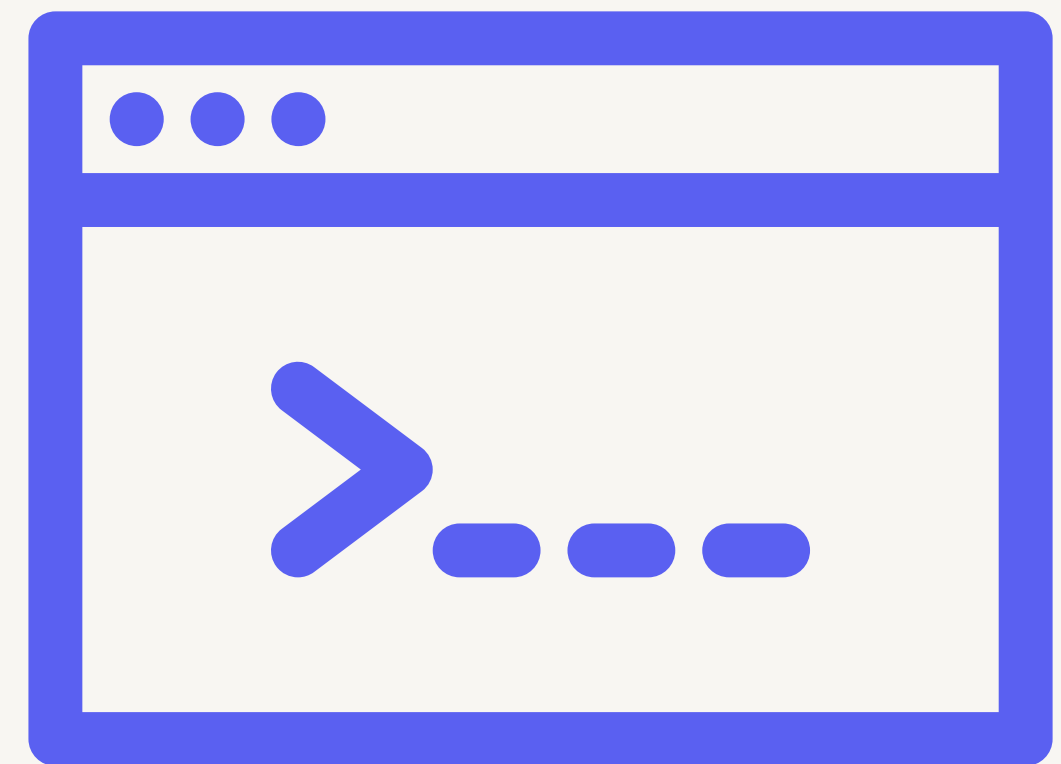
Let's learn the basic commands of Docker Compose

docker-compose up: Create and start all containers defined in the Compose file.

docker-compose down: Stop and remove all containers defined in the Compose file.

docker-compose build: Build or rebuild the images for the services defined in the Compose file.

docker-compose scale: Change the number of containers for a specific service.



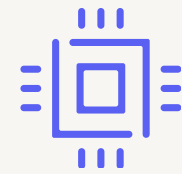
A Docker Compose for your application

It's now up to you, reuse and adapt previous Dockerfile with a Docker Compose so it runs your API and Database together

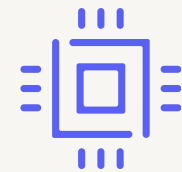


"I don't have anymore wisdom to give you." - Edouard Chhang

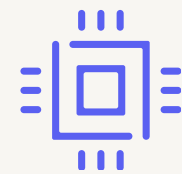
Summary



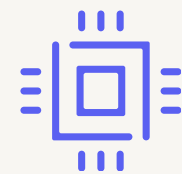
Containerization Technology



Docker Images and Containers



Isolation and Efficiency



Orchestration and Ecosystem



Thank you!

Email us at edouard.chhang@epitech.eu
if you have more questions.