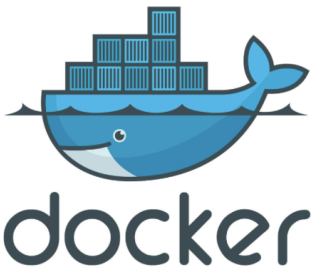
Docker

▼ Lecture

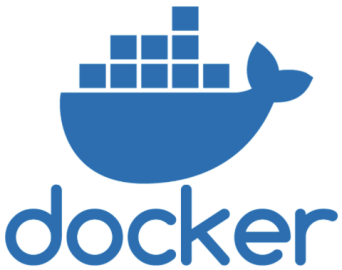
▼ Docker Intro

▼ History

- Association with VirtalCD, Ultra ISO and other virtual image technologies
- VMware and Virtual Environments
- Hello Containers as a part of system resources allocation
- Linux vs macOS vs Windows Containers

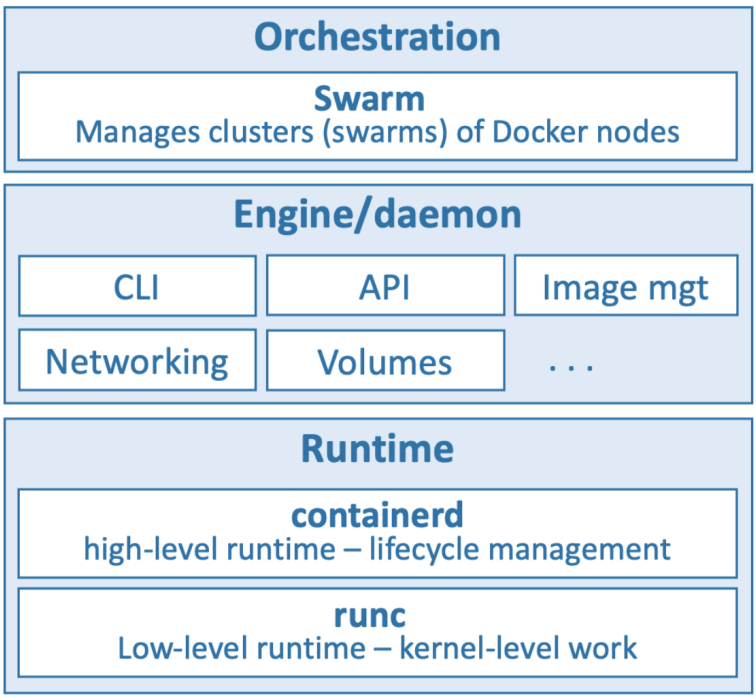


(old logo)



(new logo)

▼ The Docker technology



▼ Installing

Docker Desktop: The #1 Containerization Tool for Developers | Docker

Docker Desktop is collaborative containerization software for developers. Get started and download Docker Desktop today on Mac, Windows, or Linux.

https://www.docker.com/products/docker-desktop/

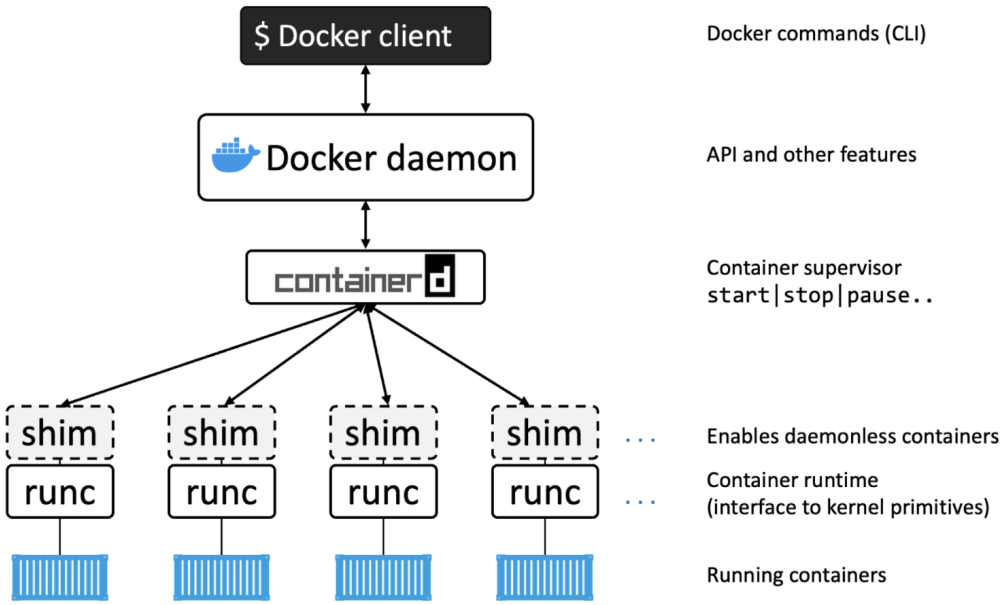
Play with Docker | Docker

Find self-paced tutorials to increase your Docker knowledge, and join a global community of collaborative developers. Play with Docker today!

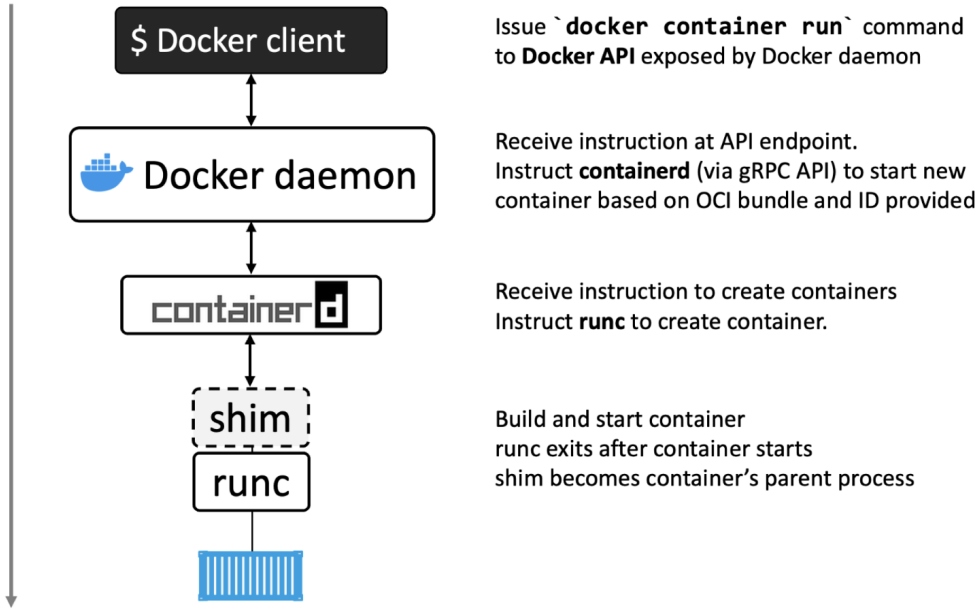
https://www.docker.com/play-with-docker/

- Server installs (Linux, CI etc)

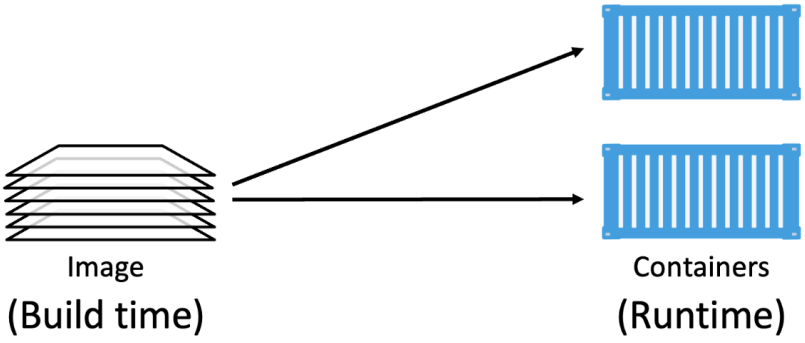
▼ The Docker Engine



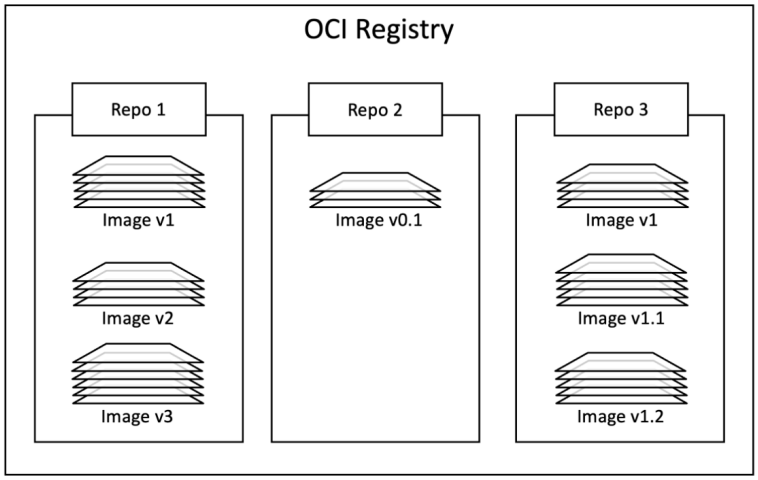
▼ Starting a new container flow



▼ Images



▼ Image registries



▼ Image naming and tagging

```
$ docker pull <repository>:<tag>

$ docker pull alpine:latest
$ docker pull postgres:15.3
```

▼ Images and layers

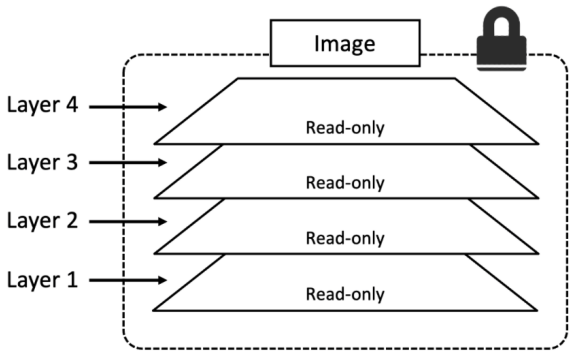


Image
Layer 4
Layer 3
Layer 2
Layer 1
Read-only
Read-only
Read-only
Read-only

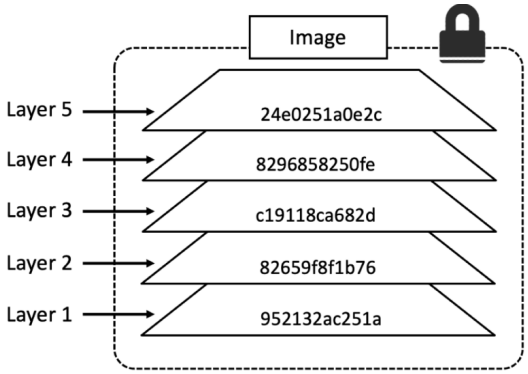


Image
Layer 5
Layer 4
Layer 3
Layer 2
Layer 1
24e0251a0e2c
8296858250fe
c19118ca682d
82659f8f1b76
952132ac251a

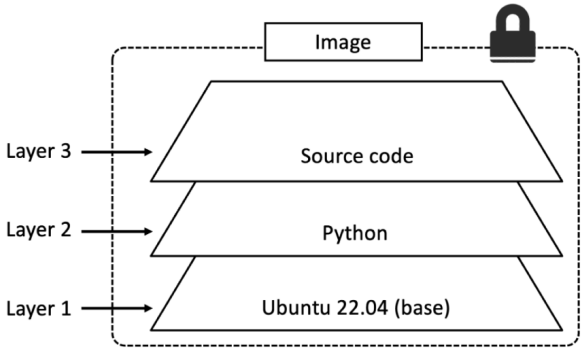


Image
Layer 3
Layer 2
Layer 1
Source code
Python
Ubuntu 22.04 (base)

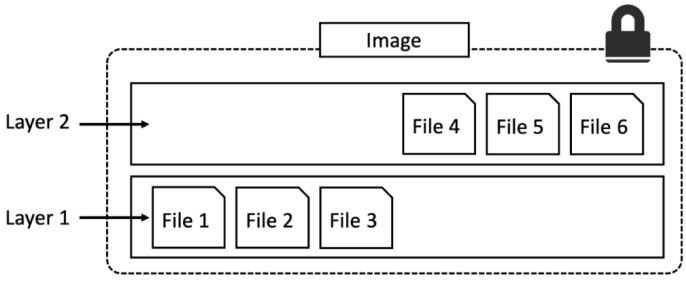
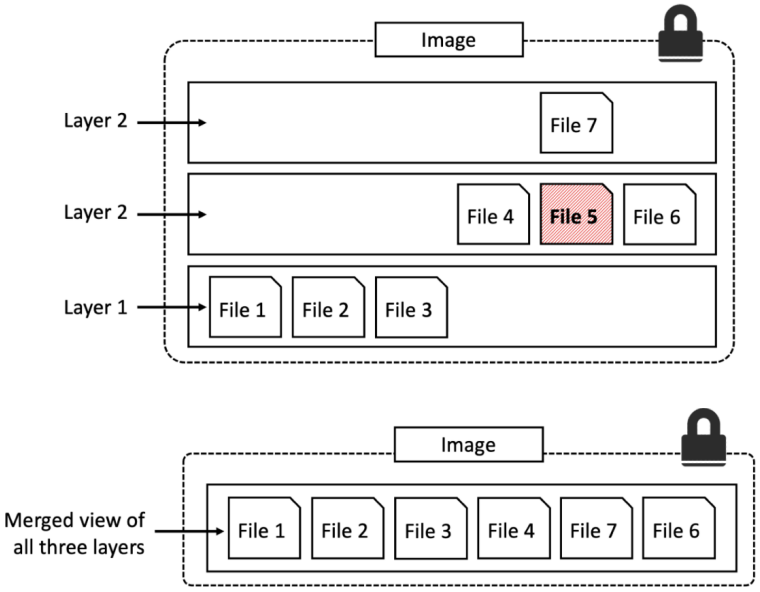
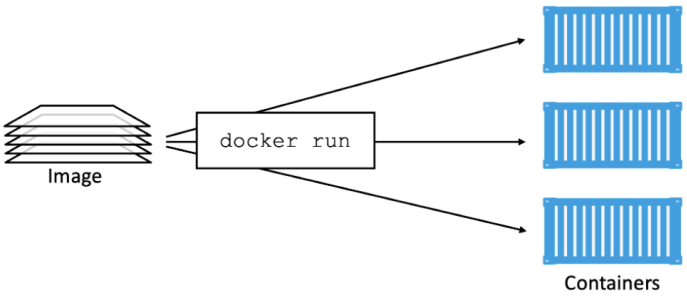


Image
Layer 2
Layer 1
File 4
File 5
File 6
File 1
File 2
File 3



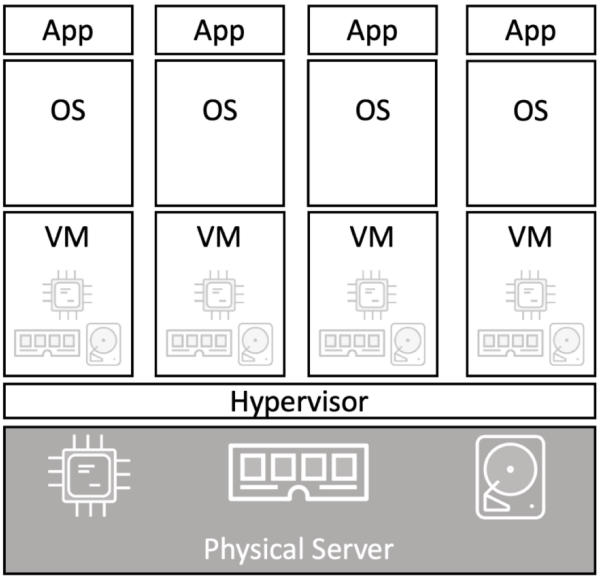
- layers can be shared

▼ Containers

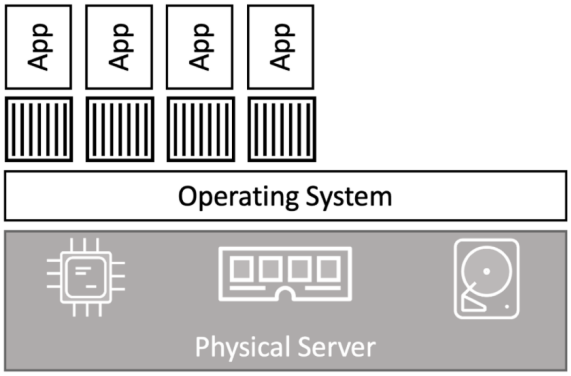


▼ Containers vs VMs

Hardware Virtualisation



OS Virtualisation



▼ Starting a single container

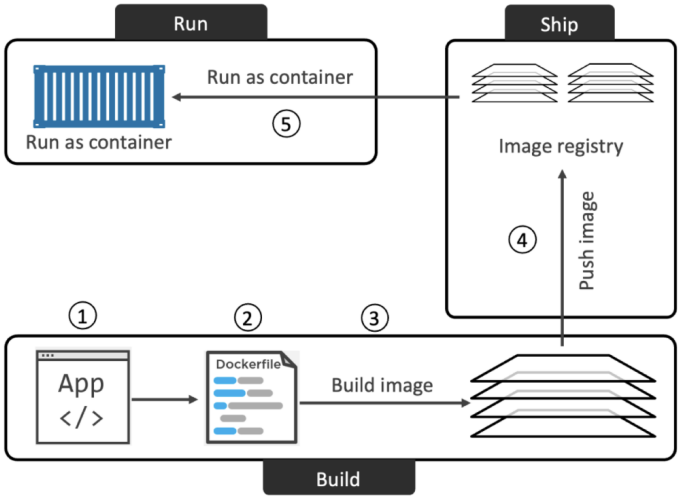
```
$ docker run -it ubuntu:latest /bin/bash

Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
79d0ea7dc1a8: Pull complete
Digest: sha256:dfd64a3b42...47492599ee20739e8eb54fbf
Status: Downloaded newer image for ubuntu:latest
root@e37f24dc7e0a:/#
```

- `docker run` tells Docker to run a new container
- `-it` flags make the container interactive and attach it to your terminal
- `ubuntu:latest` tells Docker which image should used to start the container

- `/bin/bash` is the application to run in the container

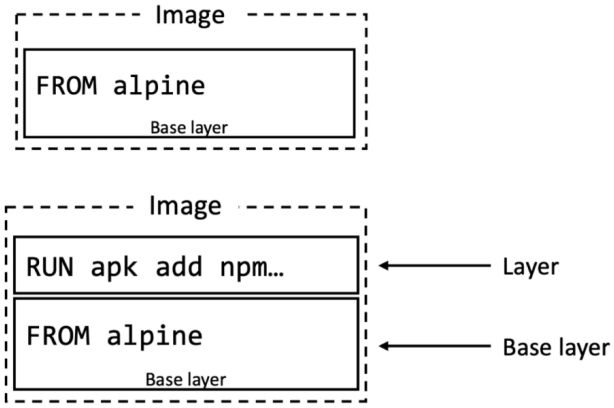
▼ Containerising an app

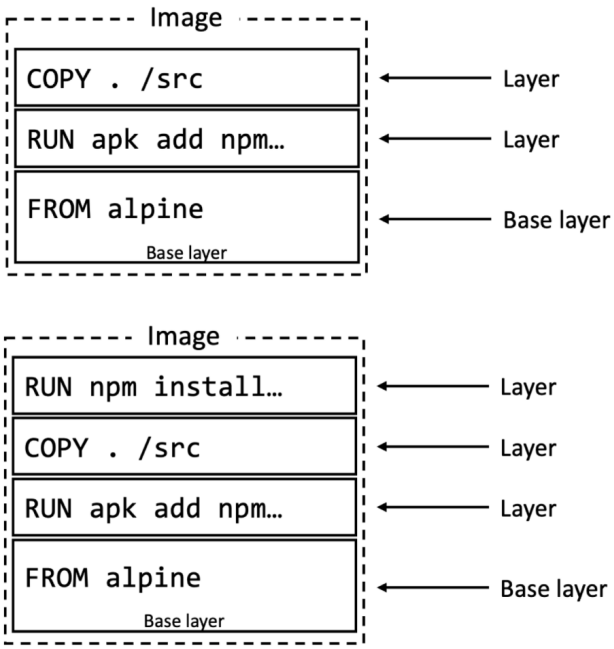
▼ Basic flow of containerising an app



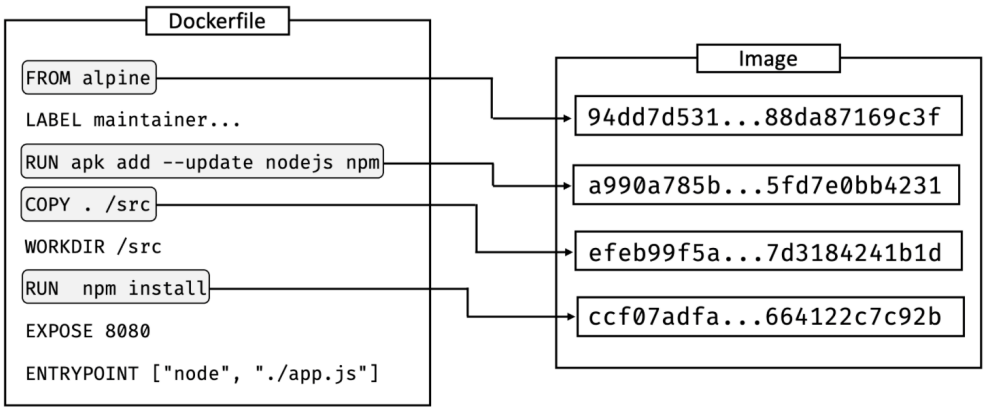
▼ Inspecting a Docker file

```
FROM alpine
RUN apk add --update nodejs npm
COPY . /src
WORKDIR /src
RUN npm install
EXPOSE 8080
ENTRYPOINT ["node", "./app.js"]
```

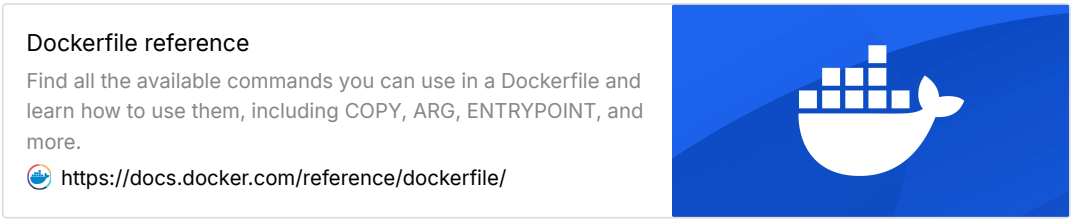




“Layer creation” vs “metadata adding” instructions



More about instructions at:

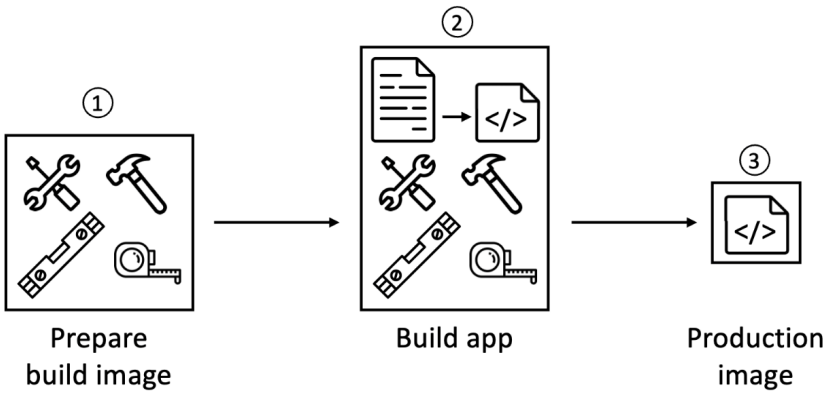


▼ Good Image recommendations

When it comes to Docker images, `big is bad!` for example:

- Big means slow

- Big means more potential vulnerabilities
- Big means a bigger attack surface



p.s. this is more advanced topic to cover in lecture, just keep in mind

▼ Other not covered topics

- Multi-stage builds (like client and server builds in same Dockerfile)
- Multi-platform builds (one image for different platforms)
- Build cache usage
- Image Squash

▼ Dockerfile Example

- Simple - and potentially small
- With Gradle build in Docker - Big and slow

▼ DVD Rental Database Dockerfile Example

▼ Starting multiple apps with Compose Example

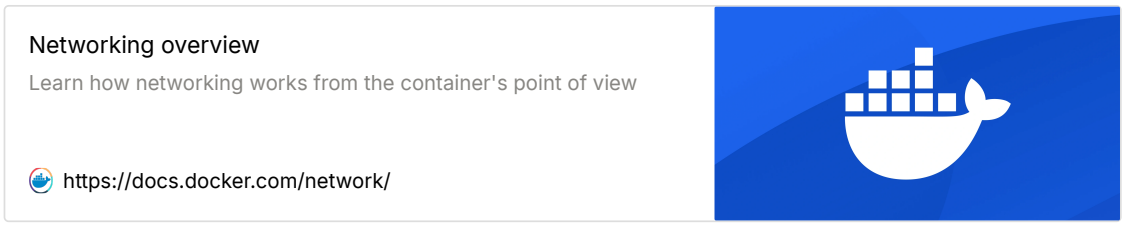
- The problem description

▼ Related topics (but not only for Compose)

Volumes

Learn how to create, manage, and use volumes instead of bind mounts for persisting data generated and used by Docker.

https://docs.docker.com/storage/volumes/



▼ Jib

- What is this?
- Why we can use it?
- `jib` vs `jibBuildTar` vs `jibDockerBuild` Gradle Tasks
  - Examples: `jibBuildTar` vs `jibDockerBuild`

▼ Jib task Publish Example

- Docker Hub Site Login
- Fetching of Image in Docker compose