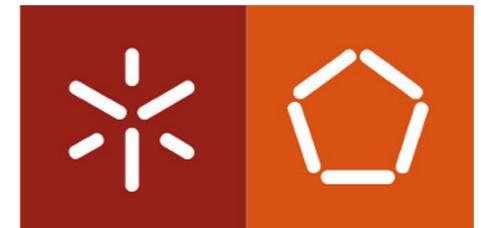


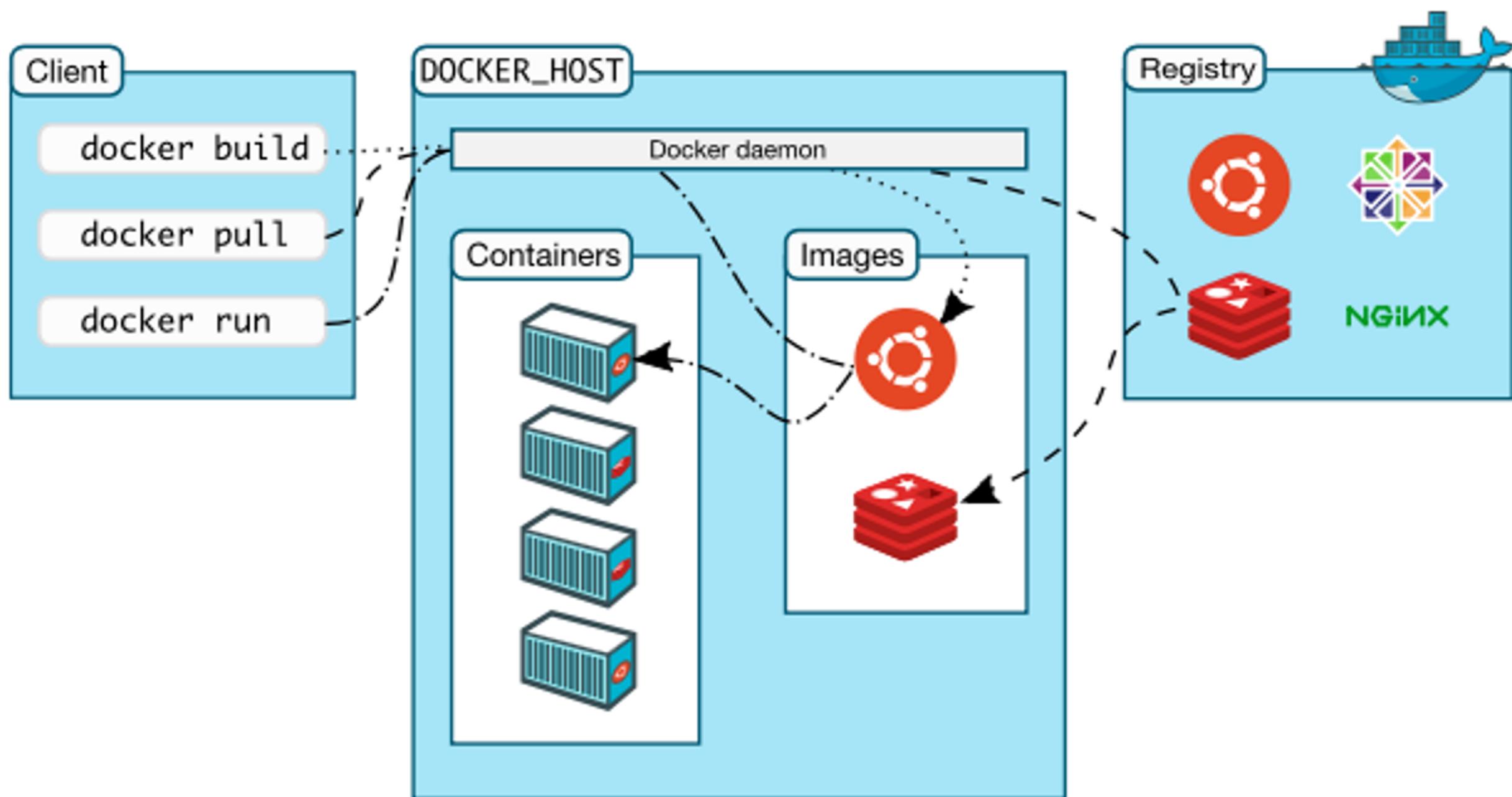
# Cloud Computing Applications and Services (Aplicações e Serviços de Computação em Nuvem)

## Docker

University of Minho  
2022/2023



# Docker



# Docker Configuration File (Dockerfile)

```
FROM node:argon

# Create app directory
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app

# Install app dependencies
COPY package.json /usr/src/app/
RUN npm install

# Bundle app source
COPY . /usr/src/app

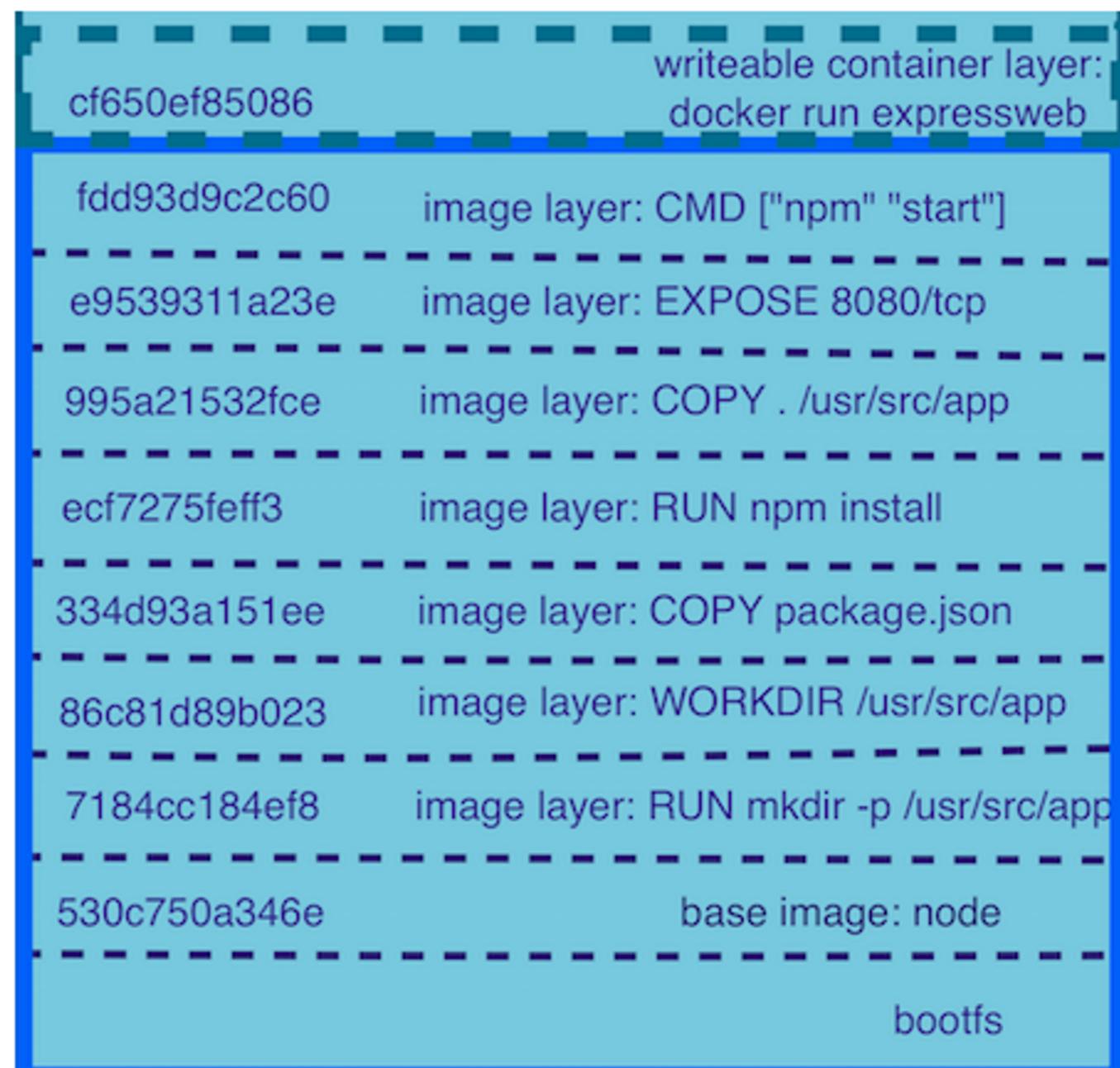
EXPOSE 8080
CMD [ "npm", "start" ]
```

# Docker Deployment

```
$ docker build -t expressweb .
Step 1 : FROM node:argon
argon: Pulling from library/node...
...
Status: Downloaded newer image for node:argon
--> 530c750a346e
Step 2 : RUN mkdir -p /usr/src/app
--> Running in 5090fde23e44
--> 7184cc184ef8
Removing intermediate container 5090fde23e44
Step 3 : WORKDIR /usr/src/app
--> Running in 2987746b5fba
--> 86c81d89b023
Removing intermediate container 2987746b5fba
Step 4 : COPY package.json /usr/src/app/
--> 334d93a151ee
Removing intermediate container a678c817e467
Step 5 : RUN npm install
--> Running in 31ee9721cccb
--> ecf7275feff3
Removing intermediate container 31ee9721cccb
Step 6 : COPY . /usr/src/app
--> 995a21532fce
Removing intermediate container a3b7591bf46d
Step 7 : EXPOSE 8080
--> Running in fddb8afb98d7
--> e9539311a23e
Removing intermediate container fddb8afb98d7
Step 8 : CMD npm start
--> Running in a262fd016da6
--> fdd93d9c2c60
Removing intermediate container a262fd016da6
Successfully built fdd93d9c2c60
```

# Docker Deployment

- The first layer is writable while the others are read-only
- Data persistency needs to be ensured with bind mounts or volumes!



# Docker Persistent Storage

- Mount a file or directory from the host machine at the container. Stored data is independent from the container's internal file system (Union FS) and persisted even if the container is terminated
- Bind mount - generic directory from the host machine. Any container or host process can access this data
- Volume - Special directory at the host managed by Docker and only accessible by containers
- <https://docs.docker.com/storage/volumes/>

# Docker Network

- Host - shares the host networking namespace. Container services are presented in the network as if they were run by the host. Ports are shared (e.g., port 80)
- Bridge - the container is seen as another node in the physical network
- <https://docs.docker.com/network/>