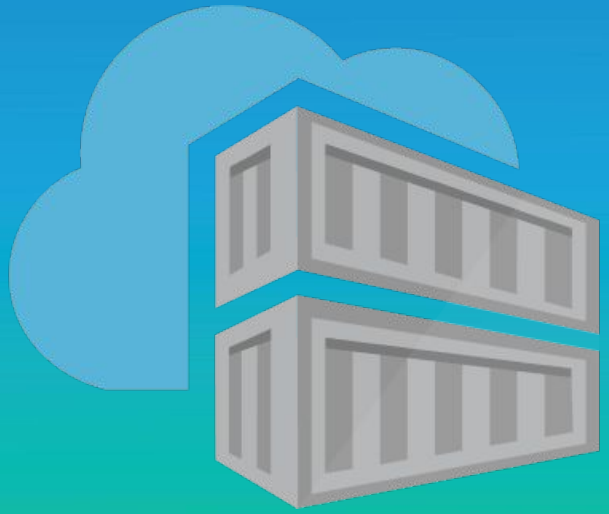
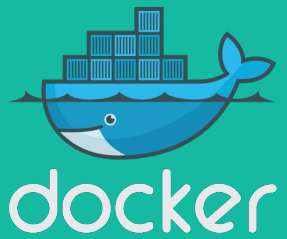
Pengenalan Docker





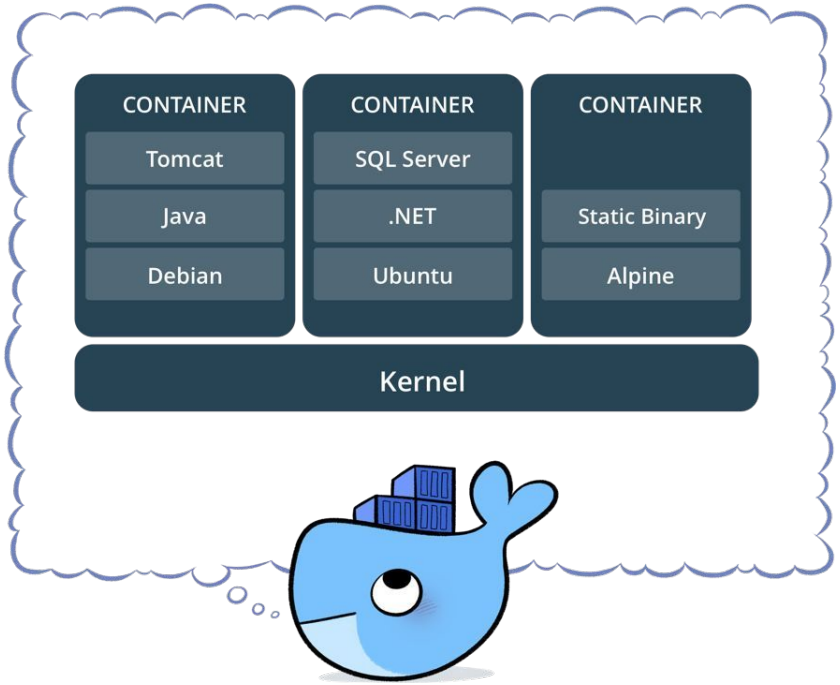
Section 1:

Docker itu apa ?

Perintah dasar docker

Dockerfiles

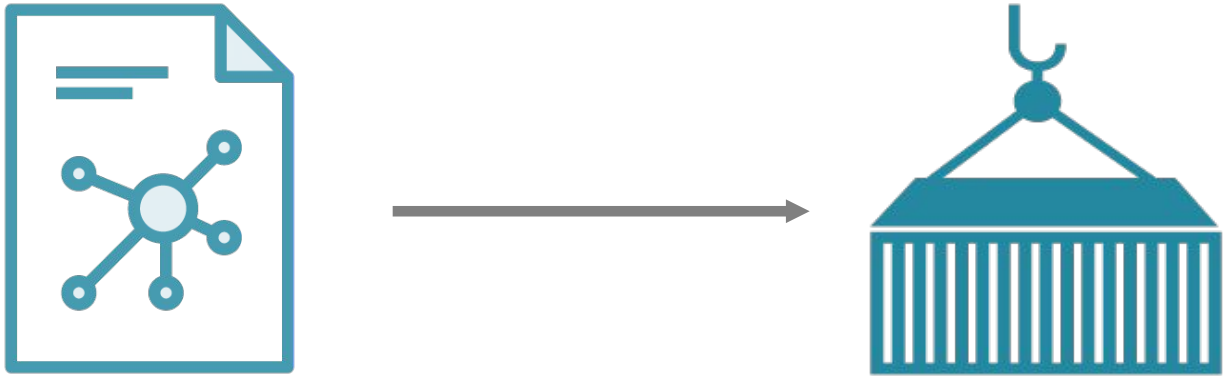
Container itu apa ?



- Standarisasi pemaketan untuk perangkat lunak dan dependensi/library
- Memisahkan/mengisolasi aplikasi satu dengan yang lain
- Menggunakan kernel OS yang sama
- Bekerja untuk semua distribusi Linux
- Dapat berjalan native juga di Windows Server 2016

docker

Tugas Images dan Container



Docker Image

Contoh: Ubuntu dengan Node.js dan Application

Docker Container

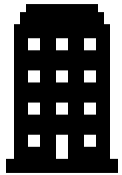
dibuat dengan menggunakan docker image dan menjalankan aplikasi yang dibuat.

docker

Docker containers bukan mesin VM

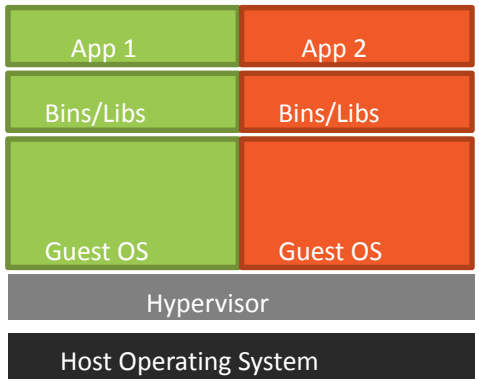
- Koneksi mudah dibuat
- Arsitektur yang sangat berbeda
- Manfaat yang berbeda secara fundamental

Virtual Mesin

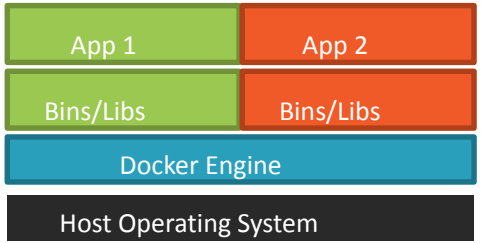


Container

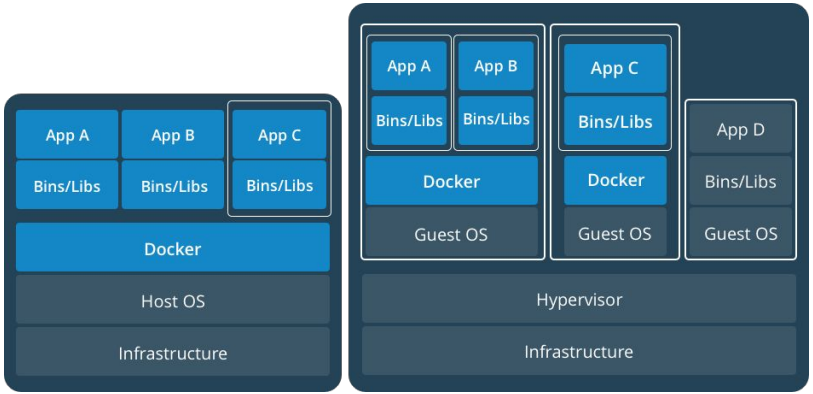
Docker Containers Versus Virtual Machines



Virtual Machines

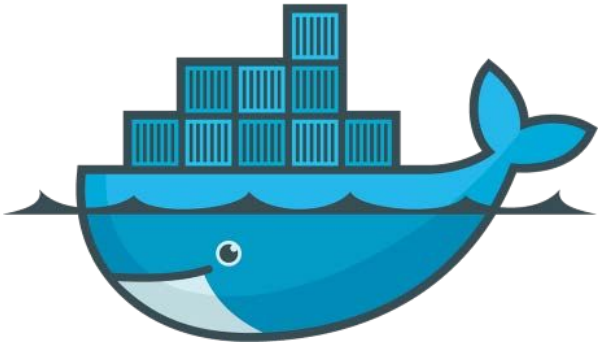


Docker Containers



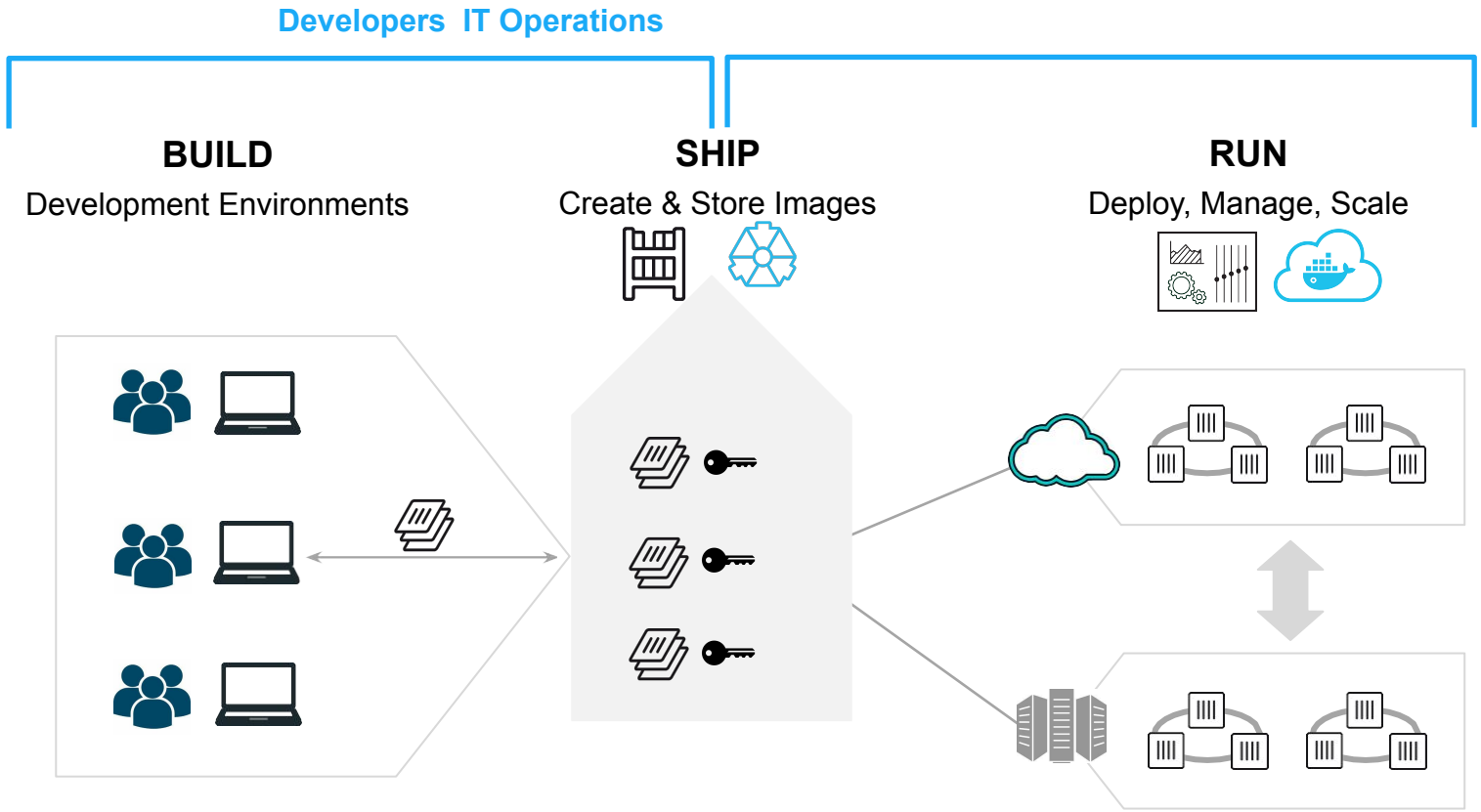
docker

Docker itu apa?



- Platform yang ringan, terbuka, dan aman
- Menyederhanakan proses pembuatan (building), pengiriman (shipping), menjalankan aplikasi (running apps).
- Berjalan secara native di Linux atau Windows Server
- Dapat berjalan dengan mode development pada Windows atau Mac (dengan mesin virtual)
- Mengandalkan "image" dan "container"

Proses di Docker: Build, Ship, Run Workflow

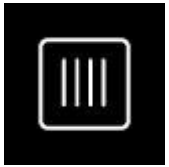


docker

Istilah dalam docker

Docker Image

Basis dari docker container yang merepresentasikan keseluruhan aplikasi.



Docker Container

Unit standar tempat layanan aplikasi berada dan dijalankan

Docker Engine

Membuat, mengirimkan dan menjalankan container docker yang di upload (deploy) secara fisik ataupun virtual pada datacenter atau layanan cloud.

Registry Service (Docker Hub(Public) or Docker Trusted Registry(Private))

Server yang menyimpan image dari container dapat berupa cloud atau server baremetal.

Contoh perintah dasar docker

```
$ docker image pull node:latest

$ docker image ls
$ docker container run -d -p 5000:5000 --name node node:latest

$ docker container ps


$ docker container stop node(or <container id>)

$ docker container rm node (or <container id>)

$ docker image rmi (or <image id>)

$ docker build -t node:2.0 .

$ docker image push node:2.0

$ docker --help
```

Dockerfile (contoh aja)

Dockerfile

```
# Create image based on the official Node 6 image from dockerhub
FROM node:latest

# Create a directory where our app will be placed
RUN mkdir -p /usr/src/app

# Change directory so that our commands run inside this new directory
WORKDIR /usr/src/app

# Copy dependency definitions
COPY package.json /usr/src/app

# Install dependecies
RUN npm install

# Get all the code needed to run the app
COPY . /usr/src/app

# Expose the port the app runs in
EXPOSE 4200

# Serve the app
CMD ["npm", "start"]
```

- Sekumpulan instruksi terkait bagaimana membangun docker image
- Perintah yang digunakan mudah dipahami dan native
- Penting untuk selalu optimalisasi Dockerfile

Section 2:
Struktur Docker Container
Docker Volumes
Volume Use Cases

Recap Dockerfile

Dockerfile

```
# Create image based on the official Node 6 image from dockerhub
FROM node:latest

# Create a directory where our app will be placed
RUN mkdir -p /usr/src/app

# Change directory so that our commands run inside this new directory
WORKDIR /usr/src/app

# Copy dependency definitions
COPY package.json /usr/src/app

# Install dependecies
RUN npm install

# Get all the code needed to run the app
COPY . /usr/src/app

# Expose the port the app runs in
EXPOSE 4200

# Serve the app
CMD ["npm", "start"]
```

Setiap perintah di Dockerfile membuat sebuah Layer

Docker Image Pull: Pulls Layers

```
$ docker pull nginx:latest
latest: Pulling from library/nginx
bc95e04b23c0: Pull complete
f3186e650f4e: Pull complete
9ac7d6621708: Pull complete
Digest: sha256:b81f317384d7388708a498555c28a7cce778a8f291d90021208b3eba3fe74887
Status: Downloaded newer image for nginx:latest
```

Docker Volumes

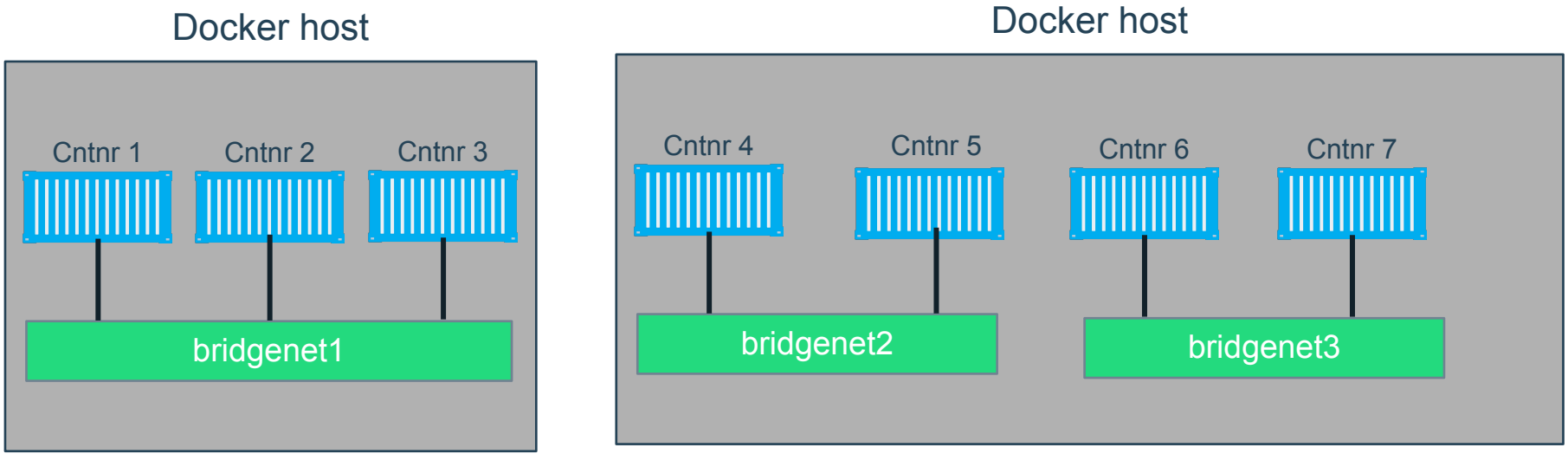
- Volume memasang (mount) direktori pada host ke dalam container dengan lokasi spesifik
- Dapat digunakan untuk berbagi dan mempertahankan data antar container
- Direktori tetap ada setelah container dihapus, kecuali Anda menghapusnya secara total sampai penyimpanan.
- Dapat dibuat di Dockerfile atau melalui perintah

Kenapa menggunakan Volumes ?

- Dapat mount source code local mesin ke container
  - `docker container run -v $(pwd):/usr/src/app/ myapp`
- Meningkatkan performa
  - Karena struktur direktori menjadi rumit, melintasi struktur direktori dapat memperlambat kinerja sistem
- Data tetap terjamin

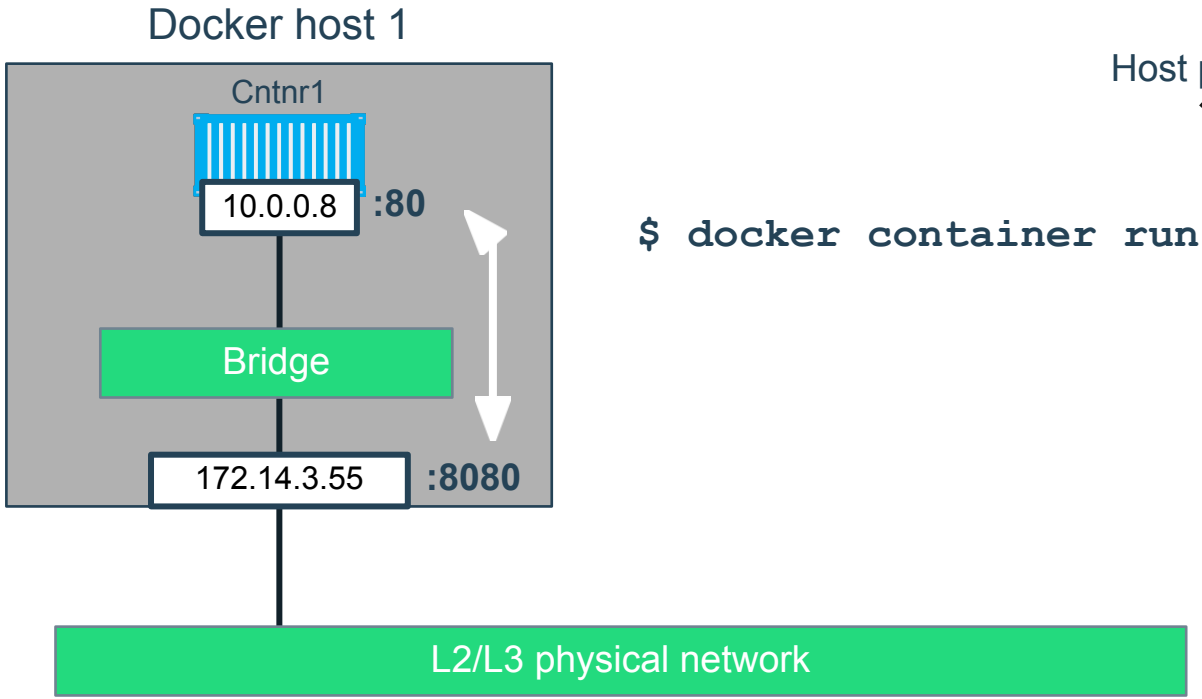
Section 3:
Networking

Apa itu Docker Bridge Networking




```
docker network create -d bridge --name
bridgenet1
```

Docker Bridge Networking dan Port Mapping



Host port

Container port

```
$ docker container run -p 8080:80 ...
```

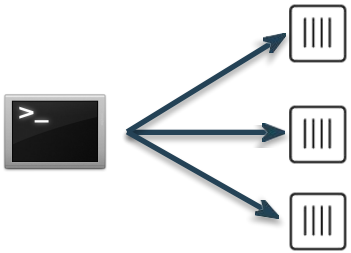

docker

Section 4:
Docker Compose

Docker Compose: Multi Container Applications

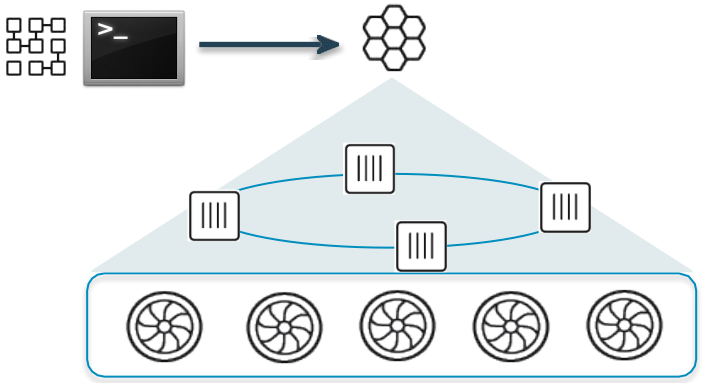
Tanpa docker compose

- Membangun dan menjalankan satu container dalam satu waktu
- Menghuubungkan kontainer secara manual
- Harus berhati-hati dengan ketergantungan library dan urutan ketika menjalankan

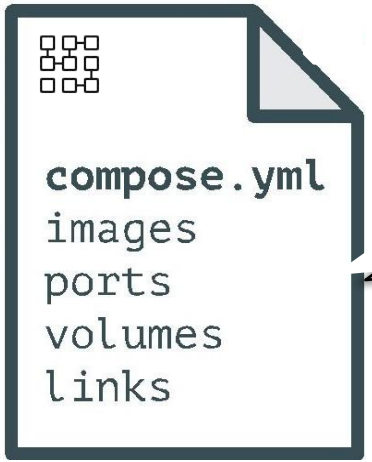


Dengan docker compose

- Membuat aplikasi dengan banyak (multi) container cukup di file compose.yml
- Cukup menjalankan satu perintah untuk menerapkan konfigurasi/menjalankan seluruh aplikasi
- Dapat Menangani dependensi/library pada container
- Dapat Bekerja dengan Docker Swarm, Networking, Volume, Universal Control Plane



Docker Compose: Multi Container Applications

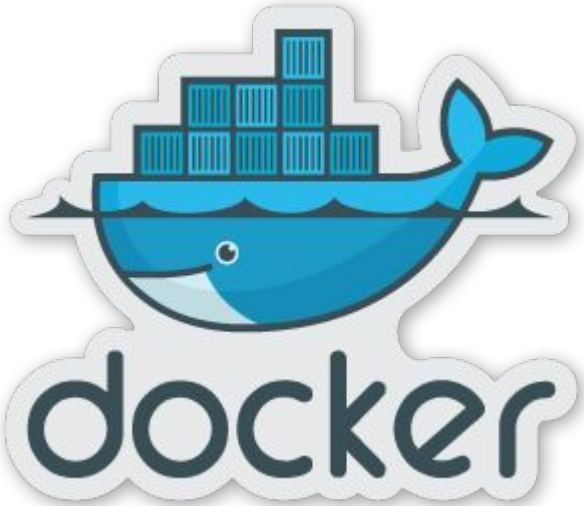



```
version: '2' # specify docker-compose version

# Define the services/containers to be run
services:
angular: # name of the first service
build: client # specify the directory of the Dockerfile
ports:
- "4200:4200" # specify port forewarding

express: #name of the second service
build: api # specify the directory of the Dockerfile
ports:
- "3977:3977" #specify ports forewarding

database: # name of the third service
image: mongo # specify image to build container from
ports:
- "27017:27017" # specify port forewarding
```

Docker Compose: Scale Container Applications

docker

Cobain belajar docker online



Online Lab :
https://killercoda.com/playgrounds/scenario/ubuntu

Online lab docker:
https://labs.play-with-docker.com/

docker