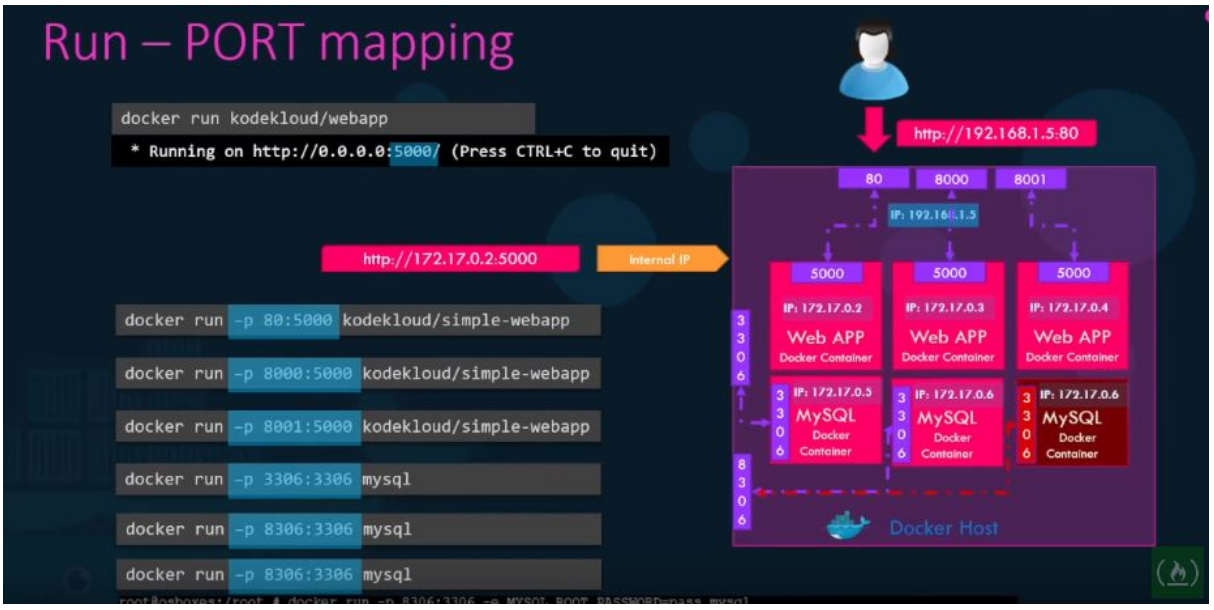
PORT mapping

docker run -p 8306:3306 mysql

8306 porta do host

3306 porta do container



VOLUME mapping

#todos dados são perdidos

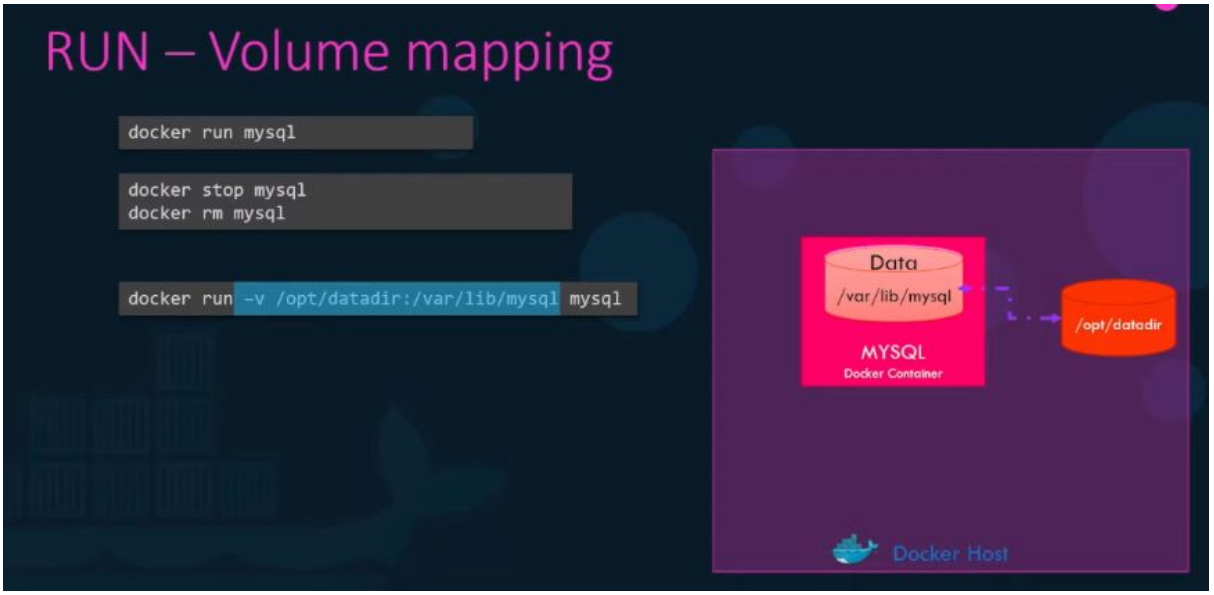
docker run mysql → salvo dados no container → docker stop mysql → docker rm mysql

#persistência de dados em um volume montado no disco local

docker run -v /opt/datadir:/var/lib/mysql mysql

/opt/datadir do host

/var/lib/mysql do container



Inspect containers : docker inspect

Container log: docker logs

ENV Variables in Docker

```
docker run -e ENV=value test
docker ispect test

{...
 “config”: {
          “Env”: { “ENV=value” }
             }
...
}
```

```
#Remover todos containers
docker rm $(docker ps -a -q)

#Remover todas as imagens
docker rmi $(docker image list -a -q)
```

Dockerfile

Create a image with:
Ubuntu OS → Update apt repo → Install dependencies using apt → install python dependencies using pip → copy source code to /opt folder → Run the web server using flask command.

Dockerfile

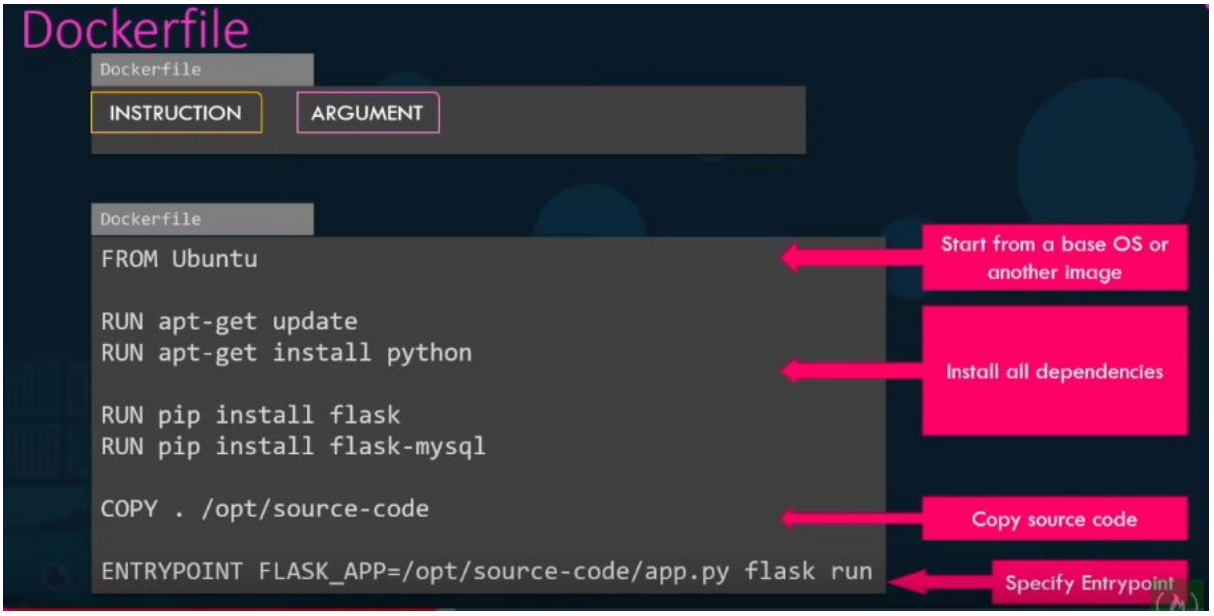
```
FROM Ubuntu

RUN apt-get update
RUN apt-get install python

RUN pip install flask
RUN pip install flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

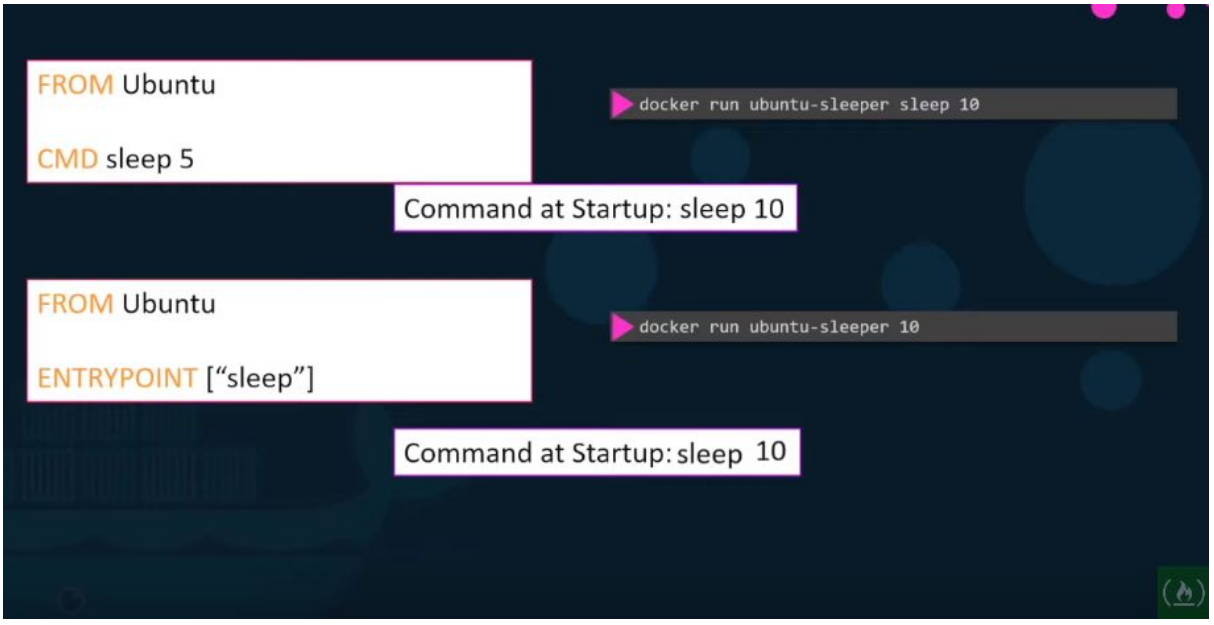


CMD

O comando passado substituirá completamente o comando de entrada da imagem

ENTRYPOINT

O parâmetro passado será agregado (appended) ao comando de entrada da imagem.



Podemos

FROM Ubuntu

ENTRYPOINT [“sleep”]

CMD[“5”]

Network

Default Networks

Bridge (Default)

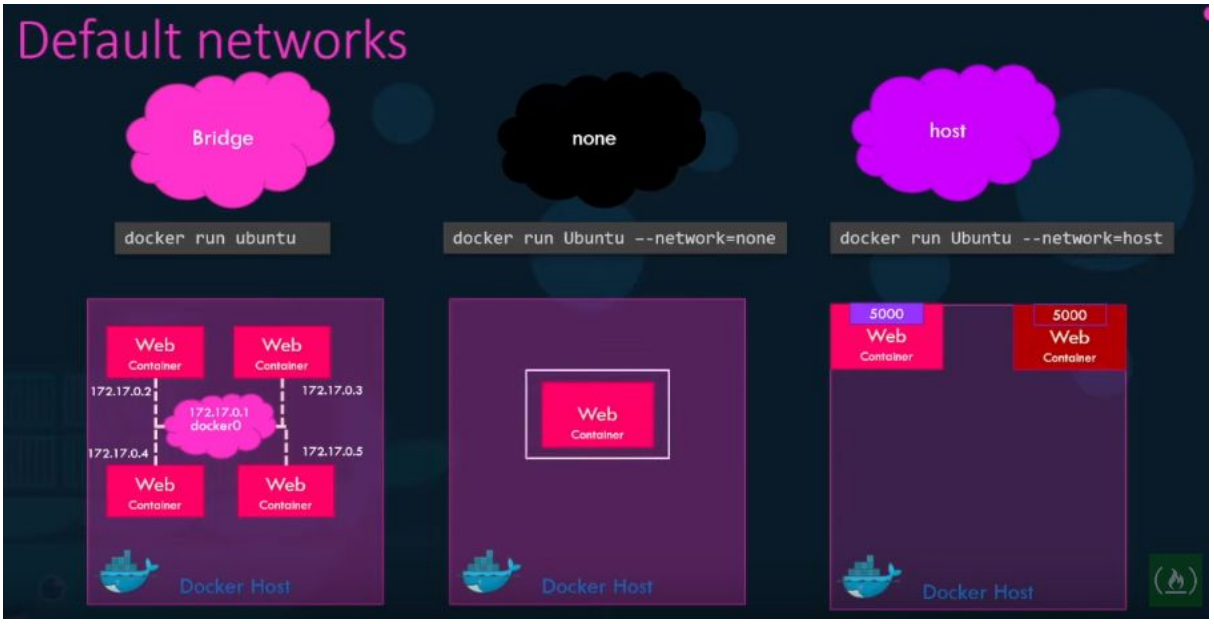
docker run ubuntu (Bridge)
Container podem acessar uns aos outros através de ips 172.17.0.2; 172.17.0.3 …
Para acessar os containers pelo host, devemos mapear as portas do container com uma porta do host.

none

Rodam isolados de qualquer network, nem entre containers nem com o host
docker run ubuntu --network=none

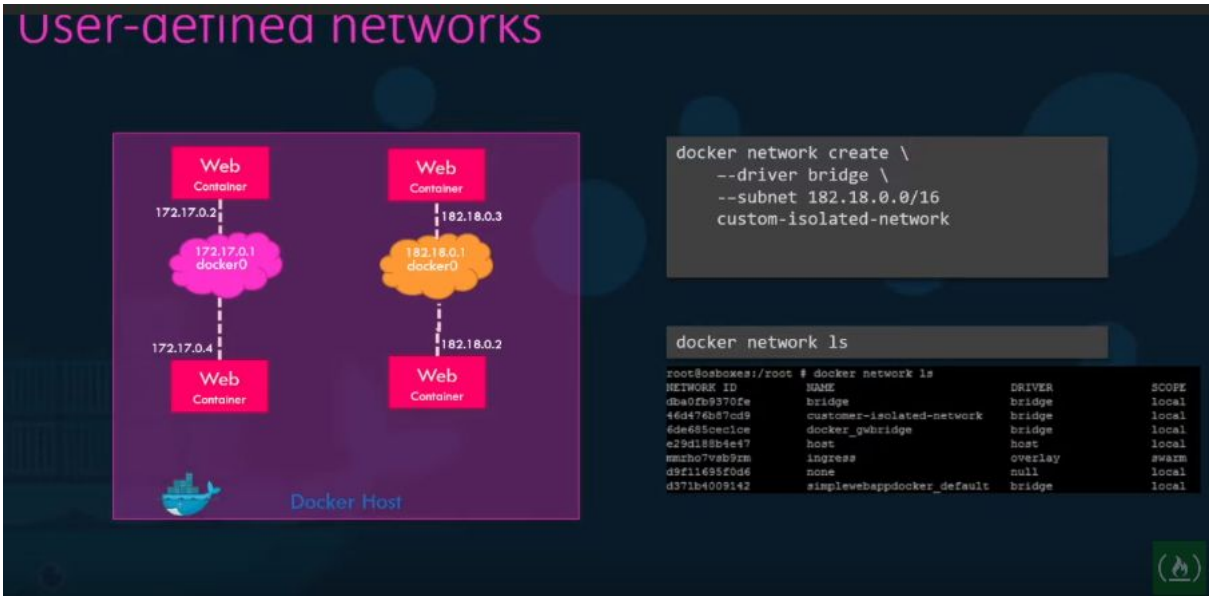
host

Não requer mapeamento de porta. Docker já roda diretamente nas portas do host.
docker run ubuntu --network=host

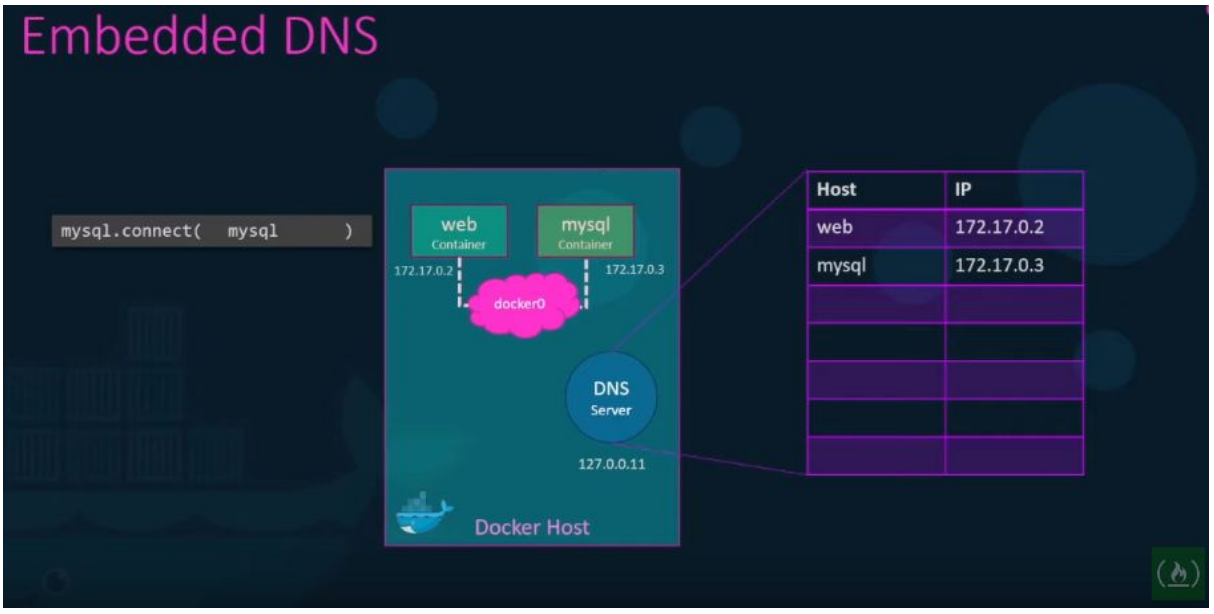


User-defined-networks

docker network create \
--driver bridge \
--subnet 182.18.0.0/16
custom-isolated-network



Embedded DNS 127.0.0.11
Qualquer container pode acessar outro container pelo seu nome


Embedded DNS
mysql.connect(mysql)
web
Container
mysql
Container
172.17.0.2
172.17.0.3
docker0
DNS
Server
127.0.0.11
Docker Host
Host
IP
web
172.17.0.2
mysql
172.17.0.3