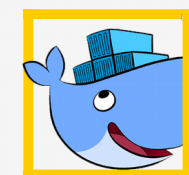


Começando com
Docker



<Diogo Alves>



- Arquiteto de Soluções e
Líder na iniciativa DevOps
da empresa Cedro



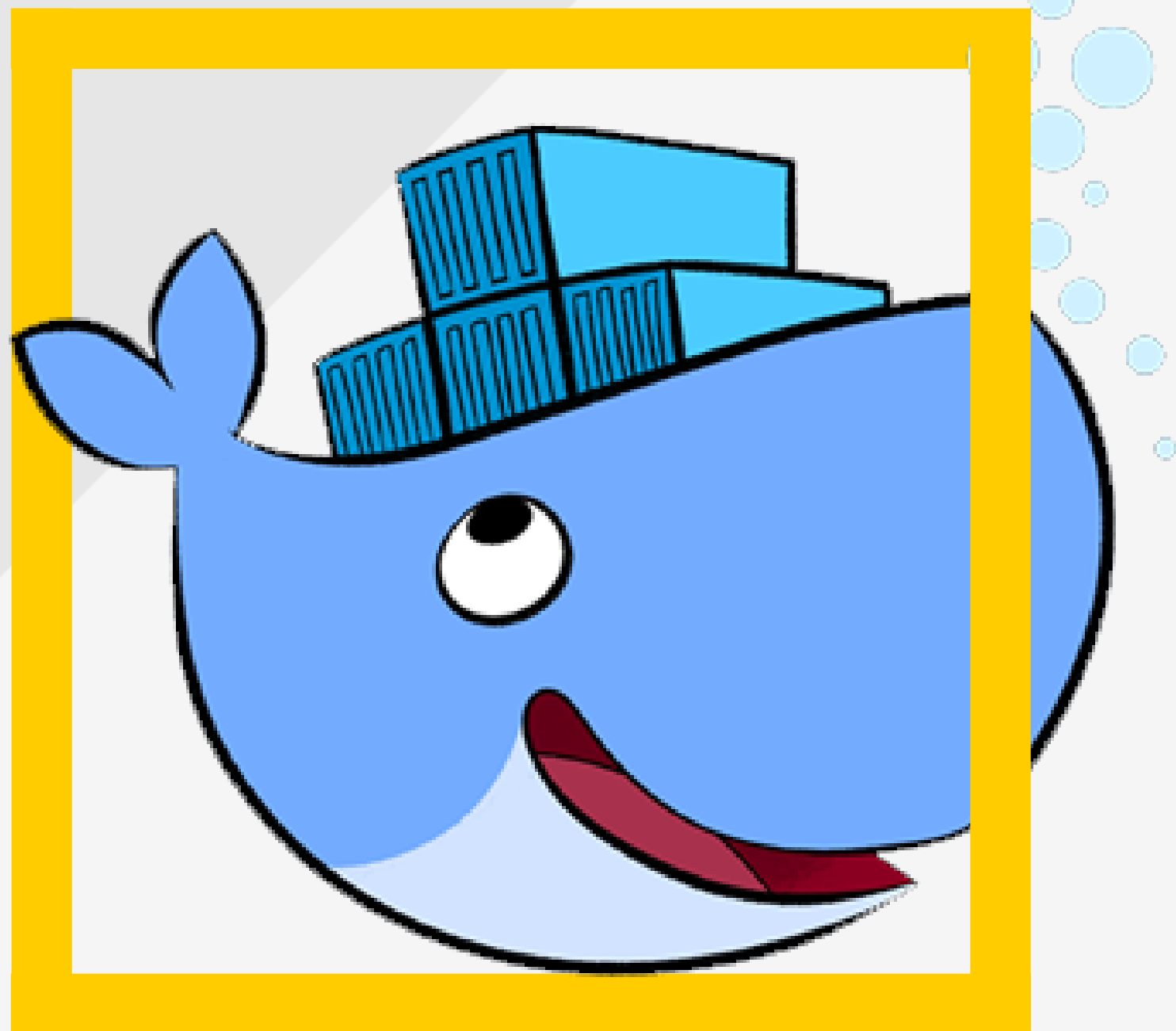
Formação:
- Sistemas de Informação



Pouco mais de 12 anos como:
- SysAdmin apaixonado por Linux
- Desenvolvedor
- Instrutor
- Entusiasta de Segurança da
Informação



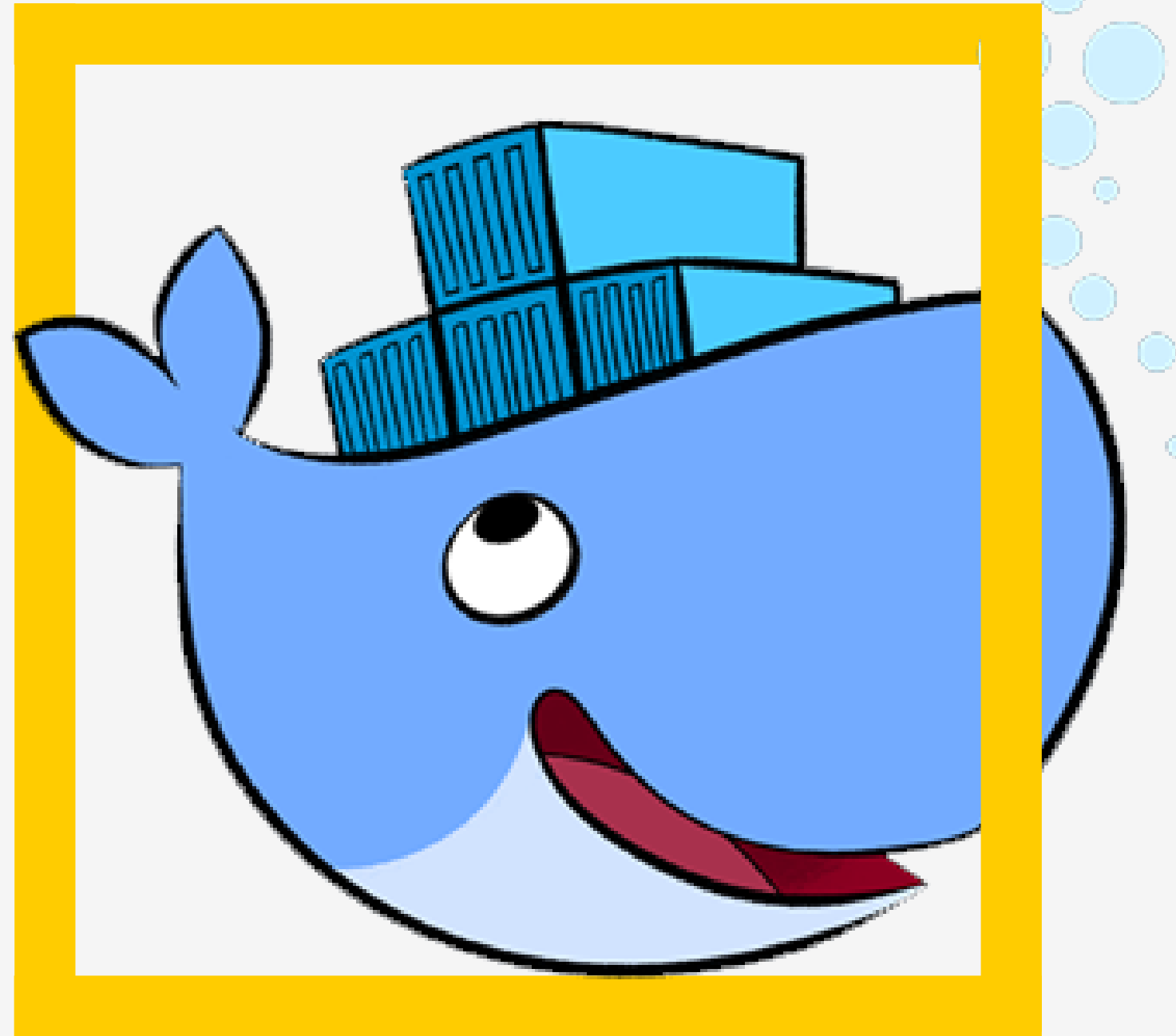
Algumas Certificações:
- CEH – Ethical Hacking
- ISO 27001
- Asterisk Administrator
- AWS



Agenda

- O que é DockerHub?
- Armazenamento no Docker
- Entendendo a rede no Docker
- Gerenciando containers com Docker-compose
- Hand-ons

Começando com
Docker



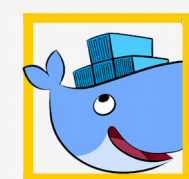
O que é
DockerHub?

<O que é DockerHub?>



Começando com Docker

- ✓ Repositório compartilhado de Images Docker
- ✓ Armazena imagens oficiais e não-oficiais
- ✓ Suporte a repositórios privados

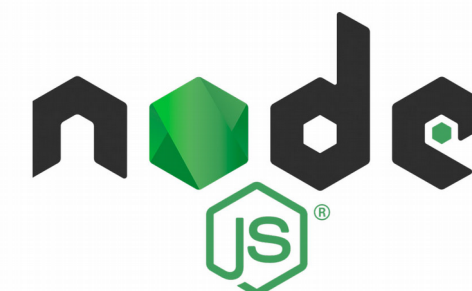


<O que é DockerHub?>

Exemplos de empresas com Images Docker oficiais



NGINX





<O que é DockerHub?>

Exemplos de Images Docker não-oficiais

The screenshot shows the Docker Hub interface for the image `jrcs/letsencrypt-nginx-proxy-companion`. The page has a blue header with the Docker Hub logo, a search bar containing "nginx", and links for "Explore", "Pricing", "Sign In", and "Sign Up". Below the header, the image name is displayed with a star icon and "Pulls 10M+". The image is described as a "Container" by "jrcs", updated 2 months ago, and is a "LetsEncrypt container to use with nginx as proxy". The "Overview" tab is selected, showing a description: "letsencrypt-nginx-proxy-companion is a lightweight companion container for nginx-proxy. It handles the automated creation, renewal and use of Let's Encrypt certificates for proxied Docker containers. Please note that letsencrypt-nginx-proxy-companion does not work with ACME v2 endpoints yet." Below this, a "Features:" section lists several bullet points. On the right, there are sections for "Docker Pull Command" (showing `docker pull jrcs/letsencrypt-nginx-proxy-companion`), "Owner" (jrcs), and "Source Repository" (Github link: `JrCs/docker-letsencrypt-nginx-proxy-companion`).

build error release v1.10 32.3MB 12 layers docker stars 507 docker pulls 98M

letsencrypt-nginx-proxy-companion is a lightweight companion container for **nginx-proxy**.

It handles the automated creation, renewal and use of Let's Encrypt certificates for proxied Docker containers.

Please note that `letsencrypt-nginx-proxy-companion` does not work with ACME v2 endpoints yet.

Features:

- Automated creation/renewal of Let's Encrypt (or other ACME CAs) certificates using **simp_le**.
- Let's Encrypt / ACME domain validation through `http-01` challenge only.
- Automated update and reload of nginx config on certificate creation/renewal.
- Support creation of Multi-Domain (SAN) Certificates.
- Creation of a Strong Diffie-Hellman Group at startup.
- Work with all versions of docker.

Docker Pull Command

```
docker pull jrcs/letsencrypt-nginx-proxy-companion
```

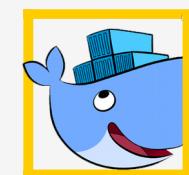
Owner

jrcs

Source Repository


Github
[JrCs/docker-letsencrypt-nginx-proxy-companion](https://github.com/JrCs/docker-letsencrypt-nginx-proxy-companion)

<https://hub.docker.com/r/jrcs/letsencrypt-nginx-proxy-companion>



<O que é DockerHub?>

Criando seu docker ID



Docker Identification


In order to get you started, let us get you a Docker ID.
Already have an account? [Sign In](#)

! Docker ID is required.

☐ I agree to Docker's [Terms of Service](#).

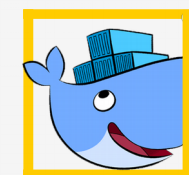
☐ I agree to Docker's [Privacy Policy](#) and [Data Processing Terms](#).

☐ (Optional) I would like to receive email updates from Docker, including its various services and products.

☐ I'm not a robot 

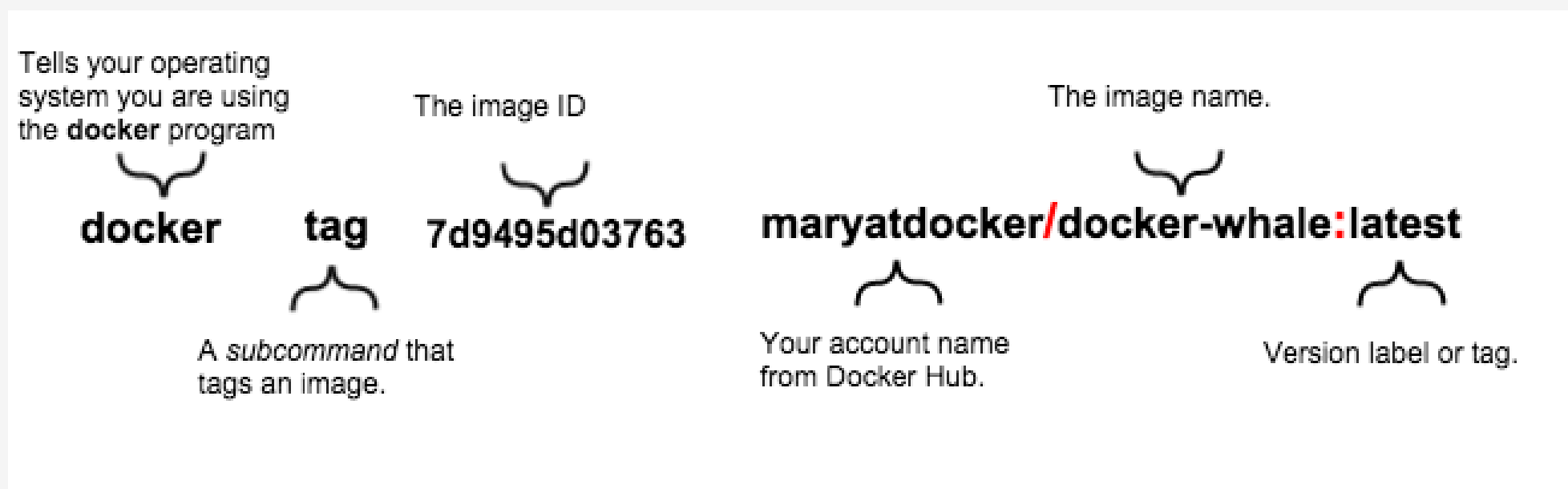
[Continue](#)

<https://hub.docker.com/signup>



<O que é DockerHub?>

Taggeando sua imagem



<O que é DockerHub?>

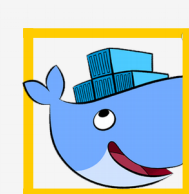


Começando com Docker

> No terminal execute

```
$ docker build . -t nome-da-image
```

```
$ docker tag id-da-imagem SEU_DOCKER_ID/nome-da-  
imagem:latest
```



<O que é DockerHub?>

> Push

```
$ docker login
```

```
$ docker image push dockerID/nome-da-imagem
```

> Pull

```
$ docker image pull dockerID/nome-da-imagem
```

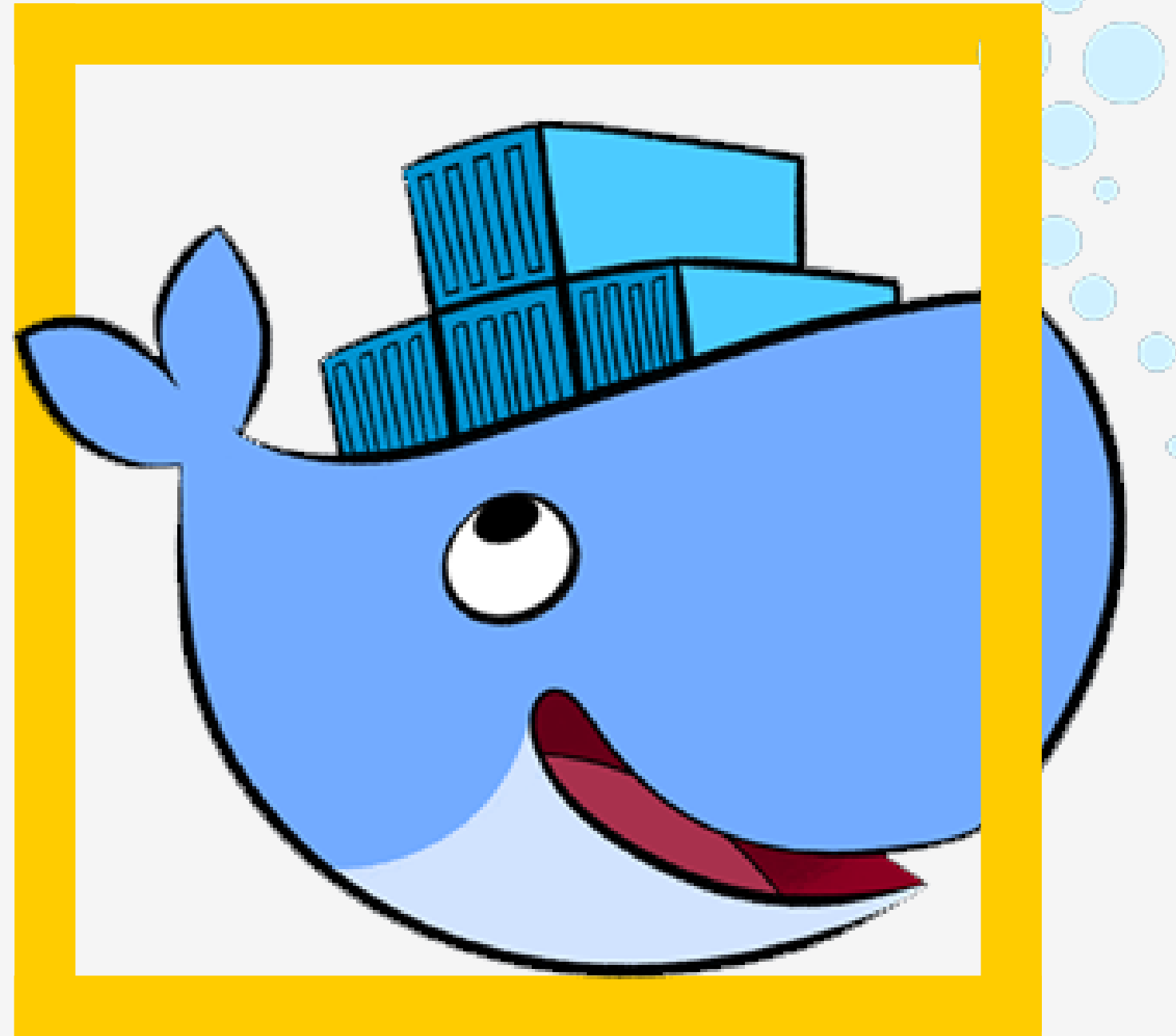
<O que é DockerHub?>



Começando com Docker

> Buscando imagens

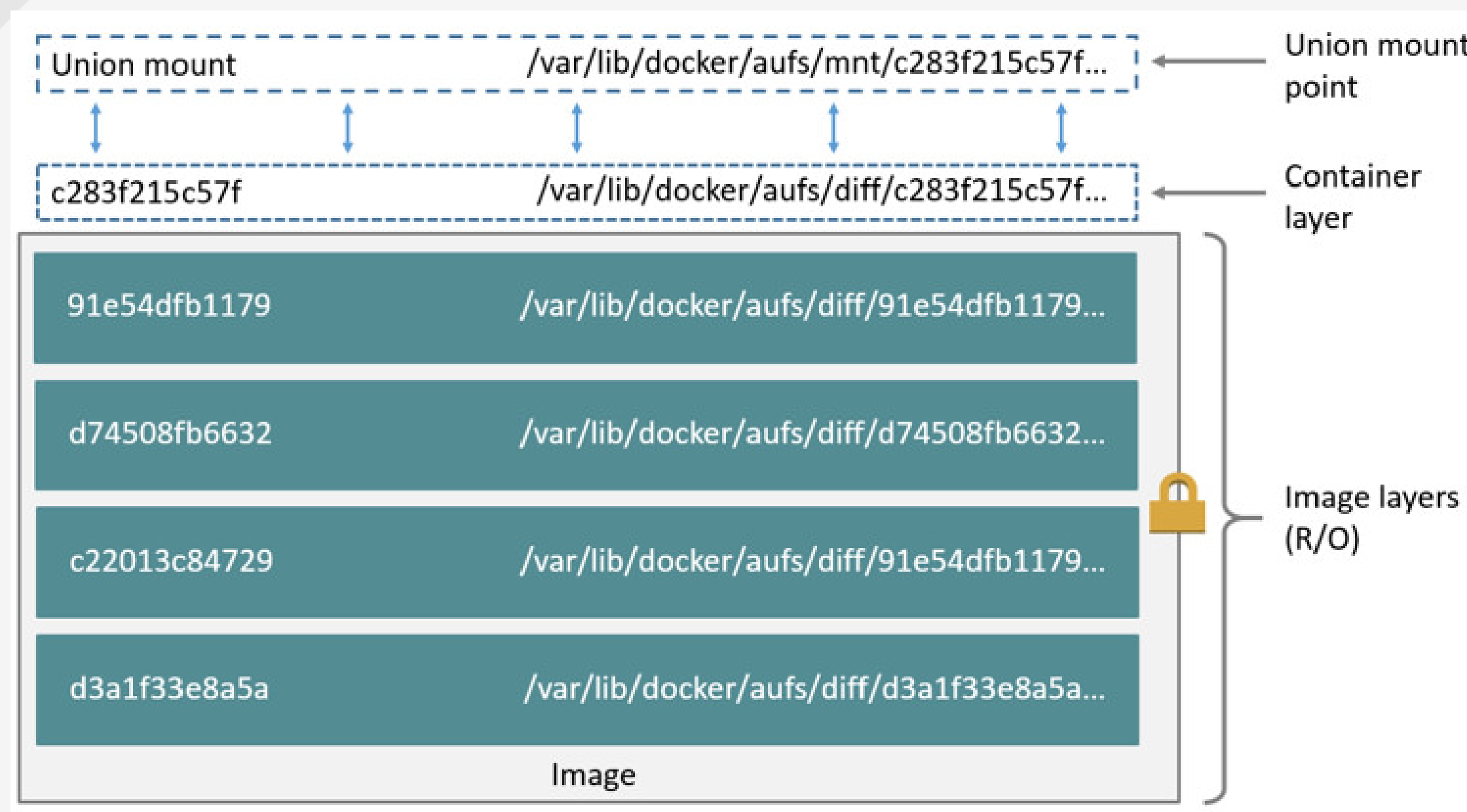
```
$ docker search nome-da-imagem
```

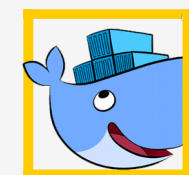


Armanzenamento no
Docker

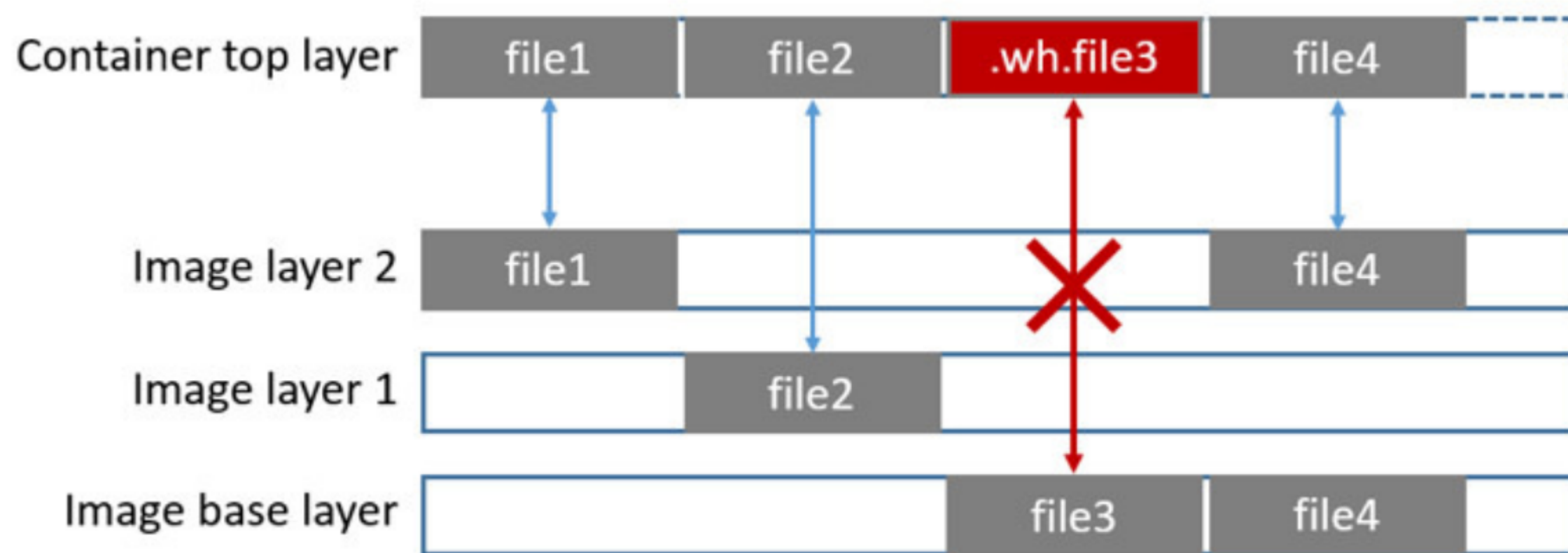


<Armazenamento no Docker>

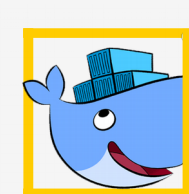




<Armazenamento no Docker>



Docker container
(AUFS storage-driver demonstrating whiteout file)



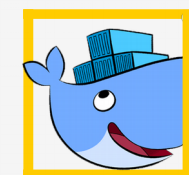
<Armazenamento no Docker>

> Utilizando Volumes

```
$ mkdir /home/user/container1
```

```
$ docker container run -it -v
```

```
/home/user/container1:/home/user/container1 ubuntu
```

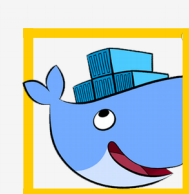



<Armazenamento no Docker>

> Mapeamento via container de dados

```
$ docker create -v /home/user/dbdata --name dbdata  
postgres /bin/true
```

```
$ docker container run -d --volumes-from dbdata  
--name db2 postgres
```

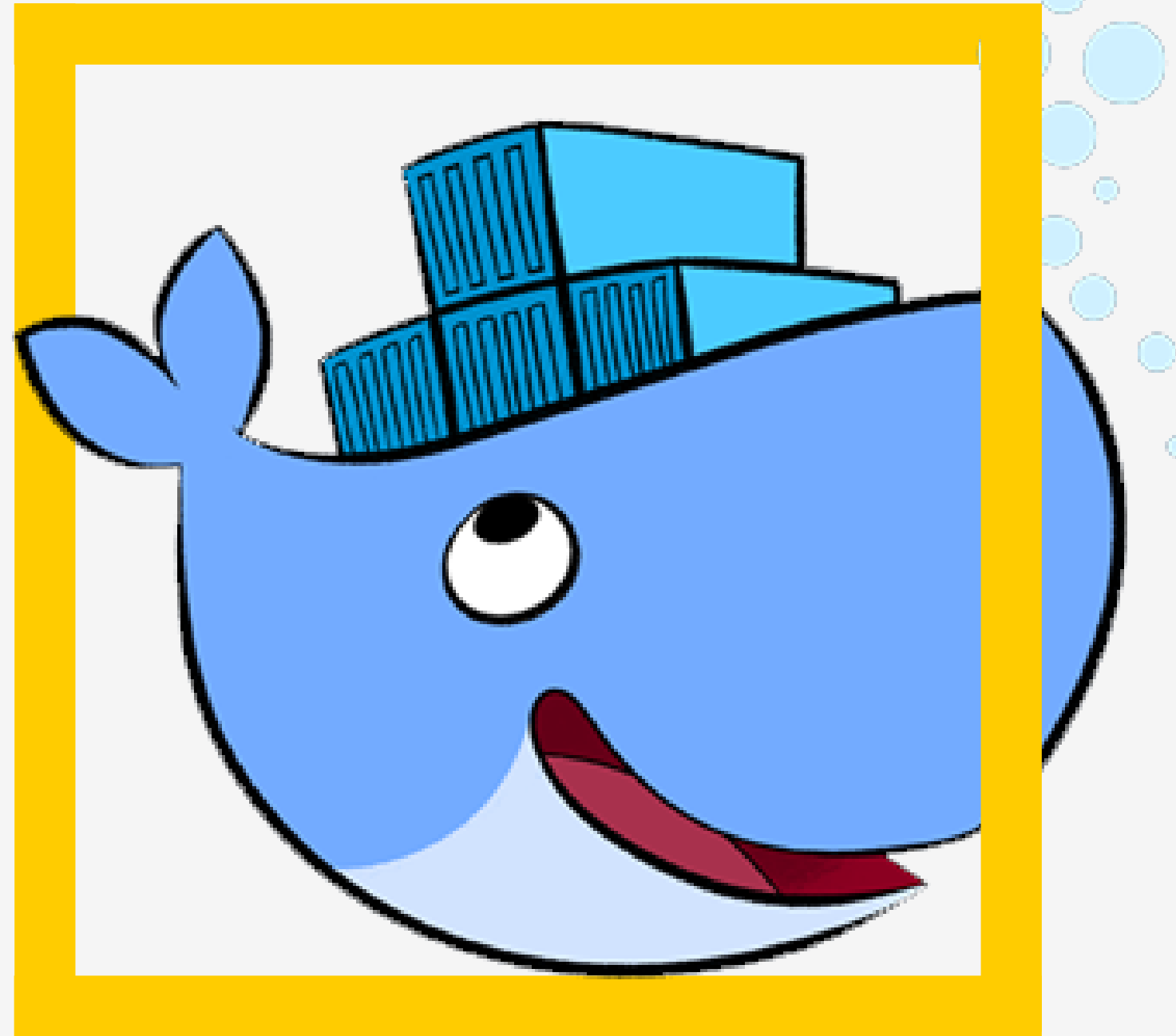


<Armazenamento no Docker>

> Mapeamento de volumes

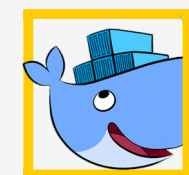
```
$ docker volume create --name dbdata
```

```
$ docker container run -d -v dbdata:/var/lib/data  
postgres
```



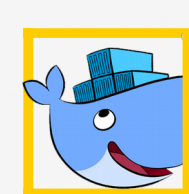
Entendendo a rede no

Docker



< Entendendo a rede Docker >

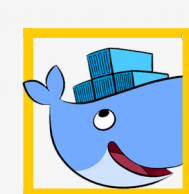
- > Redes Bridge
- > Redes None
- > Redes Overlay
- > Redes Usuários



< Entendendo a rede Docker >

> Redes Docker

```
$ docker network ls
```



<Entendendo a rede Docker>

> Exemplo

```
$ docker container run -d --name db -e  
MYSQL_ROOT_PASSWORD=minhasenha mysql
```

```
$ docker container run -d -p 80:80 --name app  
--link db tutum/apache-php
```

```
$ docker container exec -it app ping db
```

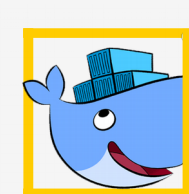


<Entendendo a rede Docker>

> Exemplo

```
$ docker network create --driver bridge rede-  
isolada
```

```
$ docker container run -itd --net rede-isolada  
alpine sh
```



< Entendendo a rede Docker >

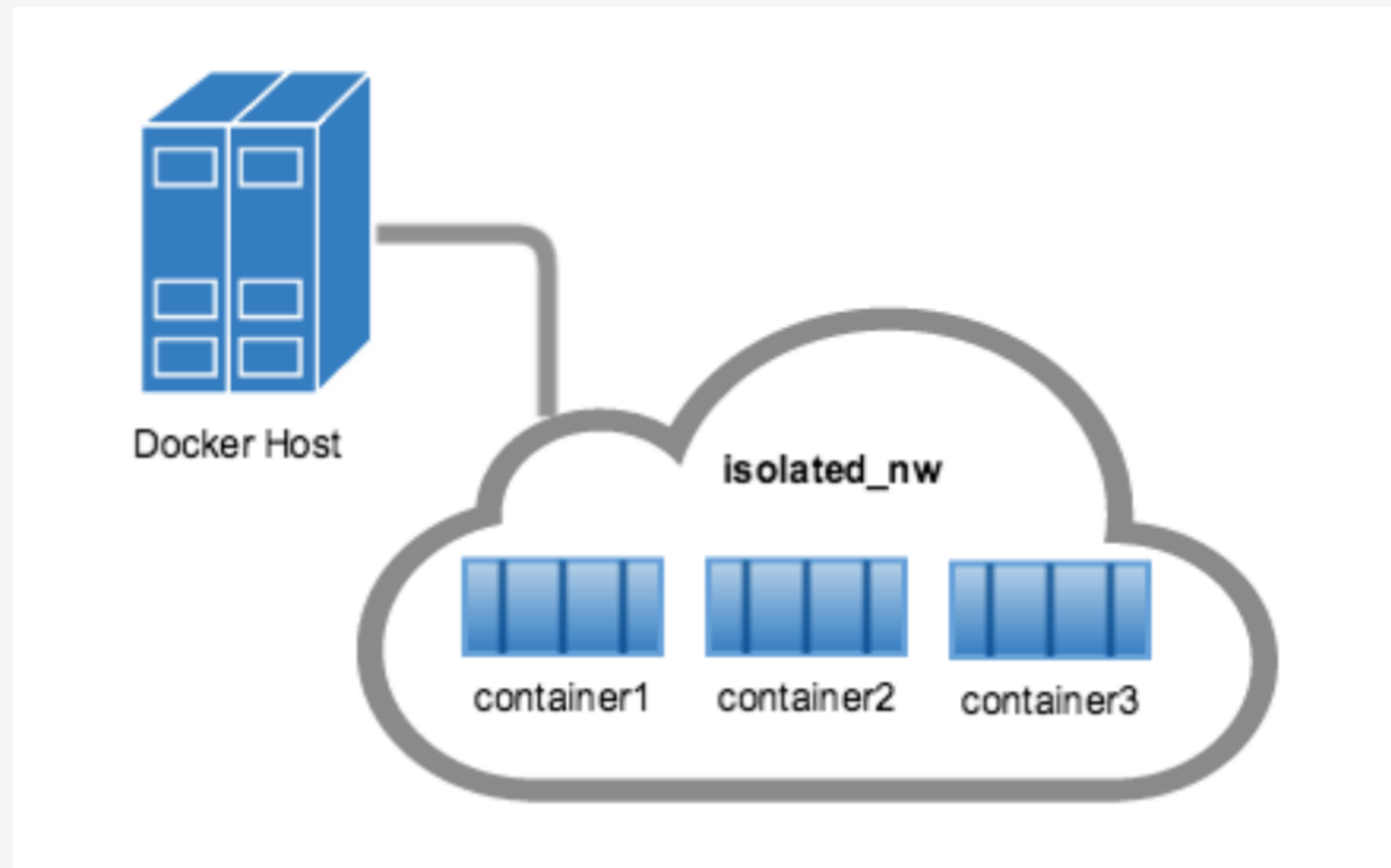
> Inspeccionado redes Docker

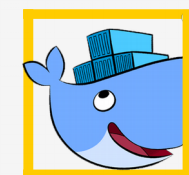
```
$ docker network inspect rede
```




< Entendendo a rede Docker >

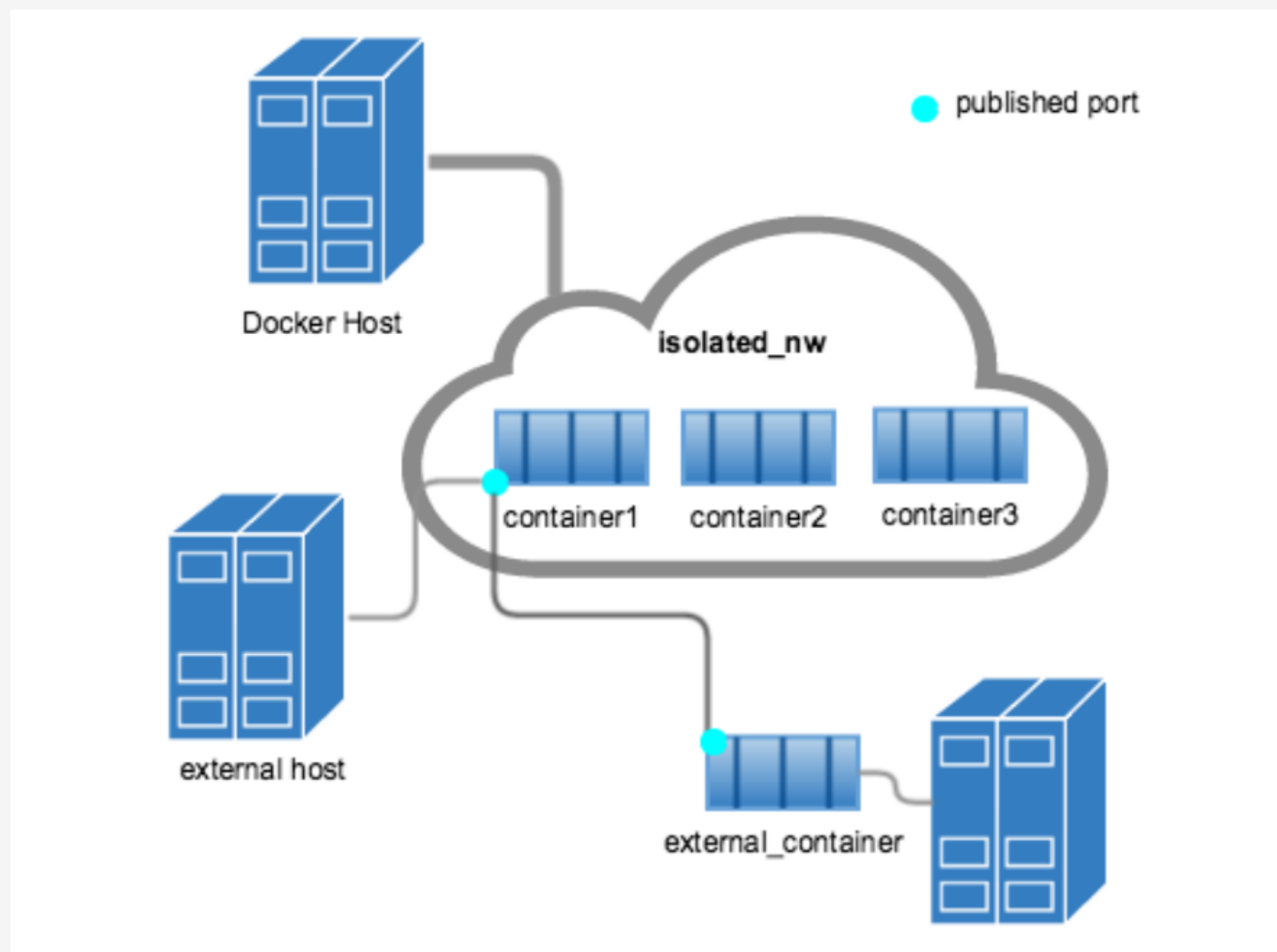
> Redes Isoladas

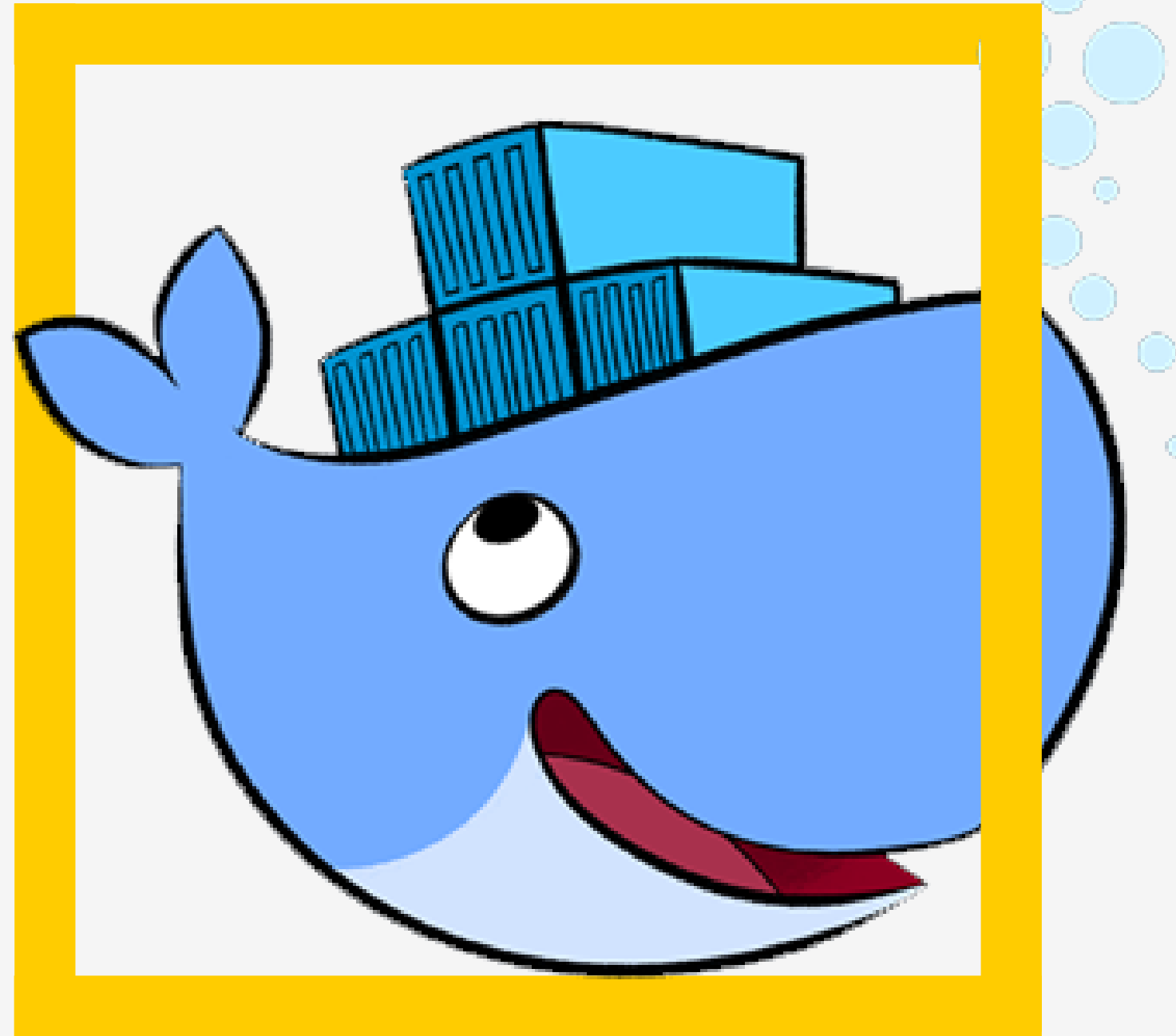




< Entendendo a rede Docker >

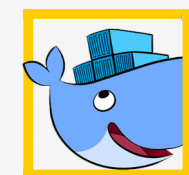
> Redes Overlay





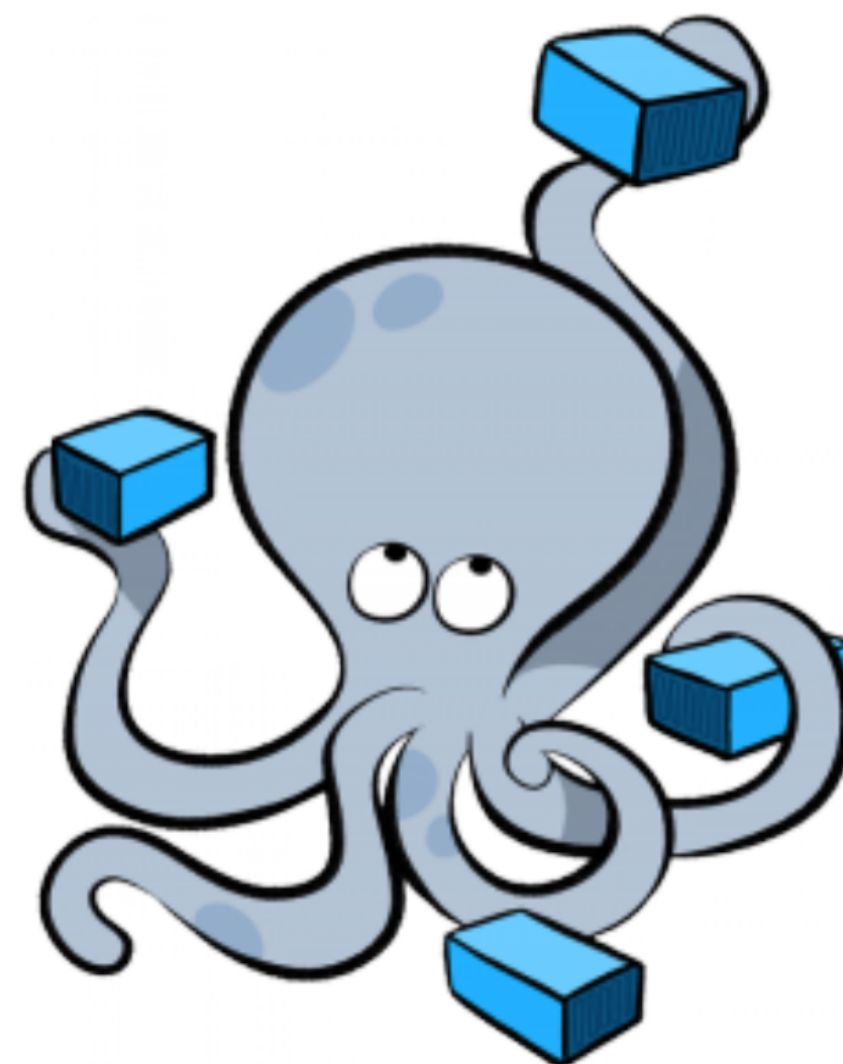
Gerenciando containers com

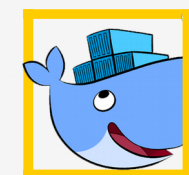
Docker-Compose



<Gerenciando containers com Docker-Compose>

```
version: '3'
services:
  web:
    build:
      context: .
      dockerfile: Dockerfile
    args:
      versao: 1
    ports:
      - "5000:5000"
  redis:
    image: redis
```

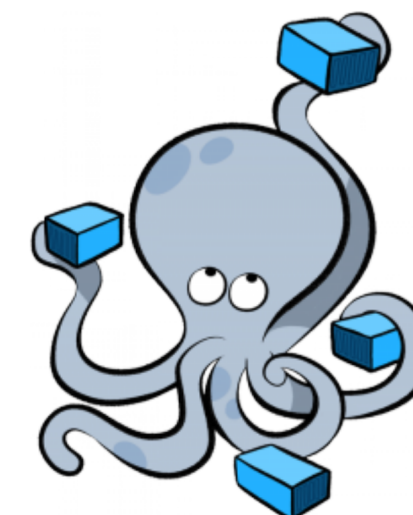




<Gerenciando containers com Docker-Compose>

> Instalando o Docker-compose

```
$ sudo curl -L  
"https://github.com/docker/compose/releases/download/  
1.24.0/docker-compose-$(uname -s)-$(uname -m)" -o  
/usr/local/bin/docker-compose  
  
$ sudo chmod +x /usr/local/bin/docker-compose
```



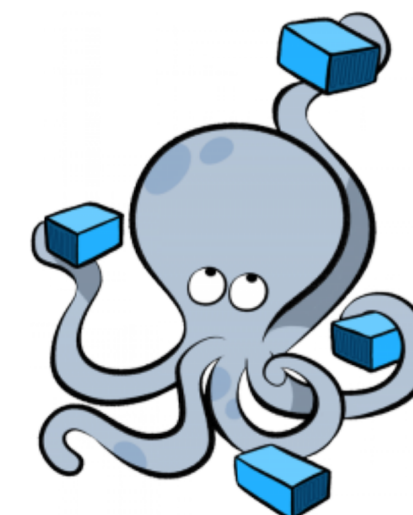


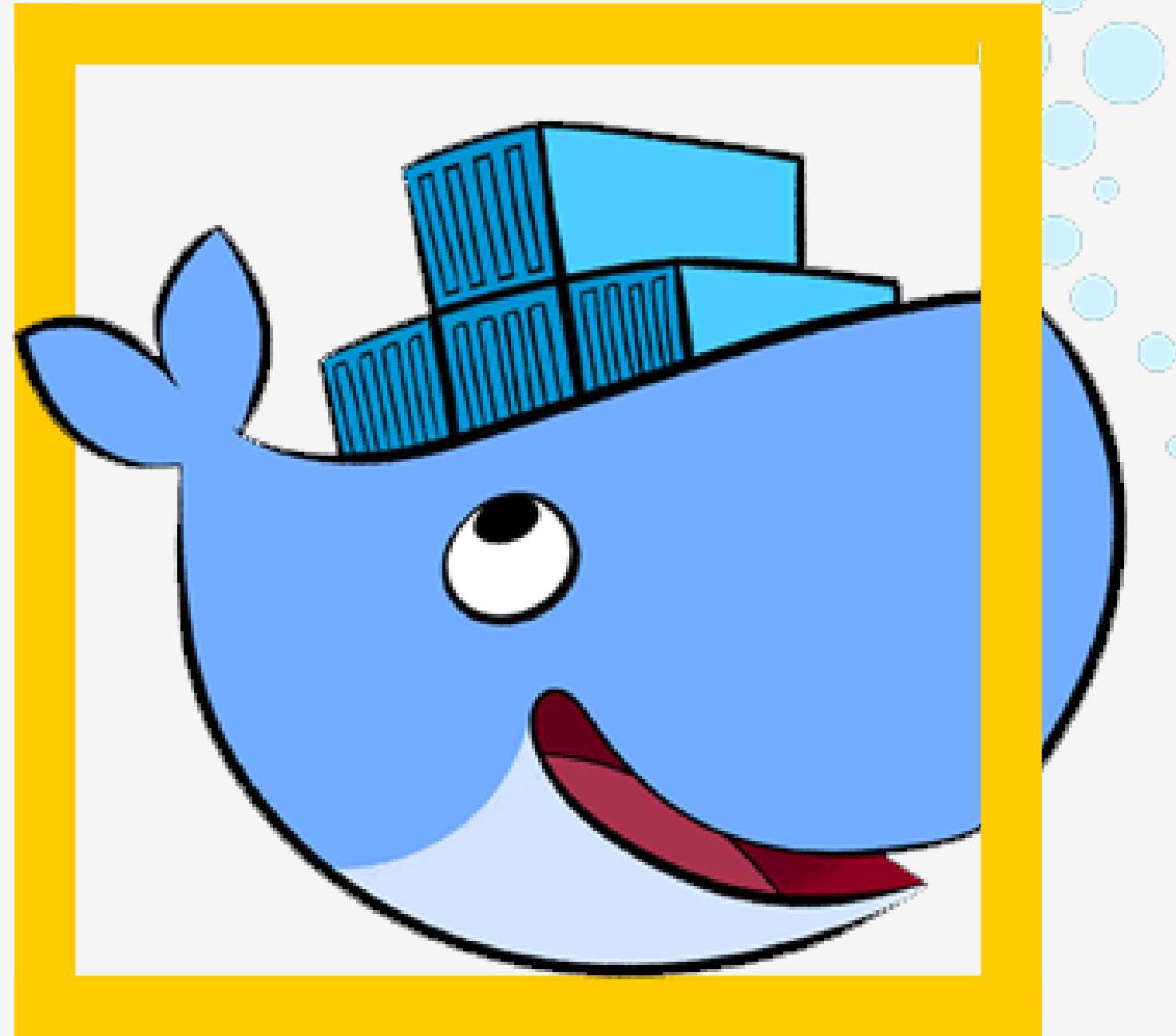
<Gerenciando containers com Docker-Compose>

> Executando o Docker-compose

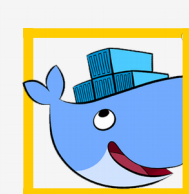
```
$ docker-compose -d up
```

```
$ docker-compose down
```





Hand-ons



<Hand-ons>

>Vamos lá?

\$ Criar um docker-compose com aplicação
Nginx expondo a porta 8080 para o host e
porta 80 para container

\$ Rede segregada

\$ persistir as configurações do Nginx em
volume /home/user/nginx



Obrigado



<https://www.linkedin.com/in/diogoalvesbarbosa/>
<https://www.instagram.com/diogo.alvesoficial/>
diogo.alves.barbosa@gmail.com