



# **Curso**

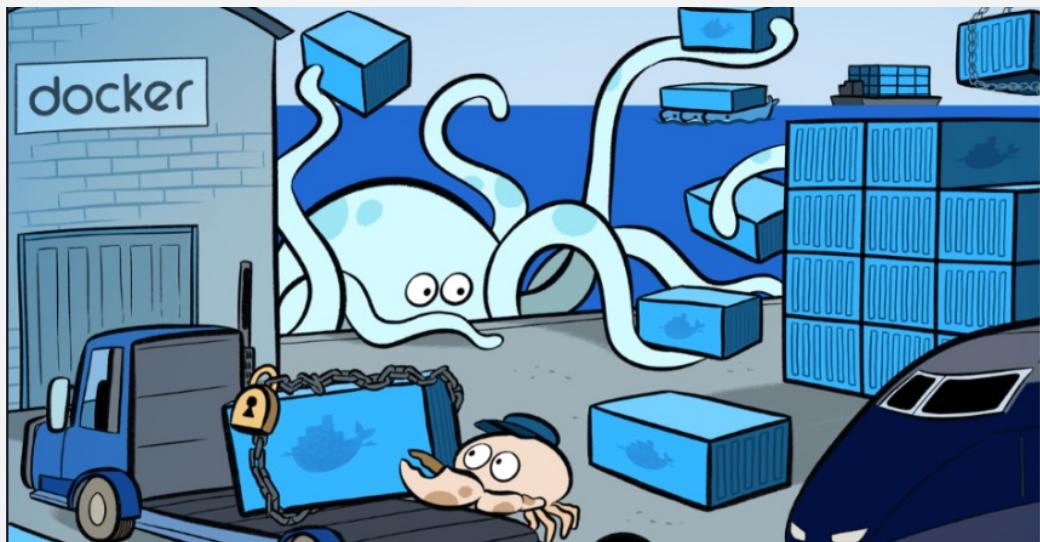
# **Qualidade de software**



# Docker



# O que é Docker?



- Estivador?
- Não é máquina virtual;
- Plataforma, open-source, que permite desenvolver, enviar e executar aplicativos como **containers**;
- Desenvolvido em GO LANG;
- Utiliza o LXC (Linux containers), um subsistema do Kernel do Linux.



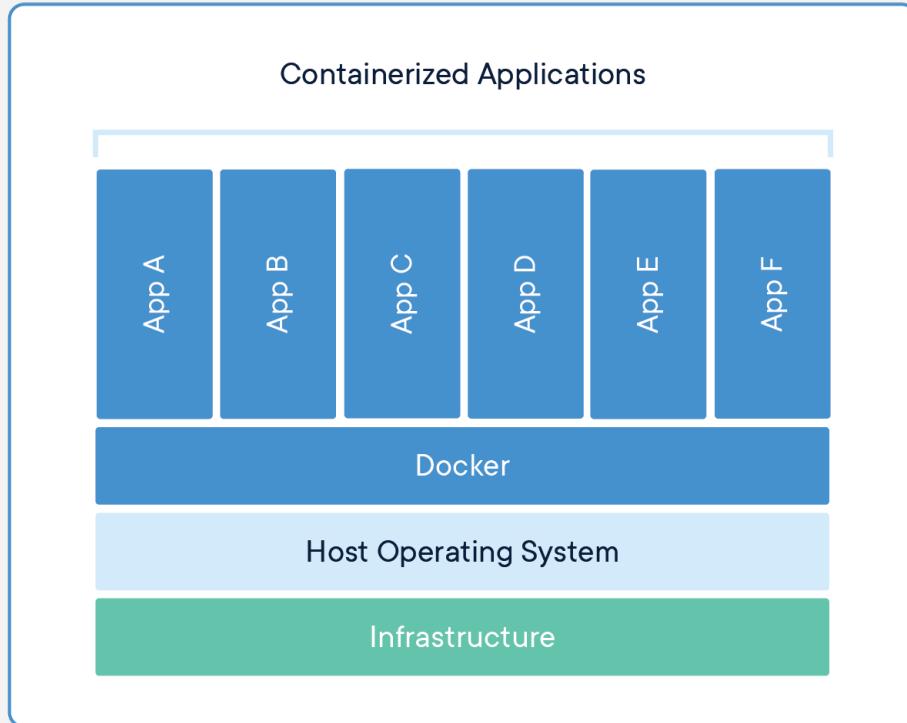
# Por que Docker?

- Padronização e replicação de ambientes;
- Ambientes de testes integrados;
- Simples implementação e fácil manutenção;
- Economia de recursos;
- Comunidade Docker ativa.

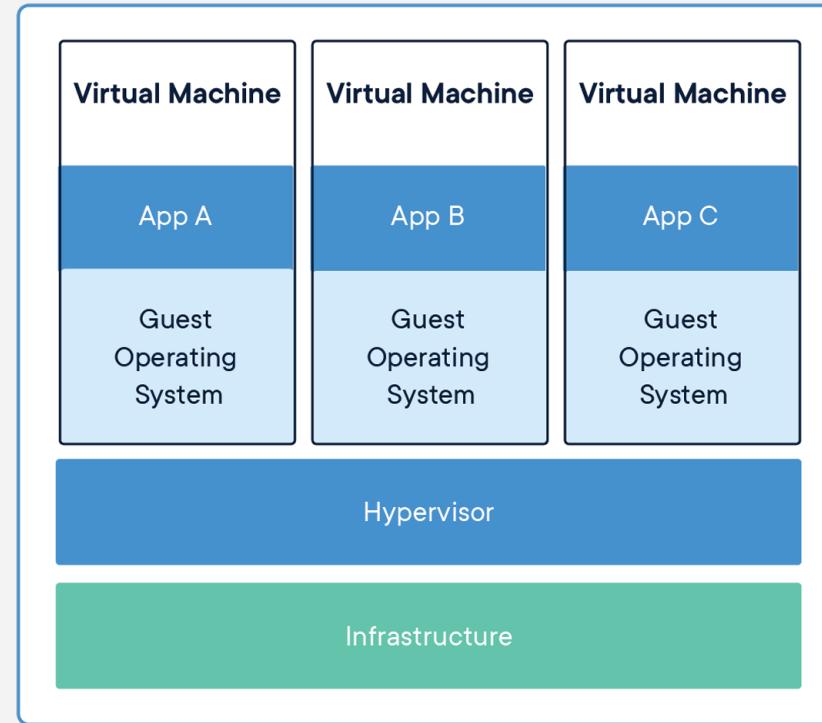
**Na minha  
máquina funciona!**



# Containers e Máquinas Virtuais

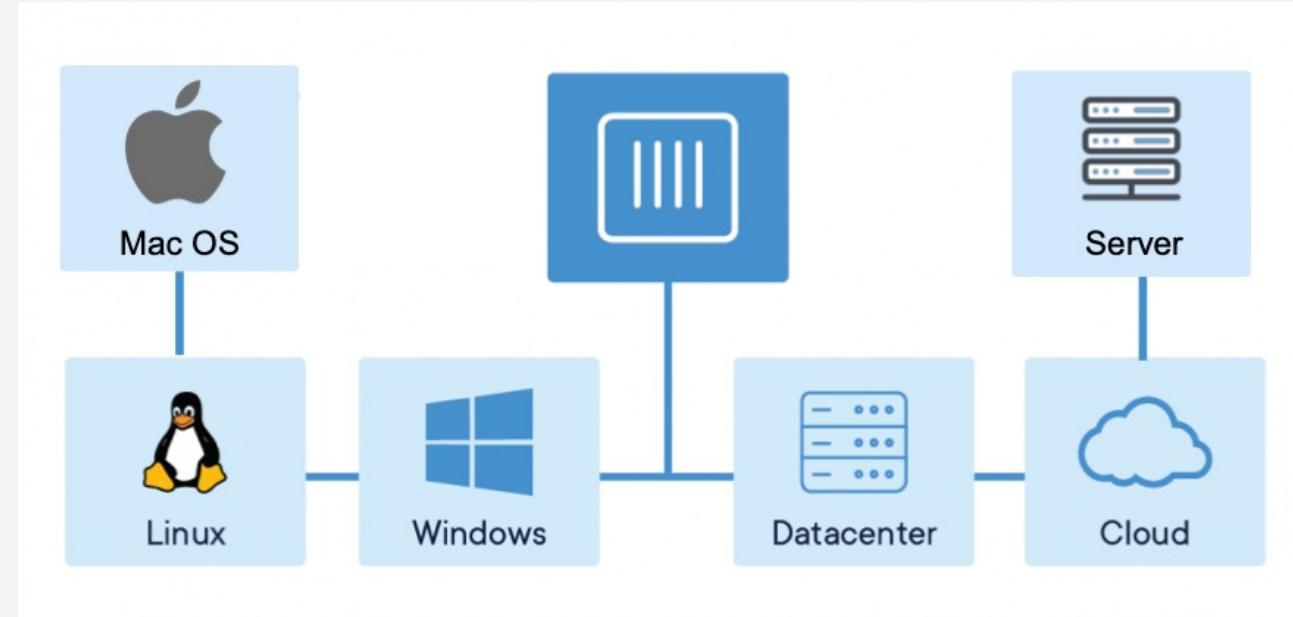


Os contêineres são uma abstração na camada do aplicativo que agrupa o código e as dependências;



As máquinas virtuais (VMs) são uma abstração do hardware físico que transforma um servidor em vários servidores.

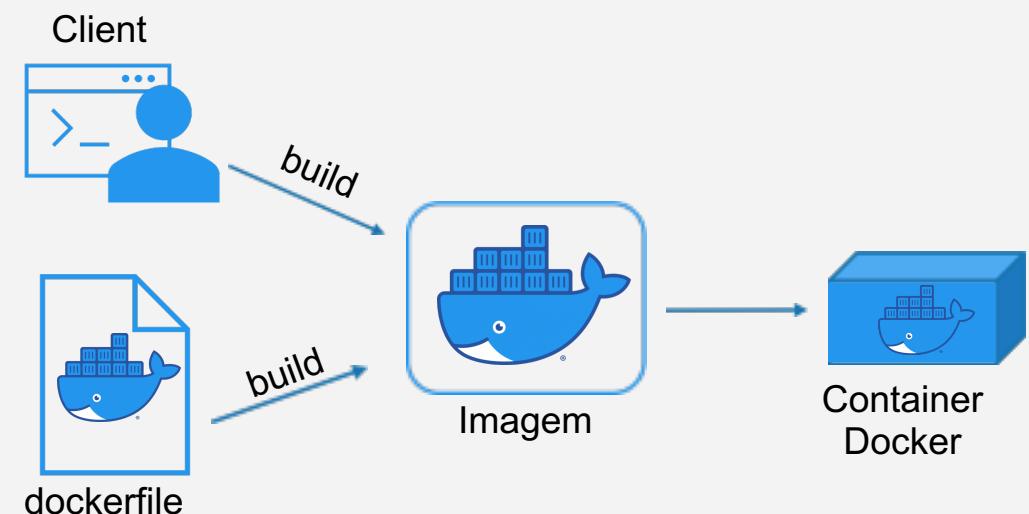
# Distribuição de containers



Os contêineres docker estão em toda parte: linux, mac os, windows, data center, nuvem, servidor, etc.

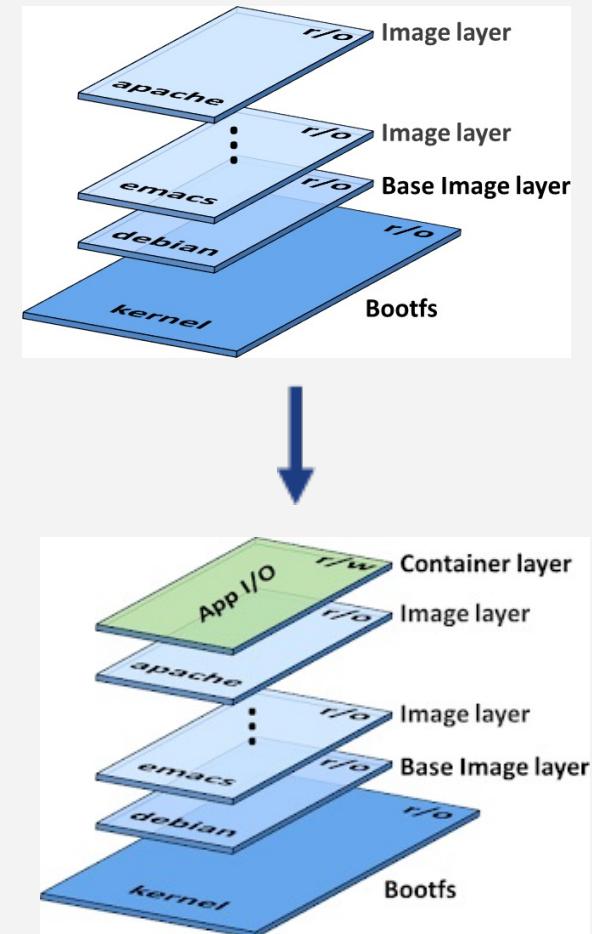
# Imagens & Containers

- **Imagen** contem informações sobre o que é necessário para criar container. Normalmente contem uma união de sistemas de arquivos (**AUFS**) em camadas empilhados (**layers**) uns sobre os outros.
- **Container** é uma instância de tempo de execução de uma imagem docker.
- Você pode criar uma imagem docker de duas maneiras:
  - método interativo
  - método dockerfile

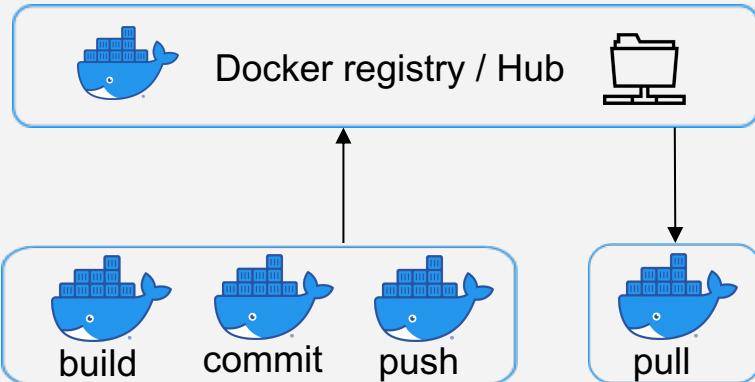


# Dockerfile & Layers (Camadas)

```
ex-build-go > 🚀 Dockerfile > ...
1  FROM golang:1.11-alpine AS build
2
3  RUN apk add --no-cache git
4  RUN go get github.com/golang/dep/cmd/dep
5
6  COPY Gopkg.lock Gopkg.toml /go/src/project/
7  WORKDIR /go/src/project/
8
9  RUN dep ensure -vendor-only
10
11 COPY . /go/src/project/
12 RUN go build -o /bin/project
13
14 FROM scratch
15 COPY --from=build /bin/project /bin/project
16 ENTRYPOINT ["/bin/project"]
17 CMD ["--help"]
18
```

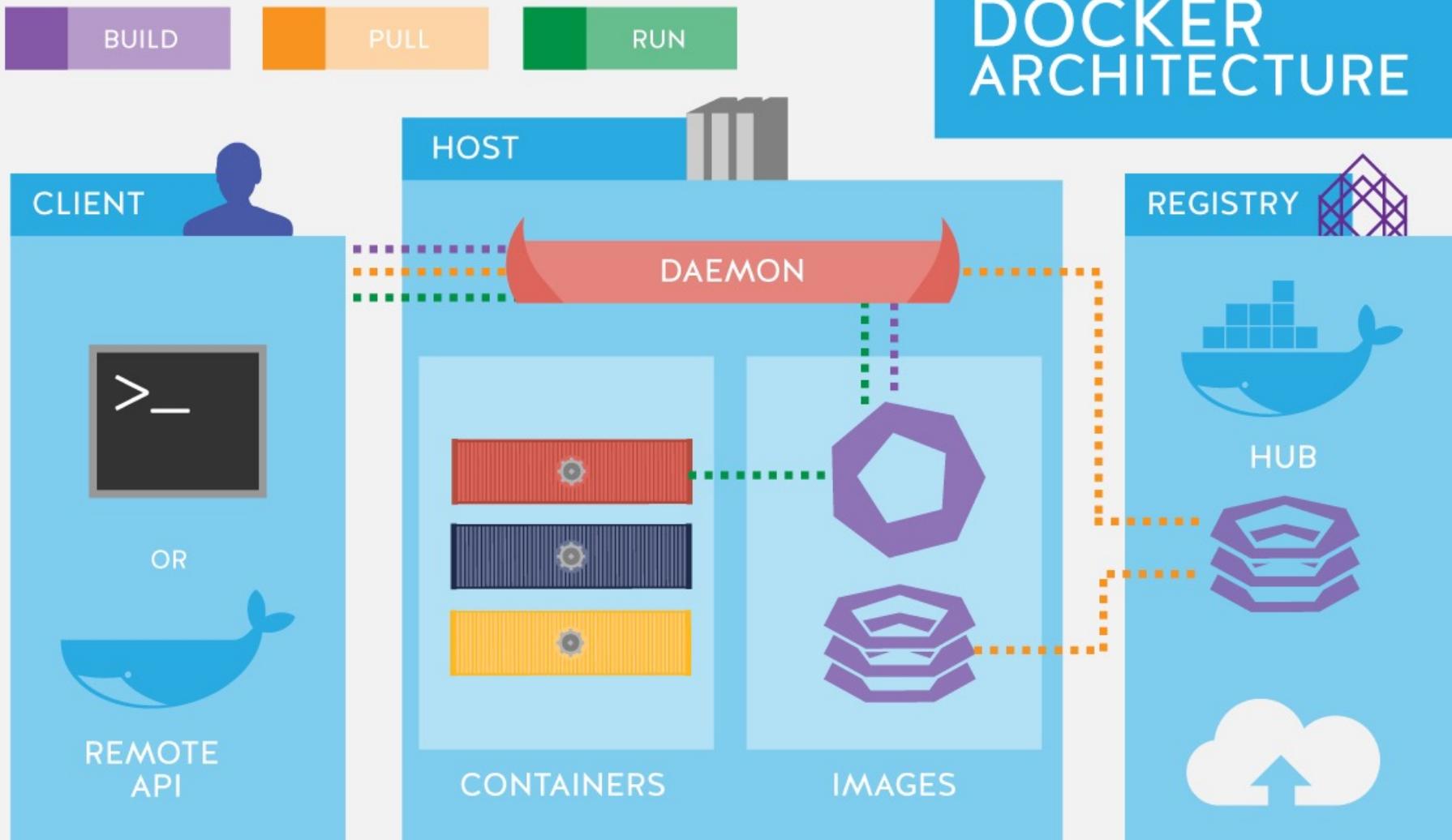


# Registry & Docker Hub



- O registry é um aplicativo do lado do servidor sem estado e altamente escalonável que armazena e permite distribuir imagens docker;
- Docker hub é o maior repositório do mundo de imagens e containers para armazenamento e compartilhamento de imagens;
- Empresas podem escolher um plano de assinatura para repositórios privados.

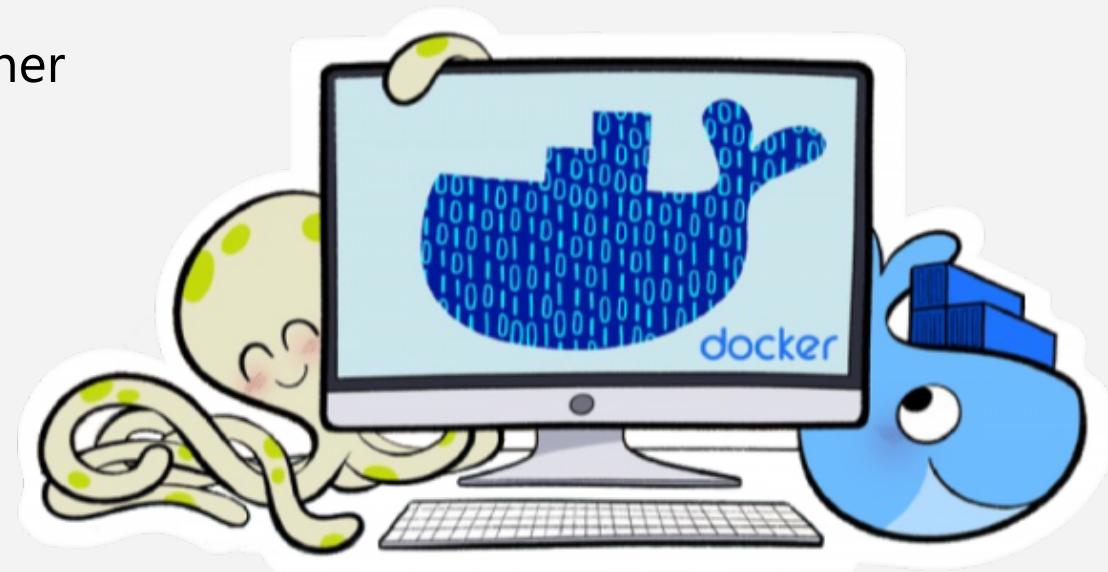
# DOCKER ARCHITECTURE



NORDICAPIS.COM

# Onde e como aprender?

- **Pwd** – (play with docker) é um playground do docker que permite aos usuários executar comandos do docker no navegador: <https://www.docker.com/play-with-docker>
- **Docker 101 tutorial** é um tutorial em um container onde aprende o essencial de docker: <https://www.docker.com/101-tutorial>
- **Docker docs** é uma documentação rica de conteúdos mantida pelo time do docker: <https://docs.docker.com/get-started/resources/>



# Comandos básicos

- **docker help** – busca comandos docker para te auxiliar
- **docker container ps** – lista os containers (somente os executados)
- **docker image ls** – lista as imagens criadas
- **docker container ps -a** – lista container (parados e executando)
- **docker pull <image>** - baixa uma imagem
- **docker run <image>** - executa uma imagem
- **docker stop <id\_container>** - para a execução de um container
- **docker search <nome\_imagem>** - busca uma imagem no Dockerhub
- **docker build** – constrói uma imagem no Dockerfile
- **docker stats <container>** - informações de uso do hardware
- **docker inspect <container>** - inspeciona o conteúdo estático do container
- **docker image prune** – limpa as imagens pendentes ou não referenciadas
- **docker rmi <id\_imagem>** - remove uma imagem do seu host
- **docker rm <id\_container>** - remove um container de seu host



# Parâmetros úteis

- **-a** - lista todas as imagens ou containers (mesmo parados)
- **-it** - entra em modo iterativo com o container
- **--name** - cria um nome para o container
- **-rm** - cria um container e remove após o uso
- **-p** – define a porta a ser usada
- **-d** - executa em modo background
- **-c** - executa um comando de console específico
- **-t** – cria um nome (tag) para o container



**Até a próxima aula!**

# Referências

- <https://www.docker.com/>
- <https://www.docker.com/play-with-docker>
- <https://docs.docker.com/samples/>
- <https://twitter.com/docker>
- <https://github.com/gomex/docker-para-desenvolvedores>
- <https://nordicapis.com/docker-containers-and-apis-a-brief-overview/>
- <https://docs.docker.com/get-started/overview/>
- <http://neokobo.blogspot.com/2017/03/docker-image.html>
- <https://mirandacontainer.com.br/historia-completa-containers/>

