



Afinal, o que é **docker**?

Luis Gustavo S. Barreto
<gustavosbarreto@gmail.com>





O que não é?

Vamos começar falando sobre o que Docker não é:

- ✗ Não é o nome da baleia do filme *free willy*
- ✗ Não é um sistema operacional
- ✗ Não é um software de emulação como VMWare ou VirtualBox
- ✗ Não tem nada a ver com Vagrant



Então, o que é?



É uma plataforma de containerização de software

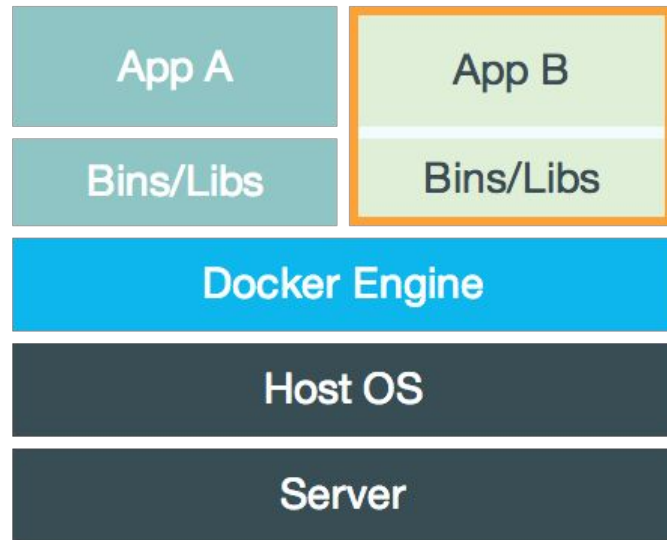
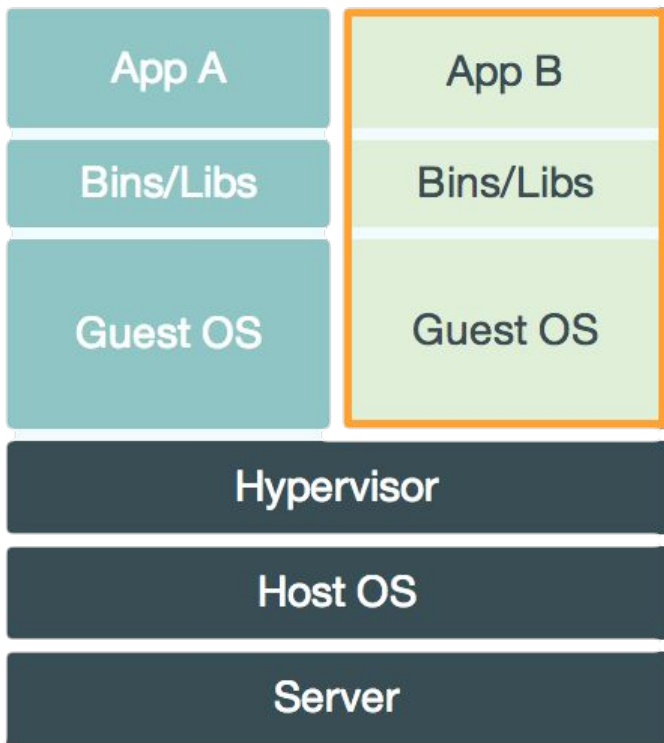


O que isso significa?




Significa que com Docker podemos criar e isolar um ambiente para rodar um software específico sem a necessidade de ter que emular um sistema operacional inteiro com uma máquina virtual



Containers vs Máquina Virtuais



Por que usar Docker?

-  **Segurança**
 - Isolamento entre as aplicações
 - Uma aplicação não pode acessar outra a menos que você autorize
-  **Desempenho**
 - Aproveitamento de recursos computacionais
 - Não é necessário gastar recursos computacionais com outros serviços
-  **Confiança**
 - Reprodutibilidade
 - Se funciona na minha, tem que funcionar na tua.



Principais Conceitos



Imagem

- A imagem é uma aplicação empacotada



Container

- Um container é basicamente uma instância de uma imagem



Dockerfile

- Um arquivo de texto que descreve passos para construir uma imagem

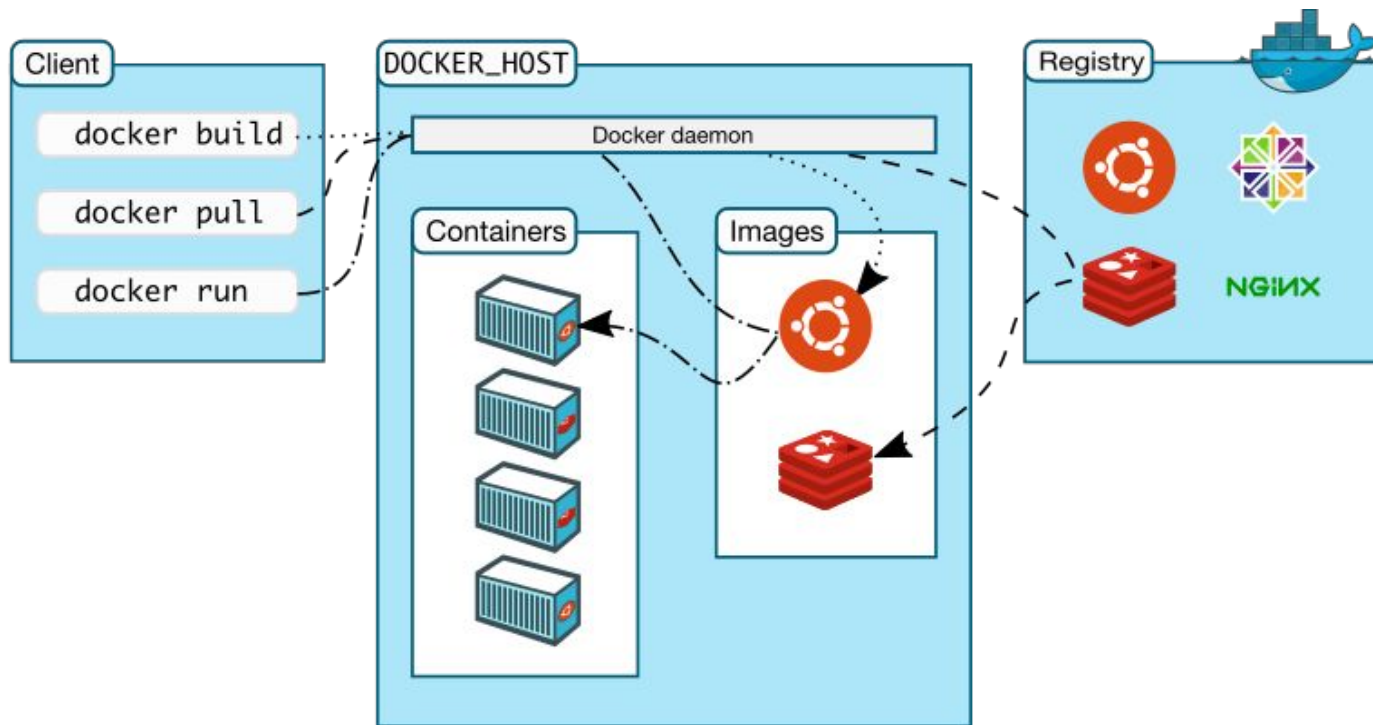


Registro

- Um servidor que hospeda imagens



Visão Geral



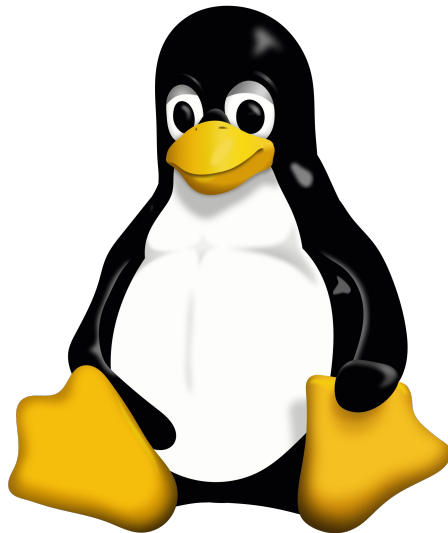
Instalação

O Docker é multiplataforma e pode ser instalado na maioria dos sistemas operacionais modernos como:

- Linux
- Windows
- Mac OS X



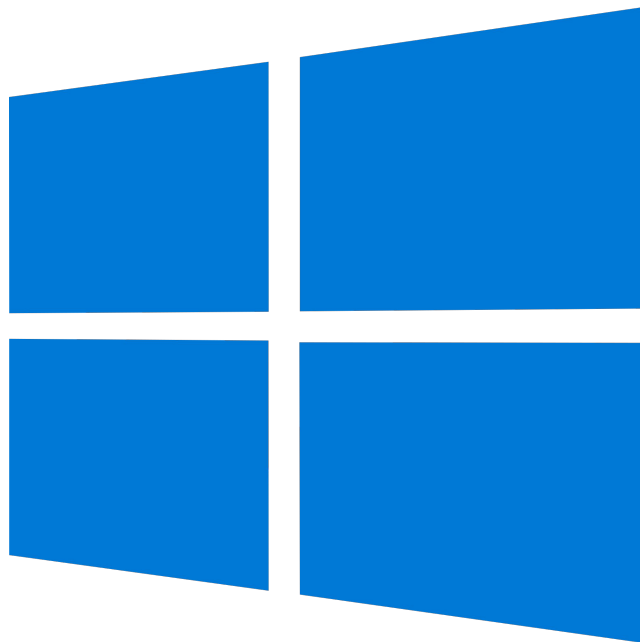
Linux



```
apt-get install docker.io
```



Windows





Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

25% complete



For more information about this issue and possible fixes, visit

<http://windows.com/stopcode>

If you call a support person, give them this info:
Stop code: CRITICAL_PROCESS_DIED

Mac OS X



Precisa vender um rim antes



Docker CLI

Ferramenta linha de comando para interagir com o docker. Suas duas principais funções são:

- Rodar imagens (containers)
 - `docker run`
- Buildar (construir) imagens
 - `docker build`
- Enviar imagens para o registro
 - `docker push`



Docker Hub: Registro

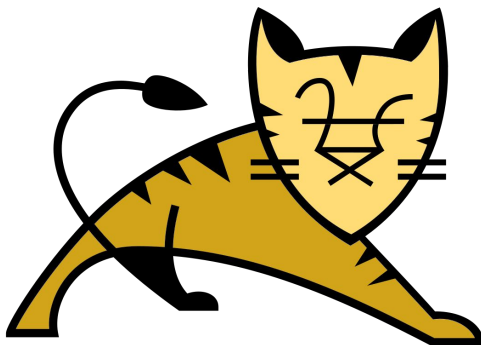
O Docker Hub é um servidor que armazena imagens docker



Apache



PostgreSQL



Exemplo: Servidor MySQL

```
docker run \  
  --name servidor-mysql \  
  -e MYSQL_ROOT_PASSWORD=senha \  
  -d mysql
```



Dockerfile

É um arquivo de texto que descreve passos para construir uma imagem



Dockerfile: Principais instruções

- **FROM**

- Informa a partir de qual imagem será gerada a nova imagem

- **RUN**

- Executa um comando durante o build da imagem

- **EXPOSE**

- Expõe uma porta em um container

- **ADD/COPY**

- Copia um arquivo/diretório pra dentro da imagem

- **ENTRYPOINT/CMD**

- Comando a ser executado ao iniciar o container



Dockerfile: Exemplo

```
FROM ubuntu:16.04
```

```
RUN apt-get update && \  
    apt-get install -y apache2
```

```
ADD www/* /var/www/html/
```

```
ENTRYPOINT ["/usr/sbin/apache2", "-k", "start"]
```



Dockerfile: *Buildando* e executando uma imagem

- **Buildando:**

- `docker build -t nome_imagem`

- **Executando:**

- `docker run nome_imagem`



Docker Compose

O Docker Compose é uma ferramenta para a criação e execução de múltiplos containers de aplicação. Com o Compose, você usa um arquivo do tipo YAML para definir como será o ambiente de sua aplicação e usando um único comando você criará e iniciará todos os serviços definidos.



Docker Compose: Exemplo

```
services:
  web:
    build: ./nodeapp
    ports:
      - '5000:5000'
    links:
      - mysql

mysql:
  image: mysql
```



Demonstração

Mais informações

<http://docs.docker.com>



É isso!
Perguntas?