

DOCKER

Container Paths Explained



O que é o Docker?

O Docker é um sistema de virtualização não convencional. O docker é uma plataforma Open Source escrita na linguagem de programação desenvolvida no Google, o Go. O docker utiliza o LXC (Linux Containers), um sub-sistema do kernel do Linux, o que permite que ele utilize apenas recursos isolados ao invés de um Sistema Operacional completo, como acontece nos ambientes de virtualização tradicionais.



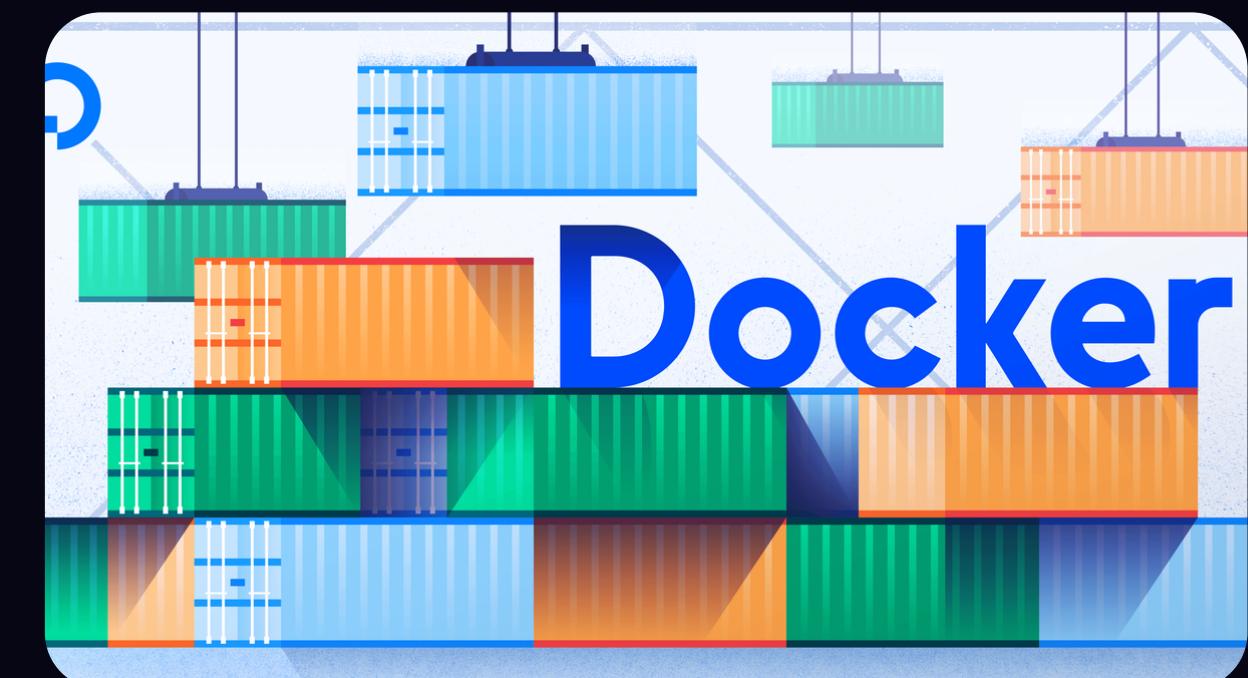
Qual o conceito de Containers?

- Com a chegada dos containers foi possível transportar mercadorias de uma forma segura, de fácil manipulação e com pouco, ou nenhum trabalho braçal no carregamento ou descarregamento. E é justamente isso que o Docker tenta fazer com nossos softwares.



Containers em Docker

- Container é um ambiente isolado dentro de um servidor, dividindo um único host de controle.
- O isolamento dos containers permite um uso limitado de disco rígido, memória RAM e CPU. Utilizando um tipo de compartilhamento de kernel, eles conseguem poupar muito mais recursos que outros métodos de virtualização como uma VM (Virtual Machine).



VANTAGENS DO USO DE CONTAINER

ECONOMIA DE RECURSOS

os containers são definidos como uma imagem em execução ao invés de serem um ambiente read-only. Nessa imagem é gerado um layer extra que armazena todos os dados de uma determinada operação. A natureza de compartilhamento do disco do container docker permite que seja criada uma pilha de imagens somente leitura, com um amontoado para cada container, economizando nos recursos utilizados.

MAIOR DISPONIBILIDADE DO SISTEMA

De uma maneira simples, o container docker deixa a máquina mais "leve", com espaço para executar outros programas. Isso acontece porque o docker compartilha o Sistema Operacional, o que torna os processos mais ágeis.

COMPARTILHAMENTO

Os arquivos podem ser compartilhados entre o container e o host, inclusive utilizando a nuvem para o processo. Isso torna mais fácil a implementação de modificações, permitindo que seja adotada uma gestão mais centralizada. Com isso, o espaço para testes fica mais leve, o que permite, por exemplo, baixar uma solução durante uma reunião.

VANTAGENS DO USO DE CONTAINER

FACILIDADE DE GERENCIAMENTO

O grupo de máquinas físicas ou virtuais onde os containers são executados é chamado de cluster e precisa ser monitorado constantemente. Para isso, foram criadas ferramentas específicas, como o Openshift, que atuam em conjunto com o docker e controlam o equipamento que permite a execução dos containers.

AMBIENTES SIMILARES

A possibilidade de transformar uma aplicação em imagem docker permite que ela seja alocada como container em ambientes diferentes, fazendo com que ela possa ser utilizada tanto no computador do desenvolvedor quanto no servidor da produção, por exemplo.

APLICAÇÃO COMO PACOTE COMPLETO

Graças à imagem do docker, o empacotamento da aplicação e das suas dependências é possível, simplificando o processo de distribuição. Assim, o ambiente pode ser alterado de maneira fácil e rápida. Por causa da estrutura de layers, somente a alteração é transferida na realização de updates.

VANTAGENS DO USO DE CONTAINER

PADRONIZAÇÃO E REPLICAÇÃO

Criar e mudar a infraestrutura é muito mais simples com o container docker, pois as imagens do docker são construídas através de arquivos de definição. Assim, o escalonamento da estrutura fica muito mais simples. Isso permite que um novo colaborador da equipe de TI posso integrar e receber o ambiente de trabalho com apenas alguns comandos, por exemplo.

ACESSO À COMUNIDADE

Outra vantagem do container docker é que ele torna possível conseguir modelos de infraestrutura prontos para integrações com alto grau de complexidade através do acesso ao repositórios de imagens docker. É possível usar as imagens do repositório e configurar os parâmetros para que ele se adeque ao ambiente.

Uma tecnologia bastante complexa e cheia de detalhes, o container docker tem aparecido como grande tendência no mercado graças a sua capacidade de executar tarefas com mais agilidade e maior rapidez. Além disso, a economia de recursos gerada pelo container docker o torna uma solução extremamente atrativa para empresas de diversos segmentos.



Instalando o Docker

- Feito a instalação, execute esse comando no terminal ***docker --version***. Se a instalação ocorreu com sucesso deve ser impresso algo semelhante a isso ***Docker version 17.03.1-ce, build c6d412e.***

Atualmente Docker está disponível em duas versões [Docker Community Edition\(CE\)](#) e [Docker Enterprise Edition\(EE\)](#).



Instalando o Docker

- O Docker trabalha com o conceito de **imagens**, para colocar um **container** em funcionamento o Docker precisa ter a **imagem** no host.
- **Essas imagens podem ser baixadas de um repositório ou criadas localmente e compiladas.**
- A nomenclatura para esse repositório é **registry**

Esse é o link <https://hub.docker.com/> para o registry do Docker.

Baixando Imagens

- Para baixar uma imagem podemos usar o comando ***docker pull*** e o nome da imagem que queremos baixar.

```
[→ ~ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
b6f892c0043b: Pull complete
55010f332b04: Pull complete
2955fb827c94: Pull complete
3deef3fcbd30: Pull complete
cf9722e506aa: Pull complete
Digest: sha256:382452f82a8bbd34443b2c727650af46aced0f94a44463c62a9848133ecb1aa8
Status: Downloaded newer image for ubuntu:latest
→ ~ ]
```

Listando imagens baixadas

- Para listar todas as imagens podemos usar o comando ***docker images***

O retorno desse comando é algo semelhante a isso:

```
[~] ~ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	ebcd9d4fca80	46 hours ago	118 MB

Executando Containers

- A partir da imagem podemos iniciar quantos containers quisermos através do comando ***docker run***
- Para acessarmos um terminal do Ubuntu podemos usar o comando ***docker run -i -t ubuntu*** ou ***docker run -it ubuntu***. O parâmetro ***-i*** indica que queremos um container interativo, o ***-t*** indica que queremos anexar o terminal virtual ***tty*** do container ao nosso host.

Listando containers em execução

- Para ver os containers em execução podemos usar o comando docker ps

Ele exibirá um retorno parecido com esse:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a6697ed945d5	ubuntu	/bin/bash"	12 hours ago	Up 4 seconds		dreamy_bassi

Listando containers em execução

- Quando encerramos um container ele não será mais exibido na saída do comando ***docker ps***, porém isso não significa que o container não existe mais.
- Para verificar os containers existentes que foram encerrados podemos usar o comando ***docker ps -a*** e teremos uma saída parecida com essa:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a6697ed945d5	ubuntu	"/bin/bash"	12 hours ago	Exited (0) 4 seconds ago		dreamy_bassi

Removendo containers

- Para **remover o container** podemos usar o comando **`docker rm`** e informar o **id do container ou o nome dele**.
- Caso tenhamos a necessidade de **remover todos os container (em execução ou encerrados)** podemos usar o comando **`docker rm $(docker ps -qa)`**. A opção **`-q`** do comando **`docker ps`** tem como saída somente os ids dos containers, essa lista de ids é passado para o **`docker rm`** e com isso será removido todos os containers.



Como são feitas as imagens?

- Uma imagem pode ser criada a partir de um arquivo de definição chamado de **Dockerfile**, nesse arquivo usamos algumas diretivas para declarar o que teremos na nossa imagem.
- Com um arquivo Dockerfile podemos compilá-lo com o comando **docker build**.
- Ao compilar um arquivo Dockerfile temos uma imagem.

O que é Dockerfile ?

- O Dockerfile nada mais é do que um meio que utilizamos para criar nossas próprias imagens. Ele serve como a receita para construir um container, permitindo definir um ambiente personalizado.

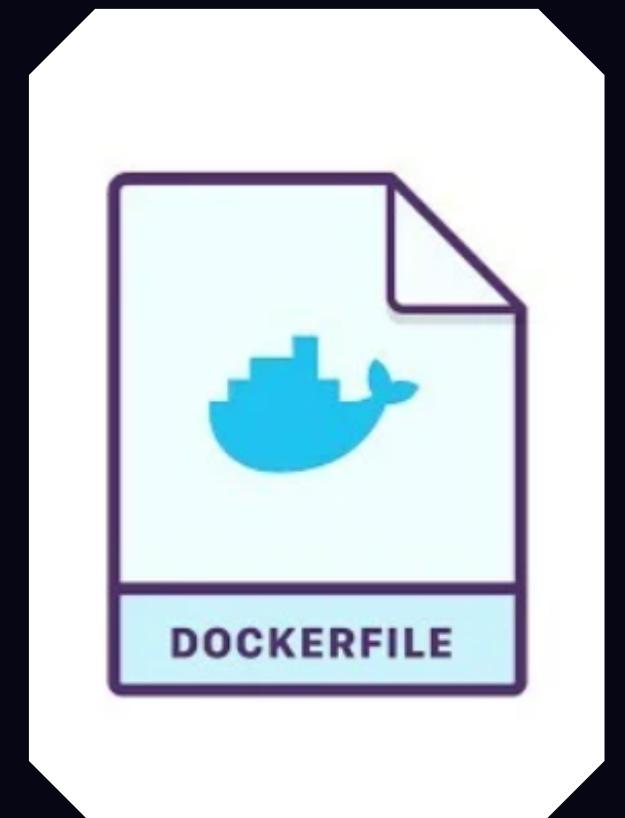
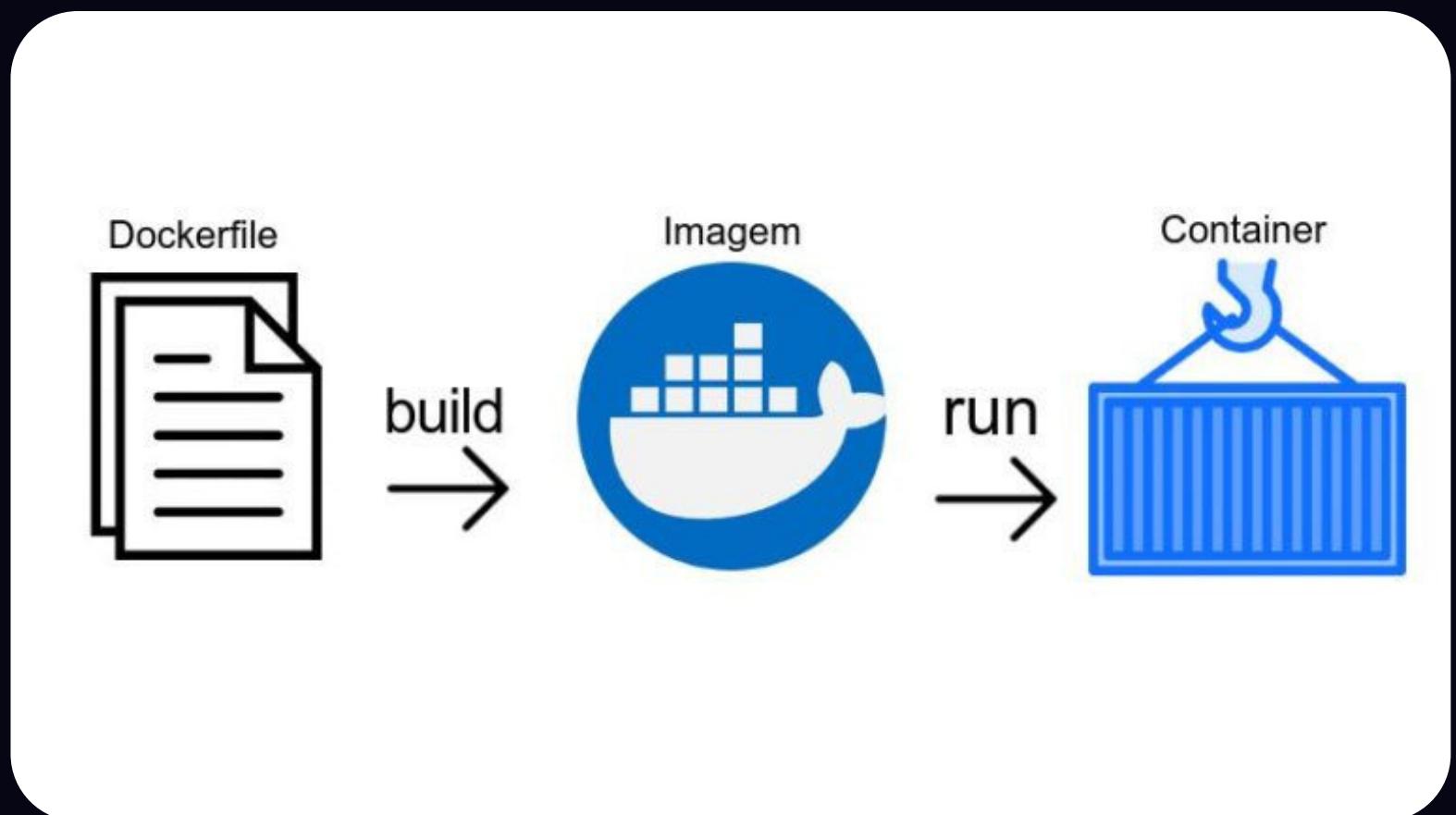


Imagen x Container

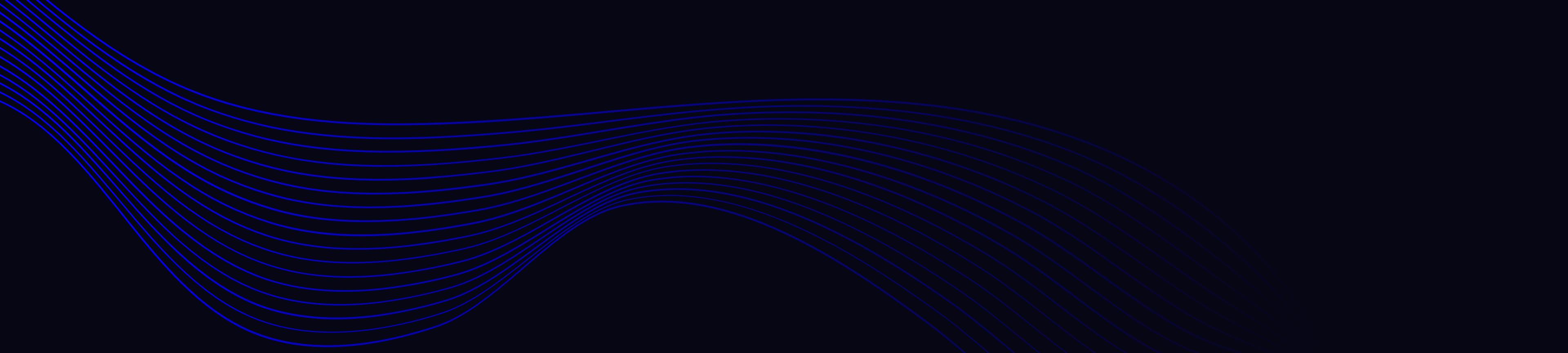
- Uma imagem nada mais é do que uma representação imutável de como será efetivamente construído um container. Por conta disso, nós não rodamos ou inicializamos imagens, nós fazemos isso com os containers.
- O container é o fim enquanto a imagem é o meio.





Casos de Uso do Docker

- **Com o Docker, você pode experimentar um novo software sem instalá-lo manualmente. O Docker também é útil se você precisar ter um software pronto rapidamente.**
- **Por exemplo, configurar um servidor MySQL com o Docker, é necessário apenas um comando via ILC (Interface de Linha de Comando) para fazer isso.**



Execução do projeto





Dúvidas

