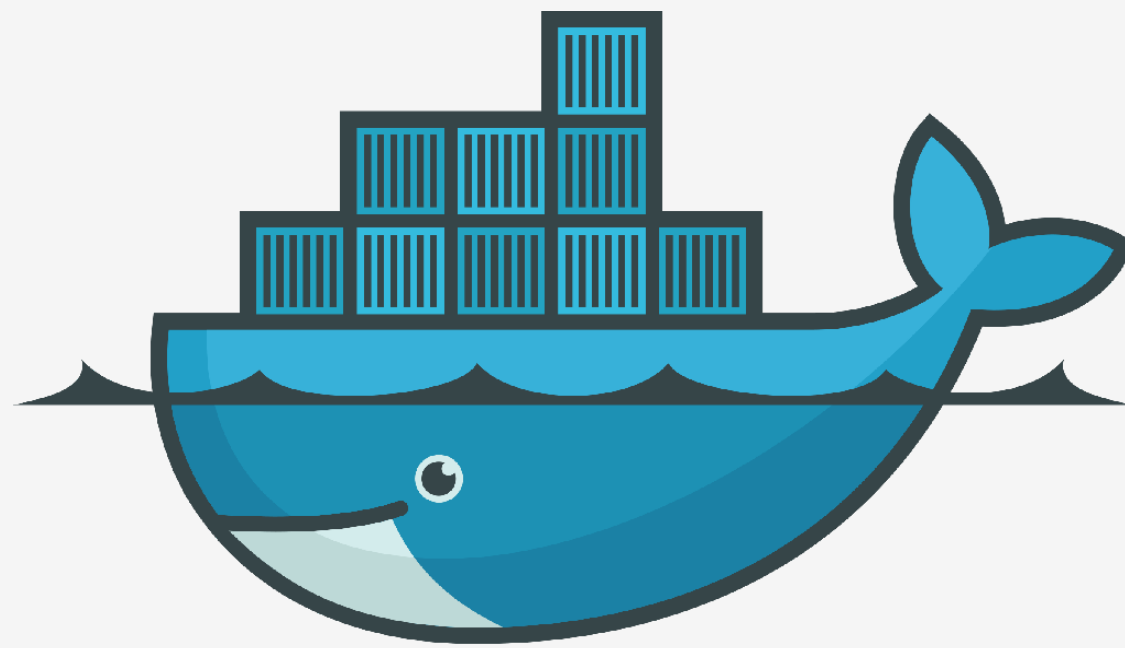


Explicando Docker para iniciantes

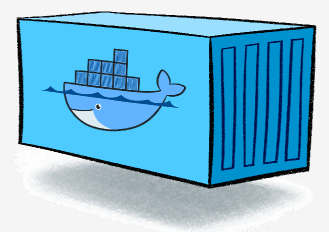
Por meio de algumas analogias



docker



 **Leandro Ucuamba**
Software Engineer



O que é Docker?

Docker é uma plataforma que permite criar, enviar e executar aplicativos dentro de "caixas" chamadas containers.

Pense em um container como uma caixa: você coloca tudo que precisa para o sua aplicação funcionar (código, bibliotecas, dependências, etc).

Isso garante que, se você levar essa caixa para outro computador, ela vai funcionar do mesmo jeito.

Containers vs Máquinas Virtuais

- **Máquinas Virtuais (VMs):**

cada VM precisa de um sistema operacional completo.

–> É como ter um apartamento inteiro para morar, mesmo que você só precise de um quarto.

- **Containers:**

compartilham o mesmo sistema operacional do computador host, mas ainda mantêm seu ambiente isolado.

–> É como ter um quarto privado dentro de uma casa, sem precisar do apartamento inteiro.

Imagens (Images)

Descreve exatamente o que será colocado dentro do container: o sistema, dependências, arquivos e o que a sua aplicação precisa para executar.

Você não pode alterar uma imagem enquanto ela está sendo usada, mas pode criar novas imagens a partir dela.

Containers

Um container é o produto final montado, criado a partir do manual (imagem).

Ele é executável e isolado do resto do sistema.

Você pode ter vários containers executando a partir da mesma imagem, cada um independente.

Dockerfile

É um arquivo de texto que contém instruções para construir uma imagem.

Exemplo de instruções:

- Qual sistema operacional usar;
- Quais pacotes instalar;
- Qual comando executar quando o container iniciar.

Docker Hub

É como a loja de aplicativos do Docker.

Você pode baixar imagens prontas ou enviar suas próprias.

Exemplo:

se você precisa de uma imagem do Node.js, é só baixar do Docker Hub em vez de instalar manualmente no computador.

Comandos básicos

Aqui estão alguns comandos que você vai usar bastante:

- **docker pull <imagem>**
 - baixa uma imagem do Docker Hub.
- **docker build -t <nome>**
 - cria uma imagem a partir de um Dockerfile.
- **docker run <imagem>**
 - inicia um container a partir de uma imagem.
- **docker ps** → lista containers em execução.
- **docker stop <id>** → para um container em execução.
- **docker rm <id>** → remove um container parado.

Resumo com analogia

Imagem -> manual de montagem.

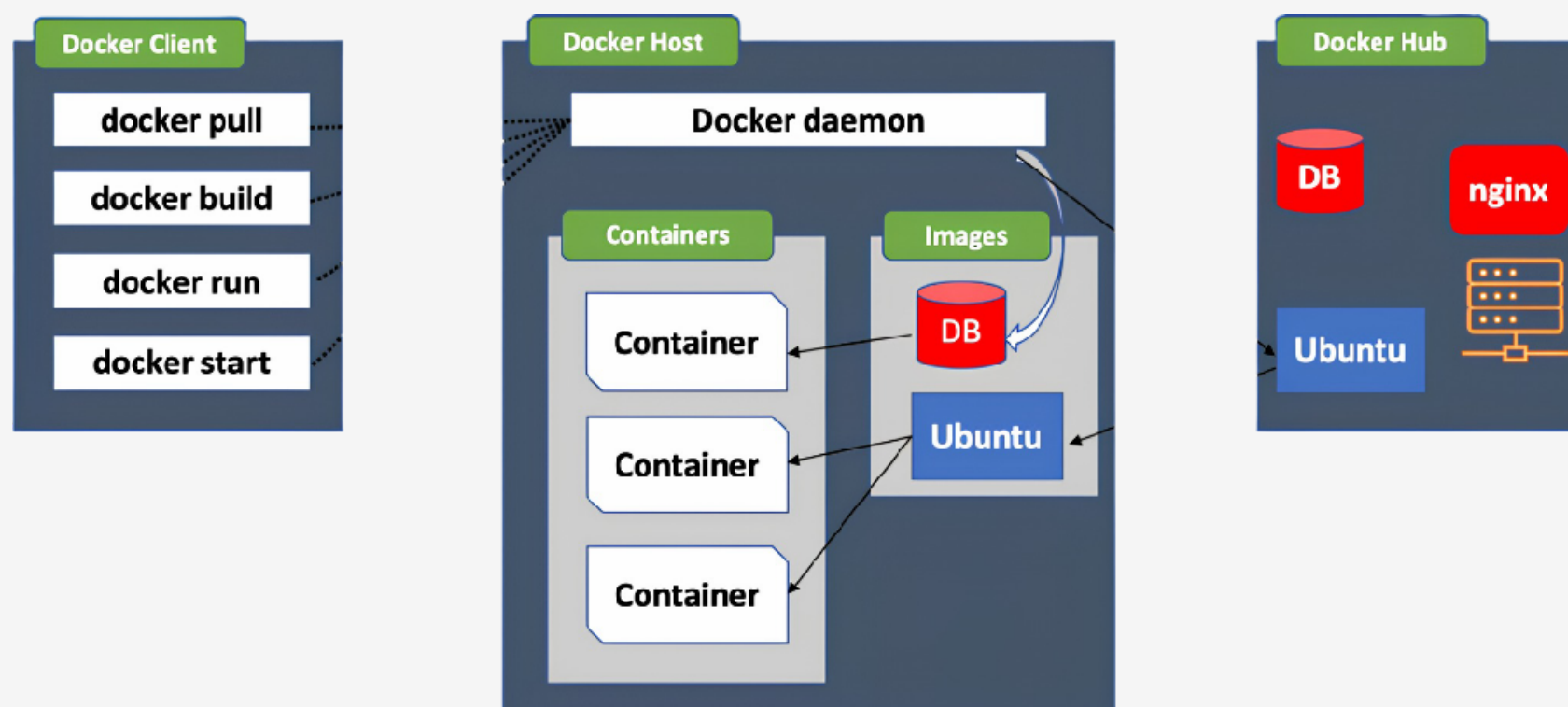
Container

-> produto montado e pronto para usar.

Dockerfile

-> instruções passo a passo.

Docker Hub -> loja de produtos prontos.



Quando o assunto é Docker, você prefere rodar
tudo localmente ou já pensa direto em
containers?

Compartilha tua experiência nos comentários!



 **Leandro Ucuamba**
Software Engineer