Ferramentas de Desenvolvimento:

DOCKER

Gabriela Costa
Guilherme Luiz
Izabela Fonseca
João Pedro Smolinski
Leonardo Prado

Como surgiu?

Nos últimos anos, a computação em nuvem e a virtualização tornaram-se essenciais para o desenvolvimento eficiente de aplicações. No entanto, desafios como gestão de ambientes complexos, consistência entre estágios de desenvolvimento e portabilidade de aplicações surgem. Nesse cenário, o Docker destaca-se como uma ferramenta revolucionária.

Contextualização

O Docker é uma plataforma de código aberto que automatiza a implantação de aplicações em ambientes isolados chamados "containers", encapsulando todas as dependências e bibliotecas. Isso garante uma execução consistente em qualquer ambiente com o Docker instalado.

Dockerfile

Um Dockerfile é um arquivo de configuração de texto que contém instruções para a construção de uma imagem Docker. Estas instruções descrevem as etapas necessárias para criar um ambiente consistente e reproduzível para a execução de um aplicativo específico. O Dockerfile inclui comandos para copiar arquivos para a imagem, instalar dependências, configurar variáveis de ambiente e executar outros passos de configuração necessários.

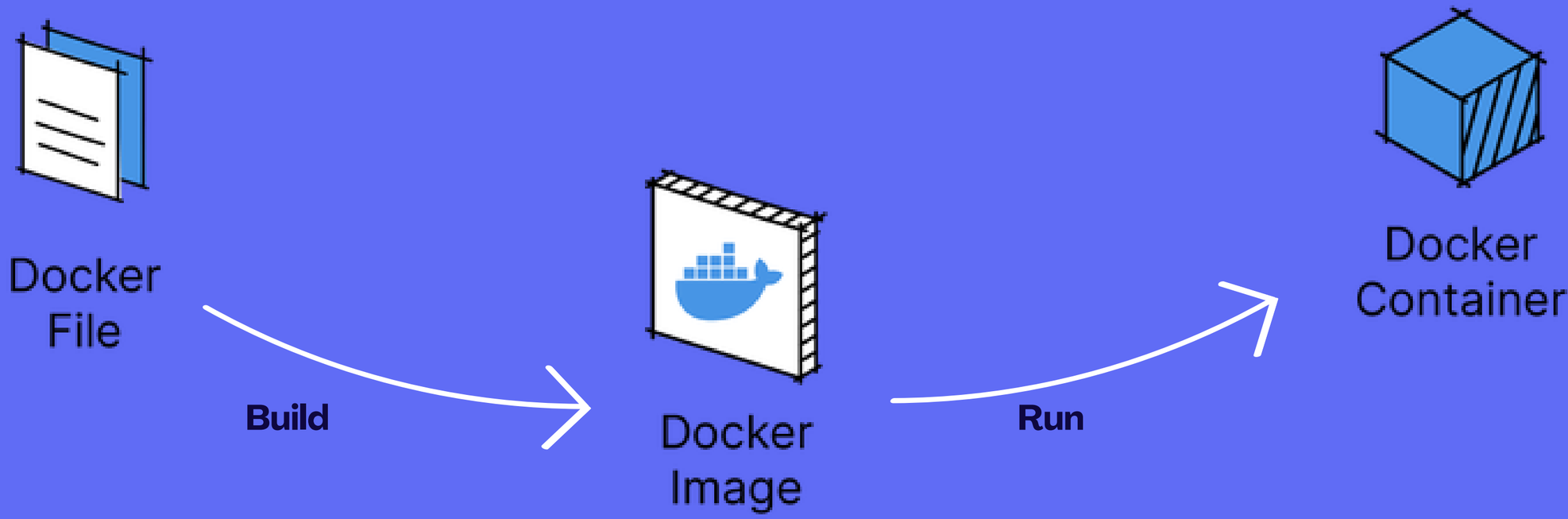
Imagem

Uma imagem Docker é uma versão executável e leve de um sistema operacional que inclui tudo o que é necessário para executar um aplicativo, incluindo código, bibliotecas, dependências e configurações. As imagens são construídas com base nas instruções fornecidas no Dockerfile. Uma imagem é um artefato imutável, o que significa que, uma vez criada, ela não é alterada.

Container

Um container é uma instância em execução de uma imagem Docker. Enquanto a imagem é a versão estática e imutável que contém a aplicação e suas dependências, o container representa a instância em execução dessa imagem, com a capacidade de se comunicar com outros containers e com o sistema operacional host.

Dockerfile ao Container



Docker:

INSTALAÇÃO:

ubuntu

Passo 1:

Atualizar Repositórios:

```
sudo apt update
sudo apt upgrade -y
```

Passo 2:

Instalação do Docker Pré-requisitos:

```
sudo apt install -y apt-transport-
https ca-certificates curl software-
properties-common
```

Passo 3:

Adicione a chave GPG oficial do Docker ao sistema:

```
curl -fsSL
https://download.docker.com/linux/ub
untu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-
keyring.gpg
```

Passo 4:

Adicione o repositório Docker ao sistema:

```
echo ”deb [signed-by=/usr/share/keyrings/docker-
archive-keyring.gpg]
https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable” | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

Passo 5:

Atualize os repositórios novamente e instale o Docker:

```
sudo apt update
sudo apt install -y docker-ce
docker-ce-cli containerd.io
```

Passo 6:

Adicionar seu usuário ao grupo Docker (opcional):

```
sudo usermod -aG docker $USER
```

Passo 7:

Verificar a Instalação do Docker:

```
docker --version
```

Passo 8:

Verificar se o Docker está Funcionando

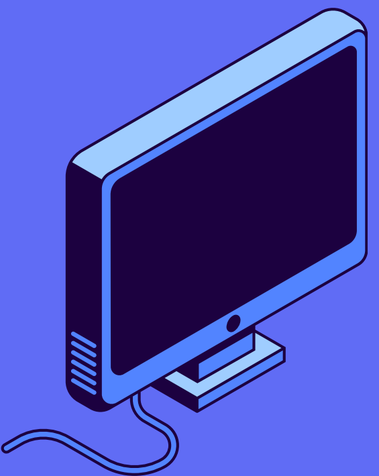
```
docker run hello-world
```

Getting Started com Docker no Ubuntu

Passo 1:

Crie um arquivo chamado Dockerfile

```
FROM node:14
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
```



Passo 2:

Construa a imagem do seu projeto

```
docker build -t nome-do-seu-projeto .
```

Passo 3:

Verifique as imagens disponíveis no seu sistema

```
docker image ls
```

Passo 4:

Execute o Contêiner:

```
docker run -p 3000:3000
nome-do-seu-projeto
```

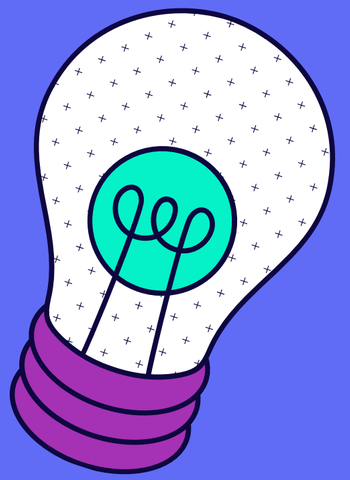
Certifique-se de substituir "nome-do-seu-projeto" pelo nome da sua imagem.

Passo 5:

Para verificar se o contêiner está em execução, use o comando:

```
docker ps
```

Este comando mostrará os contêineres em execução, incluindo o nome, ID, portas expostas, etc.

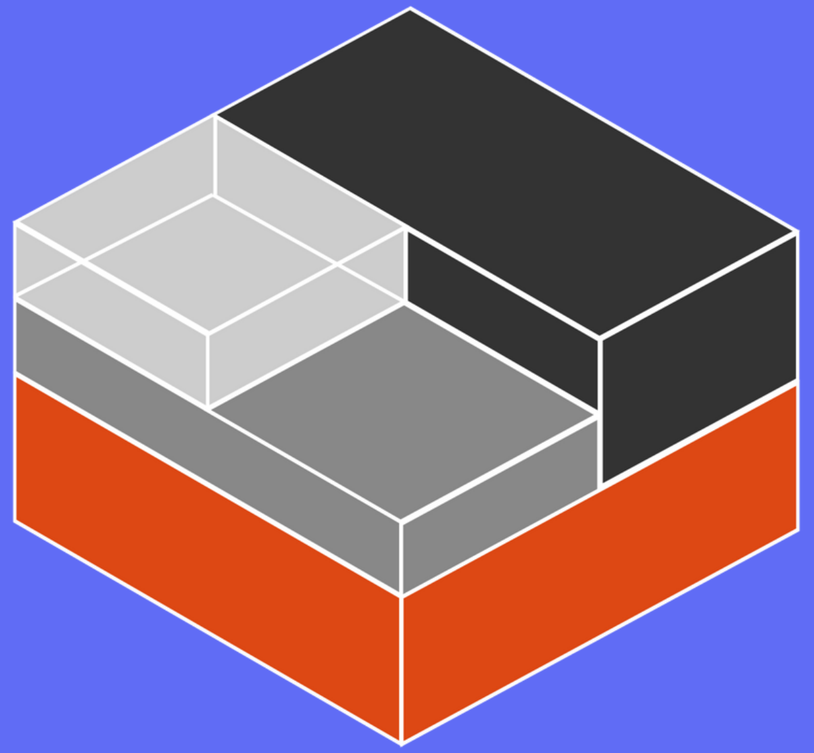


Docker:

PROGRAMAS SIMILARES

LXC

O LXC foi uma das primeiras ferramentas criadas para contornar o problema de redundância e alto gasto das Máquinas Virtuais. Foi criado antes do Docker e acabou sendo ofuscado pelo seu concorrente, que foi lançado 5 anos depois. Como tecnologia, permitia criar containers (ambientes isolados) em sistemas Linux e era mais efetivo que uma máquina virtual pois fazia com que o kernel utilizado pelo container fosse o mesmo do sistema, diminuindo os gastos.



Dentre os componentes do LXC há a biblioteca “liblxc” e suporte às linguagens de programação: Lua, Python, Ruby, Go e Haskell.

podman

De forma semelhante ao Docker, o Podman também permite criar, executar e gerenciar contêineres. Por outro lado, se difere do Docker no sentido de que não precisa de um Daemon central para que possa ser executado, o que pode ser vantajoso em alguns casos. Oferece uma transição suave para os desenvolvedores que já são familiarizados com o Docker por sua compatibilidade de ferramentas.

Também é compatível com diversas ferramentas de desenvolvimento conhecidas, como VSCode e GitHub

FIM!

Página de recursos

Use estas figuras e ilustrações na sua apresentação do Canva. Não se esqueça de excluir esta página antes de apresentar.