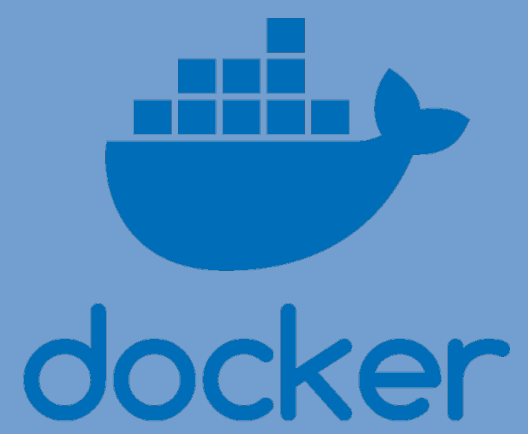


Docker: Guia Essencial para Iniciantes



título

Teste teste teste teste teste

Teste teste teste teste teste

01

Introdução

Introdução

Docker é uma plataforma de código aberto que facilita a criação, envio e execução de aplicações em contêineres. Ele permite que você empacote uma aplicação com todas as partes necessárias, como bibliotecas e outras dependências, e envie tudo isso como um único pacote. Isso garante que a aplicação funcionará em qualquer outro sistema Docker, independentemente de quaisquer configurações personalizadas que o sistema base possa ter.



docker

02

Instalação



Instalação

Windows e Mac

Baixe o Docker Desktop: Acesse docker.com e baixe o Docker Desktop para Windows ou Mac.

Instale o Docker Desktop: Siga o assistente de instalação.

Inicie o Docker Desktop: Após a instalação, inicie o Docker Desktop e siga as instruções para completar a configuração.

Instalação

Linux

Atualize o repositório de pacotes:

```
sudo apt-get update
```

Instale pacotes necessários:

```
sudo apt-get install apt-transport-https ca-certificates curl  
gnupg lsb-release
```

Adicione a chave GPG oficial do Docker:

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

Adicione o repositório do Docker:

```
echo "deb [arch=amd64  
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
```

Instale o Docker Engine:

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

Verifique a instalação:

```
sudo docker run hello-world
```

03

Criando seu primeiro Contêiner

Criando seu Primeiro Contêiner

Executando um Contêiner Simples

Vamos começar executando um contêiner simples usando uma imagem oficial do Docker. A imagem hello-world é um bom ponto de partida.

```
sudo docker run hello-world
```

Esse comando baixa a imagem hello-world e executa um contêiner que simplesmente imprime uma mensagem.

04

Trabalhando com Imagens Docker

Trabalhando com Imagens Docker

Baixando uma Imagem

Para baixar uma imagem do Docker Hub, use o comando `docker pull`. Vamos baixar a imagem do Nginx, um popular servidor web.

```
sudo docker pull nginx
```

Executando um Servidor Web com Nginx

Agora, vamos executar um servidor web Nginx:

```
sudo docker run --name meu-nginx -d -p 8080:80 nginx
```

--name meu-nginx: dá um nome ao contêiner.

-d: executa o contêiner em segundo plano (detached).

-p 8080:80: mapeia a porta 8080 do host para a porta 80 do contêiner.

Você pode acessar o servidor web Nginx em <http://localhost:8080>.

05

Criando uma Imagem Personalizada

Criando uma Imagem Personalizada

Dockerfile

Um Dockerfile é um arquivo de texto simples que contém uma série de instruções que são executadas para criar uma imagem Docker. Vamos criar um Dockerfile para uma aplicação Node.js simples.

Crie um diretório para o projeto:

```
mkdir meu-app-node  
cd meu-app-node
```

Crie um arquivo Dockerfile:

```
# Usando a imagem base do Node.js  
FROM node:14  
# Definindo o diretório de trabalho no contêiner  
WORKDIR /usr/src/app  
# Copiando o package.json e package-lock.json  
COPY package*.json ./  
# Instalando as dependências  
RUN npm install  
# Copiando o restante do código da aplicação  
COPY . .  
# Expondo a porta da aplicação  
EXPOSE 8080  
# Comando para rodar a aplicação  
CMD [ "node", "app.js" ]
```

Criando uma Imagem Personalizada

Dockerfile

Crie um arquivo app.js:

```
const express = require('express');
const app = express();
const port = 8080;
app.get('/', (req, res) => {
  res.send('Hello, Docker!');
});
app.listen(port, () => {
  console.log(`App listening at http://localhost:${port}`);
});
```

Crie um arquivo package.json:

```
{
  "name": "meu-app-node",
  "version": "1.0.0",
  "main": "app.js",
  "scripts": {
    "start": "node app.js"
  },
  "dependencies": {
    "express": "^4.17.1"
  }
}
```

Construa a imagem Docker:

```
sudo docker build -t meu-app-node .
```

Execute o contêiner:

```
sudo docker run -p 8080:8080 meu-app-node
```

Acesse a aplicação em <http://localhost:8080> para ver a mensagem "Hello, Docker!".

06

Criando uma Imagem Personalizada

Conclusão

Neste ebook, você aprendeu a instalar o Docker, criar e executar contêineres, baixar e utilizar imagens do Docker Hub e construir suas próprias imagens Docker personalizadas. Docker é uma ferramenta poderosa que simplifica a implantação e execução de aplicações, garantindo consistência e eficiência no desenvolvimento e operação.