

Docker



Alex Lopes - 1133527

Docker é uma plataforma de código aberto que permite automatizar o processo de criação, implantação e execução de aplicativos dentro de contêineres.

Contêineres são ambientes isolados que contêm todos os recursos necessários para executar um aplicativo, incluindo o código, bibliotecas, dependências e configurações.

Com o Docker, os aplicativos podem ser empacotados em contêineres consistentes e portáteis que podem ser executados em qualquer ambiente.

O que faz e como faz:

O Docker simplifica o processo de implantação de aplicativos, fornecendo uma plataforma consistente e isolada.

Ele utiliza tecnologias como namespaces e cgroups do Linux para criar contêineres leves e eficientes.

Os containers Docker podem ser construídos a partir de imagens, que são pacotes de arquivos com todos os componentes necessários para executar um aplicativo.

O Docker utiliza um formato de imagem chamado Dockerfile, que é um arquivo de texto com instruções para construir uma imagem.

O Dockerfile contém comandos para especificar a base da imagem, copiar arquivos, instalar dependências, expor portas, definir variáveis de ambiente e executar comandos durante a construção da imagem.

Onde é utilizado:

O Docker é amplamente utilizado na indústria de desenvolvimento de software para criar ambientes consistentes e reproduzíveis.

Ele é usado para implantar aplicativos em escala, facilitar a integração contínua e a entrega contínua (CI/CD), simplificar o gerenciamento de infraestrutura, fornecer ambientes de desenvolvimento isolados e muito mais.

O Docker é comumente usado em ambientes de nuvem, orquestradores de contêineres como o Kubernetes e por equipes de desenvolvimento e operações.

Quem utiliza:

O Docker é utilizado por uma ampla gama de profissionais, incluindo desenvolvedores de software, administradores de sistemas, engenheiros de DevOps e equipes de operações em empresas de todos os tamanhos.

Ele é adotado por organizações líderes em tecnologia e por uma comunidade global de desenvolvedores.

Imagens utilizadas:

Foram utilizadas duas imagens:



Imagem do serviço web Python:

A partir do Dockerfile, foi criada uma imagem personalizada baseada na imagem oficial do Python 3.9.

Essa imagem é usada para criar um contêiner que executa o serviço web Python.

A configuração da imagem inclui copiar o código-fonte da aplicação para o diretório de trabalho /app, instalar as dependências listadas no arquivo requirements.txt, expor a porta 8000 do contêiner e executar o comando `python app.py` para iniciar o serviço.

Imagem do banco de dados MySQL:

A partir do Dockerfile.mysql, foi criada uma imagem personalizada baseada na imagem oficial do MySQL 8.0.

Essa imagem é usada para criar um contêiner que executa o banco de dados MySQL. A configuração da imagem inclui definir a senha do usuário root através da variável de ambiente MYSQL_ROOT_PASSWORD, definir o nome do banco de dados através da variável de ambiente MYSQL_DATABASE, copiar o arquivo init.sql para o diretório /docker-entrypoint-initdb.d/ dentro do contêiner (esse arquivo é executado durante a inicialização do contêiner e contém comandos SQL para criar a tabela users), e expor a porta 3306 do contêiner para permitir conexões com o banco de dados.

Configuração de cada serviço no Docker Compose:

No arquivo docker-compose.yml, a configuração de cada serviço é definida da seguinte forma:

Serviço web:

services:

web:

build:

context: .

dockerfile: Dockerfile

ports:

- "8000:8000"

depends_on:

- db

O serviço web é construído a partir do diretório atual (.) usando o Dockerfile especificado.

A porta 8000 do contêiner é mapeada para a porta 8000 do host ("8000:8000"), permitindo que o serviço web seja acessado a partir do host.

O serviço web depende do serviço de banco de dados (db), o que significa que o serviço de banco de dados será iniciado antes do serviço web.

Serviço de banco de dados:

db:

build:

context: .

dockerfile: Dockerfile.mysql

O serviço de banco de dados é construído a partir do diretório atual (.) usando o Dockerfile.mysql especificado.

Além disso, o arquivo init.sql é copiado para o diretório /docker-entrypoint-initdb.d/ do contêiner MySQL, conforme especificado no Dockerfile.mysql. Esse arquivo será executado durante a inicialização do contêiner para criar a tabela users no banco de dados.

Docker Compose:

O Docker Compose é uma ferramenta que permite definir e gerenciar aplicativos multi-contêiner em um arquivo YAML, utilizamos o arquivo `docker-compose.yml` para definir e orquestrar os serviços web e de banco de dados.

Ao executar o comando `docker-compose up`, o Docker Compose lê o arquivo de composição e inicia os contêineres especificados, construindo as imagens se necessário.

O Docker Compose lida com a criação e gerenciamento das redes, volumes e dependências entre os contêineres definidos no arquivo.

Com o Docker Compose, é possível iniciar e parar facilmente os serviços, visualizar logs, escalar os serviços para múltiplos contêineres e personalizar várias configurações do ambiente. Ele fornece uma forma simplificada de configurar e implantar aplicativos compostos por vários contêineres.

Obrigado!