

Docker

Conceitos

- O que é?

O Docker é uma ferramenta em código aberto que possibilita a criação e a administração de ambientes isolados. Tem como principal objectivo implementar aplicações nesses ambientes separados da máquina original. Esses ambientes têm o nome de containers.

- O que são containers?

Os containers são ambientes isolados portáteis, que possibilitam aos desenvolvedores “empacotar” aplicações. A grande vantagem na utilização da plataforma é a velocidade em que o software é disponibilizado. Outras vantagens são: a disponibilidade maior do sistema, economia significativa de recurso e a possibilidade de compartilhamento.

- O que são as Imagens?

As imagens são como um “gerador” de containers, será a partir da imagem que o container vai ser criado. Existem várias imagens de diversos serviços que é possível descarregar.

- Como mudar uma imagem existente?

Não é possível modificar directamente a configuração de uma imagem existente, no entanto, podemos criar um container a partir dessa imagem. Podendo modificar o container e depois gerar uma nova imagem com estas modificações. Outra opção será criar uma imagem a partir de um dockerfile.

Comandos principais

- Docker pull (Utilizado para descarregar uma imagem)
- Docker ps (Listar todos os containers existentes)
- Docker inspect (Inspecionar um container específico)
- Docker exec (Executar um comando em um container que esteja correndo)
- Docker stop (Parar um container)
- Docker start (Iniciar um container)
- Docker rm (Eliminar um container)
- Docker rmi (Eliminar uma imagem)
- Docker images (Listar imagens existentes)
- Docker search (Procurar imagens)
- Docker run (Cria um container a partir de uma imagem específica e depois inicia ele)
- Docker push (Enviar uma imagem para o Docker Hub, para assim ser compartilhada)
- Docker commit (Cria uma nova imagem a partir das modificações de um container)
- Docker build (Cria uma imagem com base em um Dockerfile)

Dockerfile

É possível criar imagens a partir de um Dockerfile. Um Dockerfile é nada mais que um documento em texto que contém todos os comandos necessários para gerar uma imagem específica. Utilizando o comando *Docker build* é possível criar uma imagem com base nesse Dockerfile.

Projeto elaborado

Objetivo

O projeto consistia na exploração das ferramentas de criação de imagem. Tendo conseguido configurar uma imagem de apache2 de dois métodos diferentes. Além da criação e compartilhamento da mesma, foi possível criar um volume para compartilhar arquivos entre o *host* e o container.

O primeiro método para efetuar a criação de uma imagem, consistiu em pegar uma imagem já existente como base. Podendo assim, modificar o container e criar uma nova imagem com base no mesmo.

O processo consistiu em:

- `docker create --name diogo_apache -p 80:80 httpd`

(Esse comando irá criar um novo container com base na imagem “*httpd*”, o nome desse container será “*diogo_apache*” e correrá na porta 80).

- `Docker commit diogo-apache`

(Irá criar uma nova imagem a partir do container *diogo_apache*).

- `Docker login`

(Efetuar login de uma conta no Docker Hub, para assim, fazer upload da imagem)

- `Docker tag [ID Imagem] [nome do utilizador]/[nome repositório]:[tag]`

(Criar uma tag para a imagem)

- `Docker push [nome do utilizador]/[nome repositório]:[tag]`

(Upload da imagem para o Docker Hub)

Sendo apenas necessário efetuar os seguintes comandos para descarregar e correr a imagem:

- `Docker pull [nome utilizador]/[nome repositório]:[tag]`
- `Docker run -d -p 80:80 [nome utilizador]/[nome repositório]:[tag]`

O segundo método envolveu a configuração do apache2 em um Dockerfile, com o objetivo de gerar uma imagem com base nesse documento.

A configuração do ficheiro Dockerfile foi a seguinte:

```
FROM ubuntu:18.04
MAINTAINER diogocarreira
RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -r /var/lib/apt/lists/*
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
ENV APACHE_PID_FILE /var/run/apache2/apache2.pid
ENV APACHE_RUN_DIR /var/run/apache2
ENV APACHE_LOCK_DIR /var/lock/apache2
ENV APACHE_LOG_DIR /var/log/apache2
#Build the application
RUN mkdir -p $APACHE_RUN_DIR
RUN mkdir -p $APACHE_LOCK_DIR
RUN mkdir -p $APACHE_LOG_DIR
COPY index.html /var/www/html
# Expose [Port]
EXPOSE 80
#CMD run as soon a container is launched
CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
```

Não esquecer de criar um index.html que vai ser utilizado no servidor web.

Depois de ter o dockerfile configurado, será necessário completar o processo de criação da imagem (Docker build -t [Docker_name] .), tag e upload da mesma.

Compartilhamento de dados entre o host e o container

Para efetuar o compartilhamento de um ficheiro entre o host e o Docker Container será necessário criar um volume e especificar as pastas que serão partilhadas, tanto do host quanto do container:

- `docker run --name=[nome_container] -d -v ~/[pasta_host]:[pasta_container] -p 5000:80 [nome_do_container]`