03. Criando e compreendendo imagens

Date	@21/02/2023
Categoria	Docker
Curso	Docker: criando e gerenciando containers

Tópicos

- Entendendo imagens
- Detalhes sobre imagens
- Criando a primeira imagem
- Instruções no Dockerfile
- Faça como eu fiz: Construindo uma imagem
- Incrementando a imagem
- ARG vs ENV
- Subindo a imagem para o Docker Hub
- O que aprendemos?

Detalhes sobre imagens

Anteriormente vimos como as imagens se diferem de containers e como são compostas. Sabemos até então que imagens são “receitas” para gerar containers, porém, ainda precisamos fixar como imagens são construídas e gerenciadas pelo Docker.

Selecione as alternativas verdadeiras sobre a utilização de imagens.

- Imagens são compostas por uma ou mais camadas.

    - As camadas são a menor unidade que compõem uma imagem.
- Podemos visualizar as camadas de uma imagem através do comando `docker history`.
    - Este comando é responsável por exibir quais são as camadas de uma imagem.
- Imagens são mutáveis.
    - Imagens são **imutáveis**.
- Podemos consultar as imagens locais com o comando `docker showimages`.
    - Este comando não existe.

Instruções no Dockerfile

Já aprendemos que usamos Dockerfiles para criar nossas imagens quando queremos utilizar containers. Para isso, utilizamos algumas instruções.

Dentre as alternativas, escolha a que melhor define a utilização da instrução `FROM`.

- A instrução `FROM` é usada apenas para fins de documentação.
    - A instrução `FROM` é usada para definir a base de nossa imagem que será criada.
- A instrução `FROM` é usada para definirmos uma imagem como base para a nossa.
    - Desta maneira, podemos adicionar à nossa imagem conteúdos que utilizaremos de maneira mais prática.
- A instrução `FROM` abre um pedido para alterarmos a imagem original.
    - Este não é o comportamento da instrução `FROM`.

Faça como eu fiz: Construindo uma imagem

Agora criaremos nosso primeiro Dockerfile para poder gerar nossa primeira imagem. Consequentemente, teremos nosso primeiro container próprio com o Docker. Durante as etapas anteriores, compreendemos a composição de imagens

através de camadas e vimos a necessidade de criar imagens próprias a fim de ter um ambiente para executar nossas aplicações.

Inicialmente, crie um diretório com um nome à escolha. Dentro dele, crie um arquivo chamado `Dockerfile` e extraia o conteúdo da pasta do zip neste mesmo diretório. Este será o arquivo usado para o Docker interpretar as instruções e construir a imagem final.

Com seu editor de texto favorito, edite o arquivo e adicione o seguinte conteúdo:

```
FROM node:14
WORKDIR /app-node
COPY . .
RUN npm install
ENTRYPOINT npmstart
```

Com estas instruções acima, estamos informando ao Docker que queremos usar a imagem do `node` na versão 14 como base para nossa imagem. Definimos que nosso diretório padrão para executar os comandos dentro do container será o `/app-node` e em seguida copiamos todo o conteúdo do diretório atual Dockerfile para o diretório `/app-node` dentro do container.

Por fim, em tempo de construção da imagem, executamos o comando `npm install` e definimos que, ao executar o container gerado por esta imagem, o comando executado será o `npm start`.

Ainda dentro do diretório do Dockerfile, através do terminal, execute o comando `docker build -t <seu-nome-de-usuario-do-docker-hub>/app-node:1.0 .`. Dessa forma, sua primeira imagem será criada. Confira através do comando `docker images` se sua imagem está sendo listada.

Saber como criar e definir imagens é de suma importância, pois elas são a base para o funcionamento de um futuro container.

ARG vs ENV

Vimos como podemos utilizar variáveis em nossas imagens e containers através das instruções `ARG` e `ENV`. Exemplificamos durante a aula o cenário de passar alguma informação para ser carregada dentro do container, por exemplo, qual será a porta que a aplicação usará para executar.

- A instrução `ARG` carrega variáveis apenas no momento de build da imagem, enquanto a instrução `ENV` carrega variáveis que serão utilizadas no container.
    - Desta maneira, é possível diferenciar o que é necessário para a etapa de build e para a etapa de execução.

O que aprendemos?

Nessa aula aprendemos:

- Imagens são imutáveis, ou seja, depois de baixadas, múltiplos containers conseguirão reutilizar a mesma imagem;
- Imagens são compostas por uma ou mais camadas. Dessa forma, diferentes imagens são capazes de reutilizar uma ou mais camadas em comum entre si;
- Podemos criar nossas imagens através de Dockerfiles e do comando `docker build`;
- Para subir uma imagem no Docker Hub, utilizamos o comando `docker push`.
