UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO

**Roteiro de Aula Prática – Construção de Imagens Docker**

DISCIPLINA: DCA0132 – ENGENHARIA DE DADOS

PROFESSOR: Carlos Manuel Dias Viegas

Esta prática consiste em criar arquivos Dockerfile para a construção de imagens e posterior execução em containers Docker. A prática pode ser realizada em duplas.

O objetivo desta prática é criar duas imagens que serão executadas em dois containers diferentes, sendo uma cliente e outra servidor. O cliente enviará uma mensagem para o servidor e este exibirá em tela. Em seguida, após sucesso na criação e execução dos containers, as imagens serão enviadas para o DockerHub.

Ao final da tarefa, devem ser submetidos no SIGAA, em tarefa específica, os links para os repositórios criados, bem como os arquivos Dockerfile criados.

**Tarefas**

1. Criar os arquivos Dockerfile contendo as informações necessárias para a criação das imagens.

Dockerfile.client

|  |
| --- |
| FROM python:3-slim  WORKDIR /app  COPY clienteTCP.py /app  ENTRYPOINT ["python","clienteTCP.py"] |

Dockerfile.server

|  |
| --- |
| FROM python:3-slim  WORKDIR /app  COPY servidorTCP.py /app  CMD ["python","servidorTCP.py"] |

Referência para consulta sobre Dockerfile:

<https://docs.docker.com/engine/reference/builder/>

1. Baixar os scripts clienteTCP.py e servidorTCP.py a partir da página web:

<https://www.dca.ufrn.br/~viegas/disciplinas/DCA0132/files/Sockets/>

1. Criar dois repositórios no DockerHub (<https://hub.docker.com/>), sendo um para o cliente (com nome clientetcp) e outro para o servidor (com nome servidortcp). Marcar ambos como públicos.
2. Construir as imagens a partir dos arquivos Dockerfile.

Imagem cliente:

docker build -f Dockerfile.client -t <usuario-dockerhub>/clientetcp:v1 .

Imagem servidor:

docker build -f Dockerfile.server -t <usuario-dockerhub>/servidortcp:v1 .

PS: O ponto ao final indica a atual pasta em que estão os arquivos a serem utilizados na construção da imagem. Porém, é possível especificar o caminho para outra pasta, caso necessário.

Legenda:

|  |
| --- |
| Comando padrão para envio:  docker build [opções] <tag-da-imagem> <pasta-local>  O campo opções permite uma série de parâmetros, tais como:  -f : para especificar qual o Dockerfile de entrada para a construção da imagem  -t : para definir a tag da imagem  Referência para consulta:  <https://docs.docker.com/engine/reference/commandline/build/> |

1. Executar o container servidor:

docker run -it <usuario-dockerhub>/servidortcp:v1

1. Executar o container cliente:

docker run <usuario-dockerhub>/clientetcp:v1 <ip-do-servidor>

PS: O container cliente deve ser autossuficiente, de maneira que não é necessário acessar o terminal do mesmo para executar o script cliente. Basta apenas executar o container e passar o IP do servidor como parâmetro e aguardar o resultado na tela do servidor.

Legenda:

|  |
| --- |
| Comando padrão para execução de um container:  docker run [opções] <tag-da-imagem-criada> [parâmetros-se-necessário]  O campo opções permite uma série de parâmetros, tais como:  -it : para acesso direto ao terminal (interativo)  --name : para definir um nome ao container que será executado  -p X:Y : para definir o número da porta que será exposta para acesso ao container  -rm : para remover o container quando ele terminar a execução  Referência para consulta:  <https://docs.docker.com/engine/reference/commandline/run/> |

1. Após sucesso na execução de ambos os containers, com o cliente enviando informações para o servidor, deve-se enviar as imagens para o DockerHUB.

Fazer login no Dockerhub

docker login

Enviar a imagem do cliente para o Dockerhub:

docker push <usuario-dockerhub>/clientetcp:v1

Enviar a imagem do servidor para o Dockerhub:

docker push <usuario-dockerhub>/servidortcp:v1

Legenda:

|  |
| --- |
| Comando padrão para envio de imagens:  docker push <nome-do-usuario-dockerhub>/<tag-da-imagem-criada>  Referência para consulta:  <https://docs.docker.com/engine/reference/commandline/push/>  <https://docs.docker.com/docker-hub/> |