Certified Tech Developer
The Ultimate Degree

Objetivos

Construir um Dockerfile para executar o script em Python criado nas aulas anteriores. Vamos usar o pedrapapeltesoura.py ou pedrapapeltesouralagartospock.py. Link do pedrapapeltesoura.py

O que vamos construir?

Vamos fazer um Dockerfile para criar uma imagem personalizada para depois executarmos um container baseado nesta imagem.

Instruções

Abrir sua IDE preferida, bloco de notas ou algum editor de texto no Linux como VI, VIM ou nano.
Então devemos criar um arquivo chamado Dockerfile
O arquivo deve ter esta estrutura:

```
FROM python:alpine
WORKDIR /diretorioprinc/
COPY pedrapapeltesoura.py /diretorioprinc/
ENTRYPOINT ["python", "pedrapapeltesoura.py"]
```

Dica: Se analisarmos o que acabamos de escrever, vamos ver que é composto por quatro termos:

- FROM: Este termo serve para solicitar a imagem base ao regristry do Docker, no caso o DockerHub, vamos utilizar uma imagem que já vem com o Python instalado e utiliza o Linux Alpine como S.O.
- WORKDIR: Este cria um diretório e faz ele ser nosso diretório default ao acessar via bash.
- COPY: Este comando copia arquivos ou pastas que especificados na primeira parte do comando no nosso caso o *pedrapapeltesoura.py* para o local especificado na segunda parte do comando no caso o */diretorioprinc/* .
- ENTRYPOINT: E finalmente, este termo é um parâmetro que diz a nossa imagem que os comandos especificados devem ser executados quando o container for iniciado.

Agora que temos nosso Dockerfile pronto, temos que colocar na mesma pasta o arquivo pedrapapeltesoura.py ou outro arquivo que você queira executar:

```
    Diretório: C:\Users\nidio\Documents\GitHub\digitalhouse\Infra1\aula14


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         28/03/2022     10:38            135 Dockerfile
-a----         24/03/2022     20:32           1766 pedrapapeltesoura.py
```

Utilizaremos o comando build para construir a imagem baseada no nosso Dockerfile.

```
docker image build . --tag nomedaimagem
```

Dica: Se analisarmos o comando que acabamos de executar, vamos ver que é composto por:

- docker: para invocar o já famoso cliente Docker ou Docker CLI (Command Line Interface)
- image: este termo engloba todas as operações relacionadas às imagens no docker.
- build: ao usar este termo, no contexto de “image”, estamos indicando que queremos criar uma imagem baseada no Dockerfile que está no mesmo diretório, por isso a utilização do “.” (ponto) após o build.
- --tag ou -t: no contexto de “image” adiciona um nome para a imagem.

Para saber mais sobre os comandos que estão no grupo “image” podemos executar:

```
docker image --help
```

Podemos verificar se nossa imagem foi criada, executando o comando:

```
docker images --all
```

Dica: podemos usar o -a também.

```
→docker images -a
REPOSITORY      TAG         IMAGE ID        CREATED         SIZE
nomedaimagem    latest      3c5476bf4374    3 hours ago     47.8MB
```

```
docker container run --interactive --tty --name nomedocontainer nomedaimagem
```

Dica: Este comando tem muitos termos. Vamos analisá-los:

- docker: novamente, para invocar o cliente Docker ou Docker CLI (Command Line Interface)
- container: Para dizer que vamos trabalhar no contexto de container, mas ele pode ser suprimido e utilizarmos apenas o run:
- run: este termo indica que vamos executar um novo container.
- --interactive ou -i: para indicar que vamos usar um terminal interativo, no caso o bash.
- --name: com este parâmetro, atribuímos um nome ao nosso container, neste caso 'nomedocontainer'.
- Na última instrução colocamos o nome da imagem que vai ser utilizada em nosso container, poderia ser imagens que contidas no registry do Docker, mas iremos utilizar a imagem que nós criamos

Para saber mais sobre os parâmetros que podemos usar para o comando 'run', podemos executar:

```
docker run --help
```

Se tudo deu certo teremos o jogo pedra, papel e tesoura rodando no nosso terminal, o container se manterá ligado enquanto estivermos jogando:

```
→docker container run -it --name nomedocontainer nomedaimagem bash
1)Pedra
2)Papel
3)Tesoura
4)Sair do Programa
O que você escolhe: 2
Tua escolha:  papel
PC escolheu:  tesoura
...
Perdeu, tesoura corta papel
Jogar novamente? s/n: n
Nos vemos!
```

Caso queira jogar novamente sem criar um container novamente, podemos inicializar nosso container com o comando:

```
docker start -i nomedocontainer
```

Dica: Este comando tem poucos termos. Mas vamos analisá-los:

- docker: novamente, para invocar o cliente Docker ou Docker CLI (Command Line Interface)
- start: Para inicializar um container que já foi criado:
- -i: para indicar que vamos usar um terminal interativo, no caso o bash.
- Na última instrução colocamos o nome do container que vamos inicializar.

Para saber mais sobre os parâmetros que podemos usar para o comando 'run', podemos executar:

```
docker start --help
```

Se chegou até aqui parabéns!!!

Se você observou, viu que algumas instruções podem ter suas flags abreviadas como `--interactive` para `-i` com sua evolução com o Docker você verá que existe outras possibilidades para encurtar os comandos,