

Criando dockerfile

1. Acessando a pasta onde será criado o docker.
Comando: `cd /mnt/c/res_docker`
2. Criando a pasta onde será criado o Dockerfile
Comando: `mkdir docker`
3. Acessando a pasta onde ficará o Dockerfile.
Comando: `cd docker`
4. Criando arquivo dockerfile, abrir o editor.
Comando: `vi dockerfile`
5. Utilizando o editor.
Comando:
 - ☐ Edição: tecla i
 - ☐ Sair: tecla esc
 - ☐ Salvar: tecla :wq

Exemplo:

`FROM ubuntu:latest`

`WORKDIR /app`

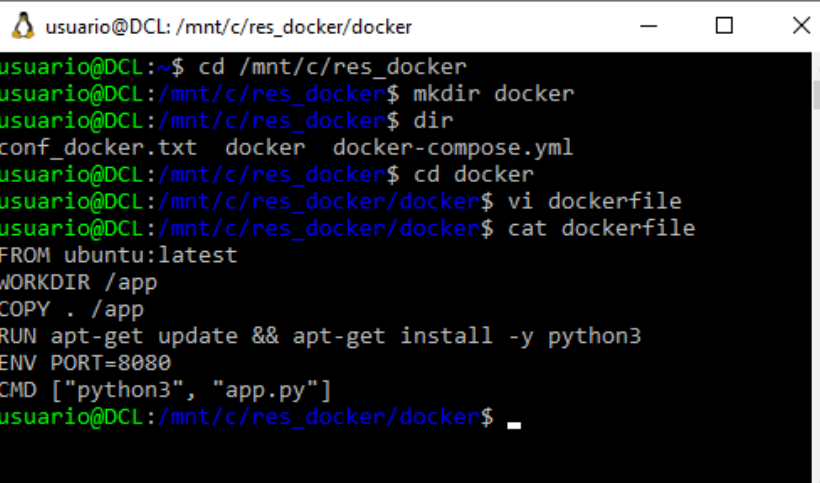
`COPY . /app`

`RUN apt-get update && apt-get install -y python3`

`ENV PORT=8080`

`CMD ["python3", "app.py"]`

6. Para verificar o conteúdo no arquivo.
Comando: `cat dockerfile`



```
usuario@DCL: /mnt/c/res_docker/docker
usuario@DCL:~$ cd /mnt/c/res_docker
usuario@DCL:/mnt/c/res_docker$ mkdir docker
usuario@DCL:/mnt/c/res_docker$ dir
conf_docker.txt  docker  docker-compose.yml
usuario@DCL:/mnt/c/res_docker$ cd docker
usuario@DCL:/mnt/c/res_docker/docker$ vi dockerfile
usuario@DCL:/mnt/c/res_docker/docker$ cat dockerfile
FROM ubuntu:latest
WORKDIR /app
COPY . /app
RUN apt-get update && apt-get install -y python3
ENV PORT=8080
CMD ["python3", "app.py"]
usuario@DCL:/mnt/c/res_docker/docker$
```

Construindo uma nova imagem a partir de um Dockerfile.

1. Criando imagem.

Comando: `docker build -t image_andreia .`

* t - Título

```
usuario@DCL:/mnt/c/res_docker/docker$ docker build -t image_andreia .
[+] Building 13.3s (7/8)
  docker:default
=> [internal] load .dockerignore
    0.2s
=> => transferring context: 2B
    0.1s
=> [internal] load build definition from dockerfile
    0.2s
=> => transferring dockerfile: 172B
    0.1s
=> [internal] load metadata for docker.io/library/ubuntu:latest
    0.0s
[+] Building 20.6s (7/8)
```

2. Para ver a imagem que rodou.

Comando: `docker image ls`

```
usuario@DCL:/mnt/c/res_docker/docker$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
image_andreia	latest	7515590efbdb	2 minutes ago	155MB
postgres	latest	75b7bff7c3ad	2 weeks ago	425MB
php-apache	latest	428ee1770c9e	5 weeks ago	478MB
jenkins/jenkins	lts-jdk17	41e27c2a574b	5 weeks ago	486MB
mongo-express	latest	bbc568e1f48f	6 weeks ago	202MB
ubuntu	latest	b6548eacb063	7 weeks ago	77.8MB
mongo	latest	5acc6a2b3549	7 weeks ago	757MB
mongo	6-jammy	a89d011f83a5	7 weeks ago	690MB
php	8.0-apache	3357132d6ece	8 weeks ago	455MB
nginx	latest	a6bd71f48f68	8 weeks ago	187MB
mysql	latest	f7fdab215ab7	2 months ago	605MB
jupyter/scipy-notebook	latest	ad65fcfebde3	3 months ago	4.14GB
phpmyadmin/phpmyadmin	latest	933569f3a9f6	6 months ago	562MB

```
usuario@DCL:/mnt/c/res_docker/docker$
```

3. Subir imagem.

Comando: `docker run image_andreia:latest`

```
usuario@DCL:/mnt/c/res_docker/docker$ docker run image_andreia:latest
python3: can't open file '/app/app.py': [Errno 2] No such file or directory
usuario@DCL:/mnt/c/res_docker/docker$
```

* Deu erro porque não é possível abrir o arquivo app.py

4. Abaixo é possível ver que ele tenta abrir o container e já fecha:

Comando: `docker ps -a`

```
usuario@DCL:/mnt/c/res_docker/docker$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f6f5bf4e590e	image_andreia:latest	"python3 app.py"	About a minute ago	Exited (2) About a minute ago		eager_ishizaka

```
usuario@DCL:/mnt/c/res_docker/docker$
```

5. Outros comandos:
 - `docker kill $ (docker ps) - para matar container`

Exemplo de Container do Ubuntu

1. Fazendo download da imagem
Comando: `docker pull ubuntu:latest`
2. Subindo a imagem Ubuntu de modo interativo.
Comando: `docker run -it ubuntu`

```
usuario@DCL:/mnt/c/res_docker/docker$ docker run -it ubuntu
root@d53b074b460d:/#
```

Comando: `ls`

```
usuario@DCL:/mnt/c/res_docker/docker$ docker run -it ubuntu
root@d53b074b460d:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@d53b074b460d:/#
```

3. Para sair.
Comando: `exit`
4. Para abrir em segundo plano, usa a variável `-d`.
Comando: `docker run -d ubuntu:latest`

```
usuario@DCL:/mnt/c/res_docker/docker$ docker run -d ubuntu:latest
cc04ebd68b18cfec383d34340e8db8a9e4351d97eb556dde593bab05bcd61a48
usuario@DCL:/mnt/c/res_docker/docker$
```

5. Verificar o container.
Comando: `docker ps -a`

```
usuario@DCL:/mnt/c/res_docker/docker$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
cc04ebd68b18   ubuntu:latest  "/bin/bash"             2 minutes ago  Exited (0) 2 minutes ago          silly_wozniak
d53b074b460d   ubuntu        "/bin/bash"             6 minutes ago  Exited (0) 3 minutes ago          eloquent_burnell
f6f5bf4e590e   image_andreia:latest  "python3 app.py"       25 minutes ago  Exited (2) 25 minutes ago          eager_ishizaka
usuario@DCL:/mnt/c/res_docker/docker$
```

O ubuntu desliga porque não tem interface gráfica dele.

Criando `docker-compose.yml`

1. Acessando a pasta, onde ficará o `docker-compose.yml`.
Comando: `cd /mnt/c/res_docker/`
2. Acessando o editor para criar o arquivo `docker-compose.yml`
Comando: `vi docker compose.yml`

3. No editor é necessário:
 - Teclar i.
 - Digitar a configuração do arquivo.
 - Apertar esc.
 - Digitar :wq

Exemplo de configuração do docker-compose.yml:

```
version: '3'

services:
  postgres:
    image: postgres:latest
    ports:
      - "5432:5432"
    networks:
      - postgres_network
    environment:
      POSTGRES_USER: pguser
      POSTGRES_PASSWORD: Pg@123
  pgAdmin:
    image: dpage/pgadmin4
    ports:
      - "8080:80"
    networks:
      - postgres_network
    environment:
      PGADMIN_DEFAULT_EMAIL: teste@teste.com.br
      PGADMIN_DEFAULT_PASSWORD: PgAdmin@123

networks:
  postgres_network:
    driver: bridge
```

4. Para verificar a configuração.
Comando cat docker-compose.yml
5. Se precisar editar.
Comando: vi \docker-compose.yml
6. Subir o docker-compose
Comando: docker-compose up

```

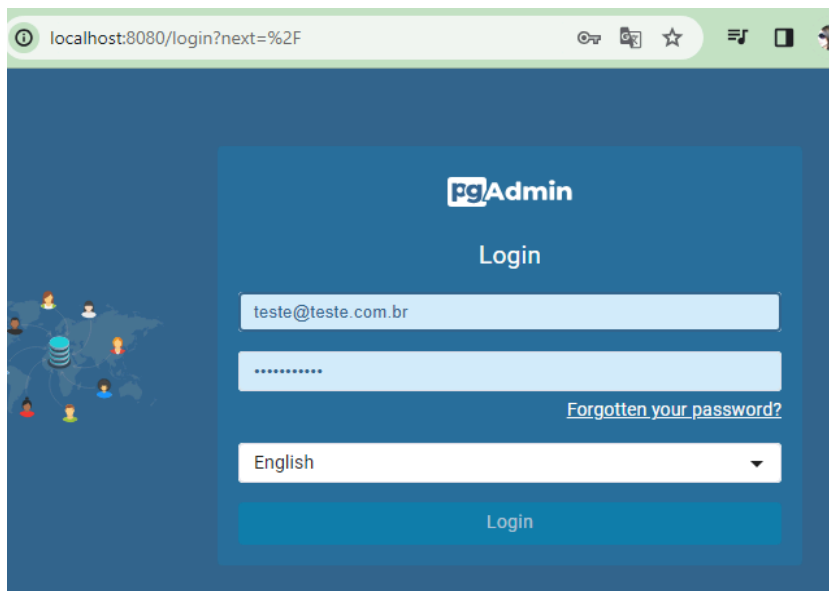
[+] Running 8/17DEFAULT_EMAIL: teste@teste.com.br
# pgAdmin Pulling
# 661f4d5534 Pull complete 61.0s
# 5af5b8b8f408 Pull complete 1.6s
# 2d4a6138fafc Pull complete 55.9s
# 8f0424a8a083 Pull complete 56.3s
# 601f71a5e00 Pull complete 56.5s
# 13f084a003e2 Pull complete 56.8s
# 42a4a4e73a6 Pull complete 56.9s
# 91c1c38004e7 Pull complete 57.0s
# fb06a9fc0ca4 Extracting 491.5kB/7.976MB 57.6s
# 779b87a8a026 Download complete 57.6s
# dc7960d095ac Download complete 57.6s
# 2716e11486a8 Download complete 57.6s
# 54175cb5dd12 Download complete 57.6s
# d5ccf38e9033 Download complete 57.6s
# 7008c2bb0275 Download complete 57.6s
# 618162bf675e Download complete 57.6s
# d5ccf38e9033 Download complete 55.0s
# 7008c2bb0275 Download complete 55.0s
# 618162bf675e Download complete 55.0s
# d5ccf38e9033 Download complete 52.9s
# 7008c2bb0275 Download complete 52.9s
# 618162bf675e Download complete 52.9s
# 2716e11486a8 Download complete 51.6s
# 54175cb5dd12 Download complete 51.6s
# d5ccf38e9033 Download complete 51.6s
# 7008c2bb0275 Download complete 51.6s
# 618162bf675e Download complete 51.6s

```

Cria o pgadmin

7. O serviço, subiu, verificar no navegador:

Link: <http://localhost:8080/>



8. Usuário e senha que estava na configuração.

9. Verificar os serviços.

```

usuario@DCL:~$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
image_andreia       latest             7515509efbdb       2 hours ago        155MB
ubuntu              latest             e34e831650c1       7 days ago         77.0MB
dpape/pgadmin4      latest             76a6f9095825       11 days ago        491MB
postgres            latest             75b7bff7c3ad       2 weeks ago        425MB
php-apache          latest             428eef770c9e       5 weeks ago        478MB
jenkins/jenkins     lts-jdk17         41e27c2a574b       5 weeks ago        486MB
mongo-express       latest             bbc568e1f48f       6 weeks ago        202MB
ubuntu              <none>            b6548eacb063       7 weeks ago        77.8MB
mongo               latest             5acc6a2b3549       7 weeks ago        757MB
mongo               6-jammy           a89d011f83a5       7 weeks ago        690MB
php                 8.0-apache        3357132d6ece       8 weeks ago        455MB
nginx               latest             a6bd71f48f68       8 weeks ago        187MB
mysql               latest             f7fdab215ab7       2 months ago       605MB
jupyter/scipy-notebook latest             ad65fcfebde3       3 months ago       4.14GB
phpmyadmin/phpmyadmin latest             933569f3a9f6       6 months ago       562MB

usuario@DCL:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS              PORTS                               NAMES
34b2812f1c99   postgres:late "docker-entrypoint.s..." 6 minutes ago      Up 6 minutes       0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   res_d
05b141660e61   dpape/pgadmin4 "/entrypoint.sh"         6 minutes ago      Up 6 minutes       443/tcp, 0.0.0.0:8080->80/tcp, :::8080->80/tcp   res_d
0cker-pgAdmin-1 cc04ebd68b18   ubuntu:latest           "/bin/bash"           About an hour ago   Exited (0) About an hour ago               silly
_wozniak        453b074b469d   ubuntu                "/bin/bash"           About an hour ago   Exited (0) About an hour ago               eloqu
ent_burnell     f6f5bf4e590e   image_andreia:latest    "python3 app.py"       2 hours ago        Exited (2) 2 hours ago               eager
_ishizaka

```