

TP – ARQ. NUVEM – AULA FINAL – DOCKER COMPOSE

Anna Beatriz dos Santos Bovo Nunes – 216898

Eduardo Timm Buonacura – 211861

1) CRIAR INDEX.HTML E DOCKERFILE

The screenshot shows a terminal window with two tabs: 'index.html' and 'dockerfile'. The 'index.html' tab displays the content of the file, which is an HTML page with a blue background, white text, and a large 'Oh my god, i'm using Docker!' heading. The 'dockerfile' tab shows the Dockerfile code, which specifies the base image as 'nginx:alpine', removes the default index.html, and copies the local 'index.html' file to the container's /usr/share/nginx/html directory.

```
index.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <title>Serviço web 8080</title>
6   <style>
7     body {
8       background-color: #392f41;
9       color: #bc09f3;
10      font-family: 'Segoe UI', sans-serif;
11      text-align: center;
12      padding-top: 50px;
13    }
14    h1 {
15      font-size: 3rem;
16      margin-bottom: 10px;
17    }
18    p {
19      font-size: 1.5rem;
20    }
21  </style>
22 </head>
23 <body>
24   <h1>Oh my god, i'm using Docker!</h1>
25   <p>Sou uma página Docker. Eu amo a Anna! 😊</p>
26 </body>
27 </html>
28
```

```
dockerfile
1 FROM nginx:alpine
2 RUN rm /usr/share/nginx/html/index.html
3 COPY index.html /usr/share/nginx/html|
```

2) BUILDAR IMAGEM CUSTOMIZADA

```
unisanta@Ubuntu-24:/minhapasta/class-docker-main/TP-FINAL$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
meu-web-app     latest    fc97ccb16f2b   11 minutes ago  52.8MB
nginx          latest    d261fd19cb63   9 days ago   152MB
nginx          alpine   d4918ca78576   2 weeks ago   52.8MB
unisanta@Ubuntu-24:/minhapasta/class-docker-main/TP-FINAL$
```

(rodamos docker build -t meu-web-app . mas precisamos reiniciar a VM por conta da porta)

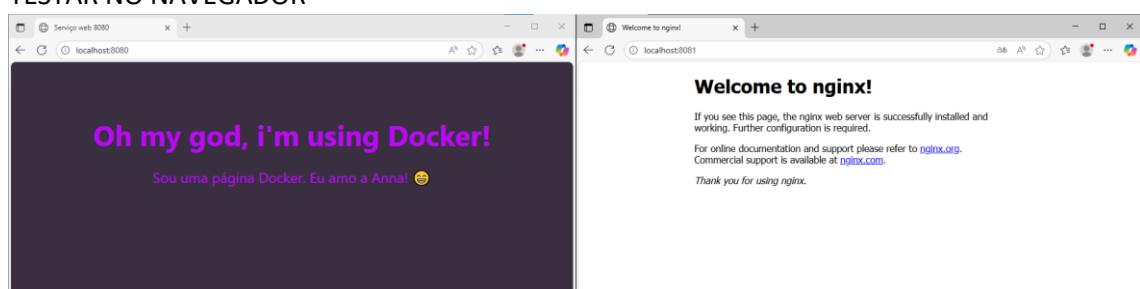
3) CRIAR DOCKER-COMPOSE.YML COM DOIS SERVIÇOS

```
TP-FINAL > docker-compose.yml
 1  version: '3.8'
 2
 3  services:
 4
 5    app_customizado:
 6
 7      build: .
 8      image: meu-web-app:latest
 9      container_name: web_custom_c
10    app_padrao:
11      build: .
12      image: nginx:latest
13      container_name: web_padrao_c
14      ports:
15        - "8080:80"
16
17
```

4) SUBIR TUDO COM DOCKER COMPOSE UP

```
unisanta@Ubuntu-24:/minhapasta/class-docker-main/TP-FINAL$ ls
docker-compose.yml  dockerfile  index.html  log_compose.txt
unisanta@Ubuntu-24:/minhapasta/class-docker-main/TP-FINAL$ docker-compose up -d
Starting web_custom_c ... done
Starting web_padrao_c ... done
unisanta@Ubuntu-24:/minhapasta/class-docker-main/TP-FINAL$ docker-compose ps
      Name            Command           State           Ports
      web_custom_c   /docker-entrypoint.sh  nginx ...   Up    0.0.0.0:8080->80/tcp,:::8080->80/tcp
      web_padrao_c   /docker-entrypoint.sh  nginx ...   Up    0.0.0.0:8081->80/tcp,:::8081->80/tcp
```

5) TESTAR NO NAVEGADOR

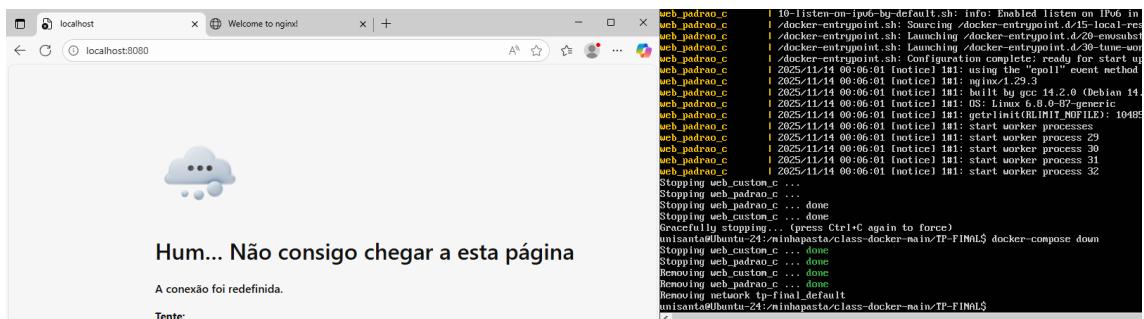


6) EXPLORAR COMANDOS EXTRAS (LOGS, PS, EXEC)

Name	Command	State	Ports
web_custom_c	/docker-entrypoint.sh nginx -g "daemon off;"	Up	0.0.0.0:8080->80/tcp,:::8080->80/tcp
web_padrao_c	/docker-entrypoint.sh nginx -g "daemon off;"	Up	0.0.0.0:8081->80/tcp,:::8081->80/tcp

Exportamos os logs para um TXT:

7) FINALIZAR COM DOCKER-COMPOSE DOWN



RIP nosso site :(