



DEVOPS TOOLS & CLOUD COMPUTING

JOÃO CARLOS MENK

Checkpoint 02

25/04/2024

Dockerfile

Gabriel Sampaio

RM 552342

Gabriel Neves

RM 552244

Livia Freitas

RM 99892

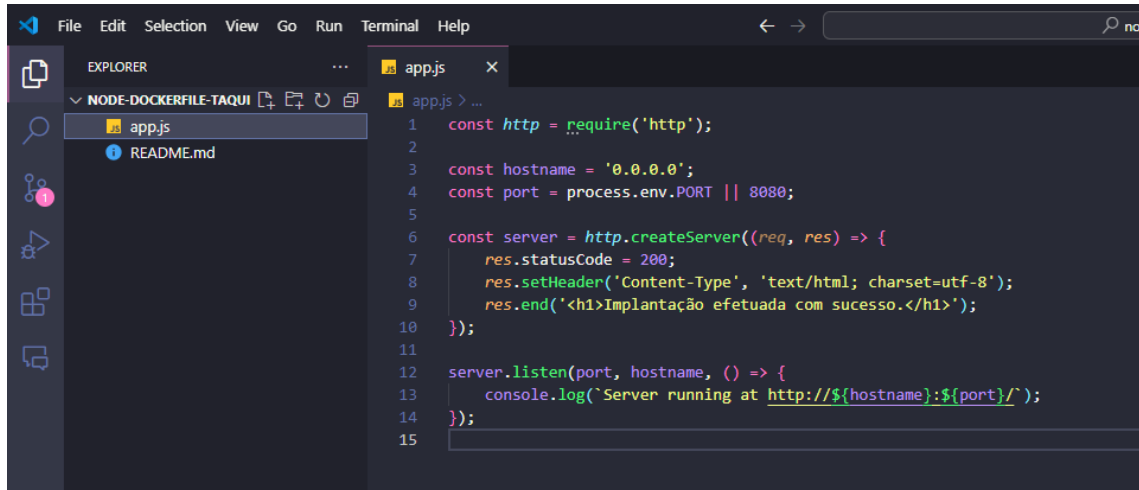
Renato Romeu

RM 551325

3) Desafio Node

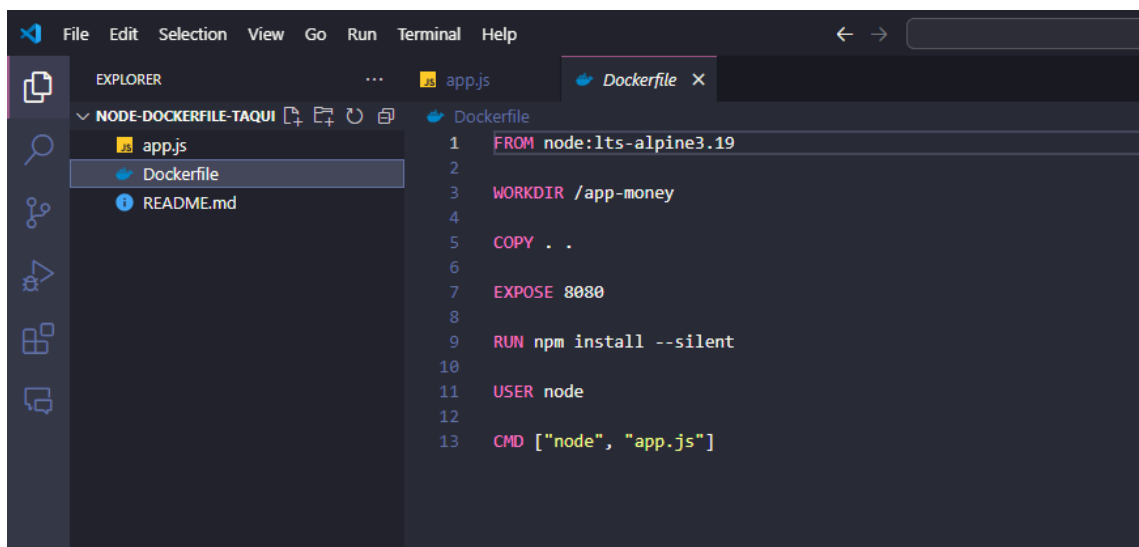
Repositório: <https://github.com/Neveszera/node-dockerfile-taqui.git>

→ Programa node.js que imprime a mensagem solicitada:



```
1  const http = require('http');
2
3  const hostname = '0.0.0.0';
4  const port = process.env.PORT || 8080;
5
6  const server = http.createServer((req, res) => {
7    res.statusCode = 200;
8    res.setHeader('Content-Type', 'text/html; charset=utf-8');
9    res.end('<h1>Implantação efetuada com sucesso.</h1>');
10  });
11
12  server.listen(port, hostname, () => {
13    console.log(`Server running at http://${hostname}:${port}/`);
14  });
15
```

→ Dockerfile:



```
1  FROM node:lts-alpine3.19
2
3  WORKDIR /app-money
4
5  COPY . .
6
7  EXPOSE 8080
8
9  RUN npm install --silent
10
11  USER node
12
13  CMD ["node", "app.js"]
```

➔ Empacotando e criando imagem:

```
C:\Users\GABRI\node-dockerfile-taqui>npm install
up to date, audited 1 package in 627ms
found 0 vulnerabilities

C:\Users\GABRI\node-dockerfile-taqui>docker build -t dimmoney-node .
[+] Building 2.0s (9/9) FINISHED                                docker:default
-> [internal] load build definition from Dockerfile
-> transferring dockerfile: 179B                                0.0s
-> [internal] load metadata for docker.io/library/node:its-alpine3.19 1.6s
-> [internal] load .dockerignore                                  0.0s
-> transferring context: 2B                                       0.0s
-> [1/4] FROM docker.io/library/node:its-alpine3.19@sha256:7a91aa39772e2d72599f374e0b0172be1d86eeb73f1d3f36a4b2 0.0s
-> [internal] load build context                                  0.1s
-> transferring context: 4.59kB                                    0.0s
-> CACHED [2/4] WORKDIR /app-money                             0.0s
-> CACHED [3/4] COPY . .                                          0.0s
-> CACHED [4/4] RUN npm install --silent                         0.0s
-> exporting to image                                             0.1s
-> exporting layers                                              0.0s
-> writing image sha256:e2b228433ed8df1d8bd951f4ace30be08057b72d2106546a63cd8dc728ea7 0.1s
-> naming to docker.io/library/dimmoney-node                    0.0s

What's Next?
 1. Sign in to your Docker account + docker login
 2. View a summary of image vulnerabilities and recommendations + docker scout quickview

C:\Users\GABRI\node-dockerfile-taqui>
```

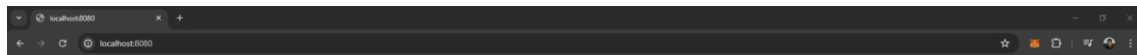
➔ Criando Volume e rodando Container utilizando volume apontando para o diretório:

```
C:\Users\GABRI\node-dockerfile-taqui>docker volume create dimmoneyVolume
dimmoneyVolume

C:\Users\GABRI\node-dockerfile-taqui>docker container run --name dimmoney-node-552244 -d -v dimmoneyVolume:/app-money -p 8080:8080 dimmoney-node
78060fb0f15a5ff2d19a57ede7fc32674193cc90c72097a7381ea3171f1a77a0

C:\Users\GABRI\node-dockerfile-taqui>
```

➔ Aplicação:



Implantação efetuada com sucesso.

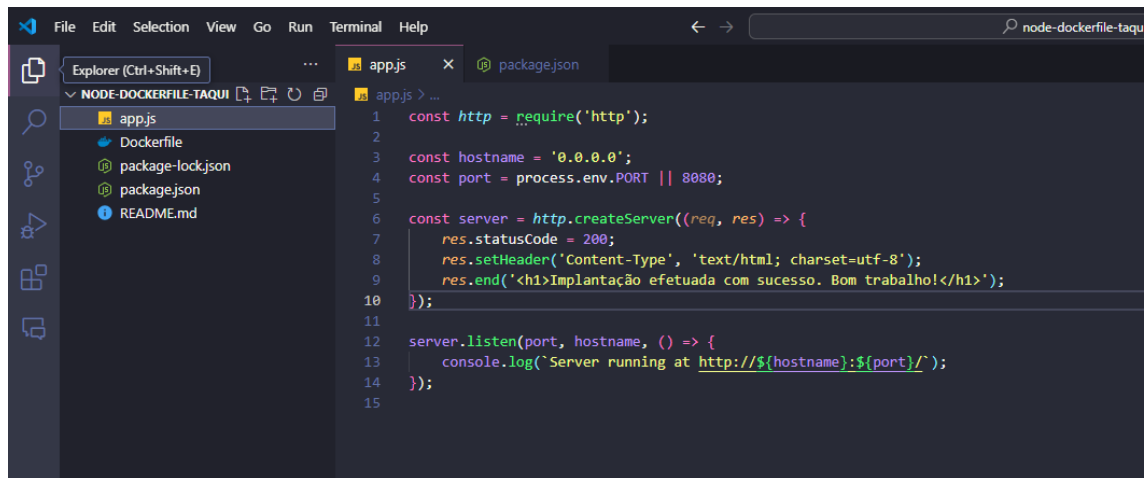
➔ Iniciando redeploy, parando e removendo container:

```
C:\Users\GABRI\node-dockerfile-taqui>docker stop dimmoney-node-552244
dimmoney-node-552244

C:\Users\GABRI\node-dockerfile-taqui>docker rm dimmoney-node-552244
dimmoney-node-552244

C:\Users\GABRI\node-dockerfile-taqui>
```

➔ Alterando a mensagem:

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows the file structure of a project named 'NODE-DOCKERFILE-TAQUI', including files like 'app.js', 'Dockerfile', 'package-lock.json', 'package.json', and 'README.md'. The main editor area displays the 'app.js' file with the following code:

```
1  const http = require('http');
2
3  const hostname = '0.0.0.0';
4  const port = process.env.PORT || 8080;
5
6  const server = http.createServer((req, res) => {
7      res.statusCode = 200;
8      res.setHeader('Content-Type', 'text/html; charset=utf-8');
9      res.end('<h1>Implantação efetuada com sucesso. Bom trabalho!</h1>');
10 });
11
12 server.listen(port, hostname, () => {
13     console.log(`Server running at http://${hostname}:${port}/`);
14 });
15
```

➔ Empacotando e criando novamente a imagem:

```
C:\Users\GABRI\node-dockerfile-taqui>npm install
up to date, audited 1 package in 444ms
found 0 vulnerabilities

C:\Users\GABRI\node-dockerfile-taqui>docker build -t dimmoney-node .
[+] Building 1.3s (9/9) FINISHED
-> [internal] load build definition from Dockerfile
-> [internal] load build definition from Dockerfile
-> [internal] load metadata for docker.io/library/node:its-alpine3.19
-> [internal] load .dockerignore
-> [internal] load build context
-> [1/4] FROM docker.io/library/node:its-alpine3.19@sha256:7a91aa39772e2d4f6fcdad2e2d72599f374e0b0172be1d06eeb73f1d33f36a4b2
-> [internal] load build context
-> [internal] load build context
-> [2/4] WORKDIR /app-money
-> [3/4] COPY . .
-> [4/4] RUN npm install --silent
-> exporting to image
-> exporting layers
-> writing image sha256:2b28413e080d10b051f4ace30bed0057b72d2106546a53c0d1cf28ea7
-> naming to docker.io/library/dimmoney-node

What's Next?
1. Sign in to your Docker account + docker login
2. View a summary of image vulnerabilities and recommendations + docker scout quickview

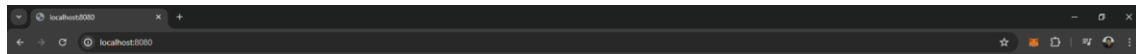
C:\Users\GABRI\node-dockerfile-taqui>
```

➔ Criando novamente o container utilizando o volume apontando para o diretório:

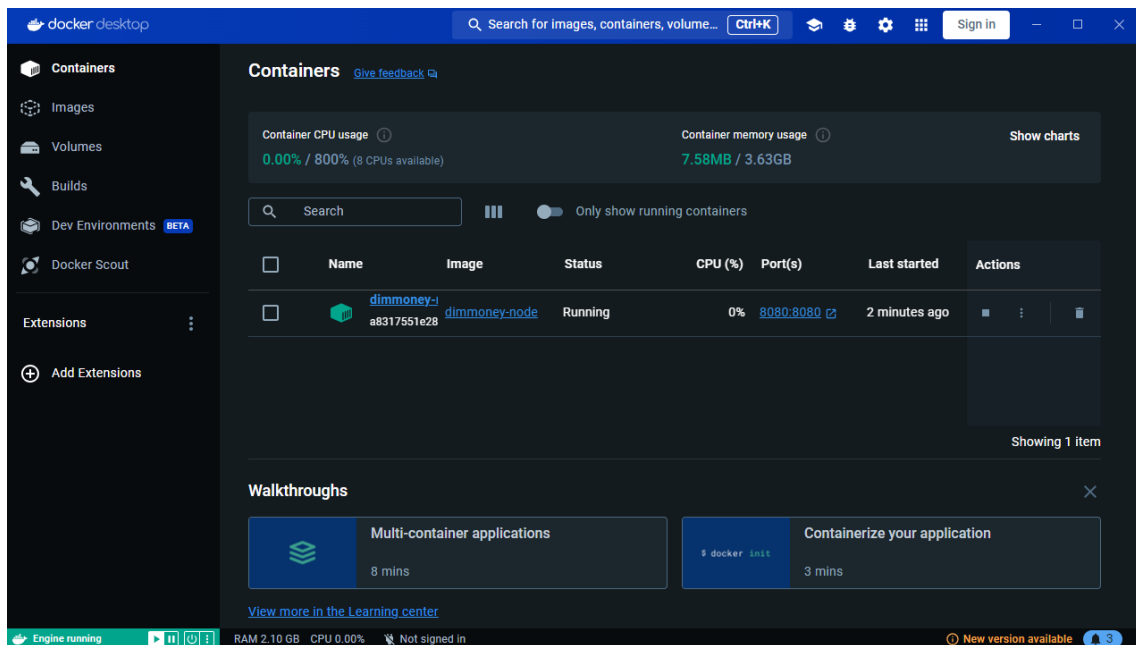
```
C:\Users\GABRI\node-dockerfile-taqui>docker container run --name dimmoney-node-552244 -d -v dimmoneyVolume:/app-money -p 8080:8080 dimmoney-node
a8317551e28cf8297e552920195dee0b55213249a7fc30256518ede007c0d3bb3

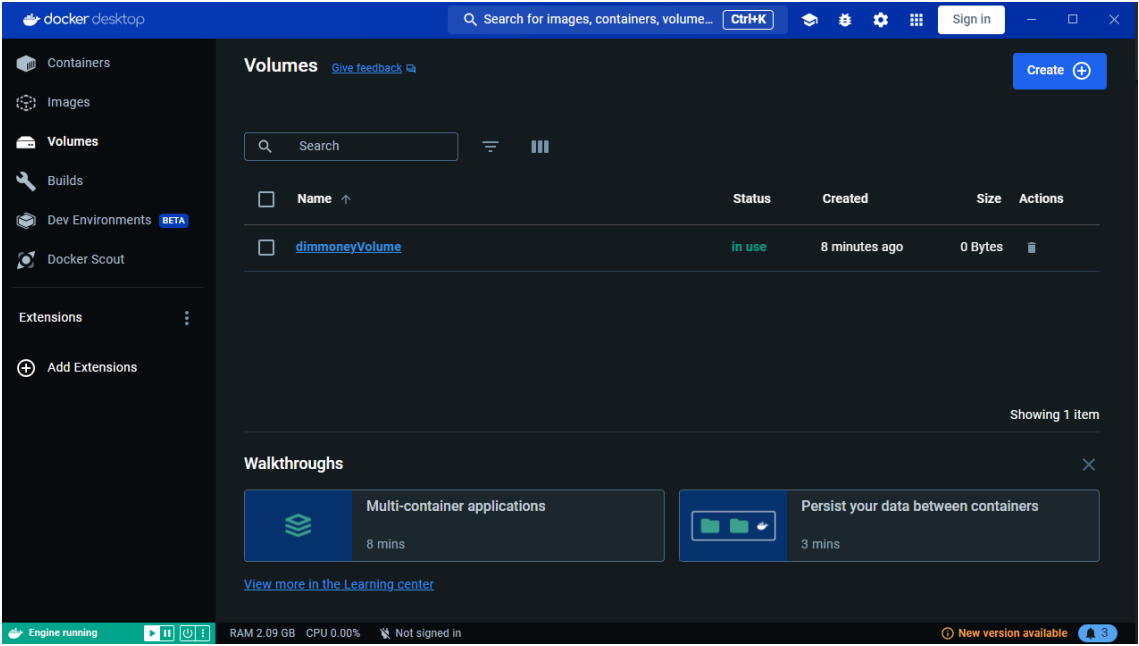
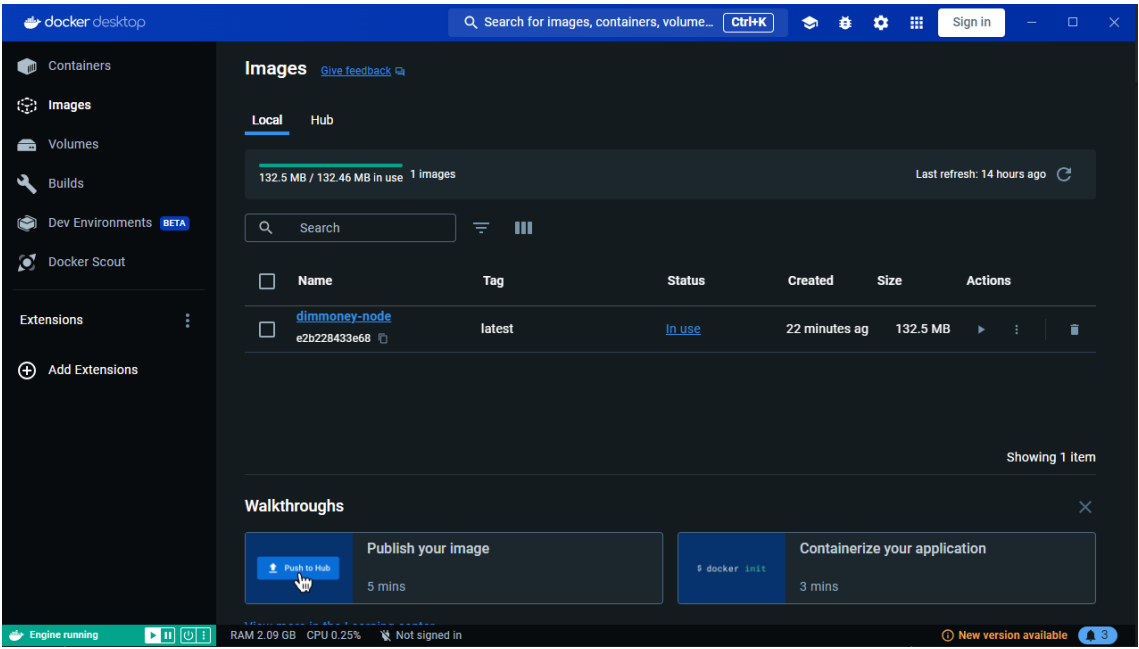
C:\Users\GABRI\node-dockerfile-taqui>
```

→ Aplicação após redeploy:



→ Docker Desktop





Considerações:

Talento vence jogos, mas só o trabalho em equipe vence campeonatos.

Michael Jordan