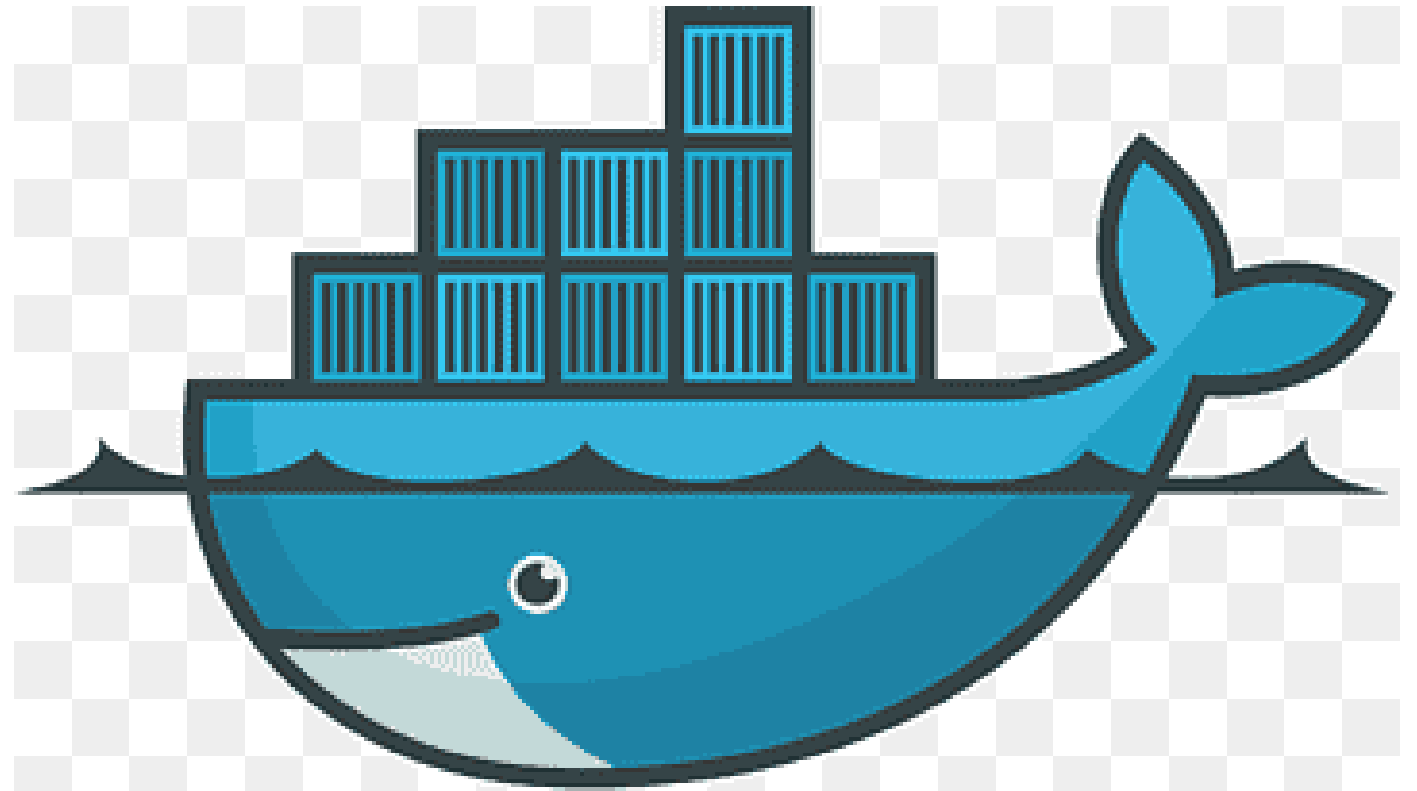


Implementação  
de servidor web  
usando

---



docker

## **Passo a passo de como implementar:**

- 1 - Criar um repositório no github
- 2 - Depois clonar esse repositório para o vscode

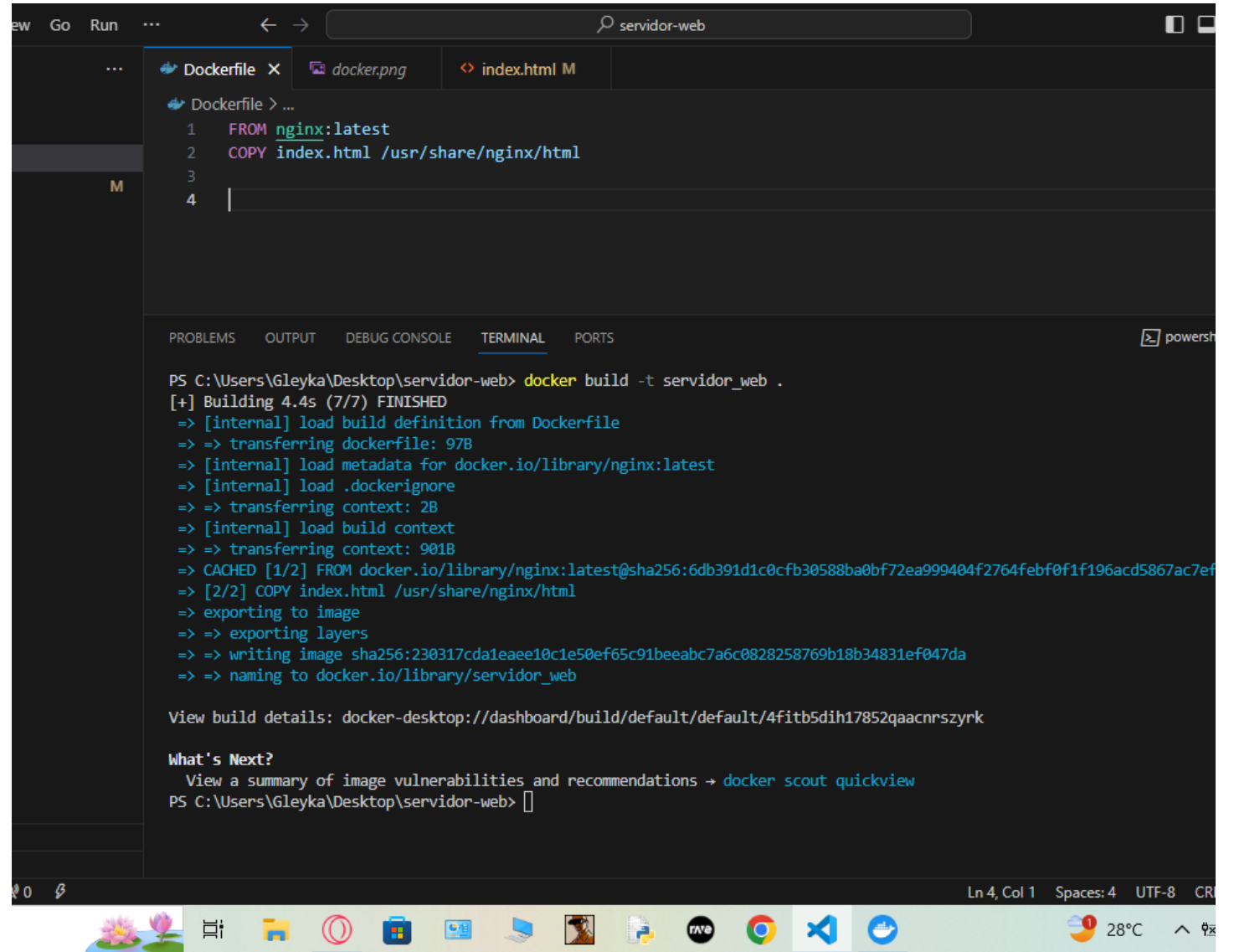
Para clonar:

*git clone <https://github.com/Gleyka11/servidor-web.git>*



Dentro do diretório onde o repositório foi clonado.

Crie o arquivo Dockerfile



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a directory named 'servidor-web' containing a 'Dockerfile', 'docker.png', and 'index.html'. The Dockerfile contains the following content:

```
Dockerfile > ...
1 FROM nginx:latest
2 COPY index.html /usr/share/nginx/html
3
4 |
```

The terminal window shows the output of the command `docker build -t servidor_web .`. The output indicates that the build was successful, showing the steps taken to create the image, including loading the build definition, transferring the Dockerfile, loading metadata for the nginx:latest image, and copying the index.html file to the container's filesystem.

```
PS C:\Users\Gleyka\Desktop\servidor-web> docker build -t servidor_web .
[+] Building 4.4s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 97B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 901B
=> CACHED [1/2] FROM docker.io/library/nginx:latest@sha256:6db391d1c0cfb30588ba0bf72ea999404f2764feb0f1f196acd5867ac7ef
=> [2/2] COPY index.html /usr/share/nginx/html
=> exporting to image
=> => exporting layers
=> => writing image sha256:230317cd41eae10c1e50ef65c91beeabc7a6c0828258769b18b34831ef047da
=> => naming to docker.io/library/servidor_web

View build details: docker-desktop://dashboard/build/default/default/4fitb5dih17852qaacnrszyrk

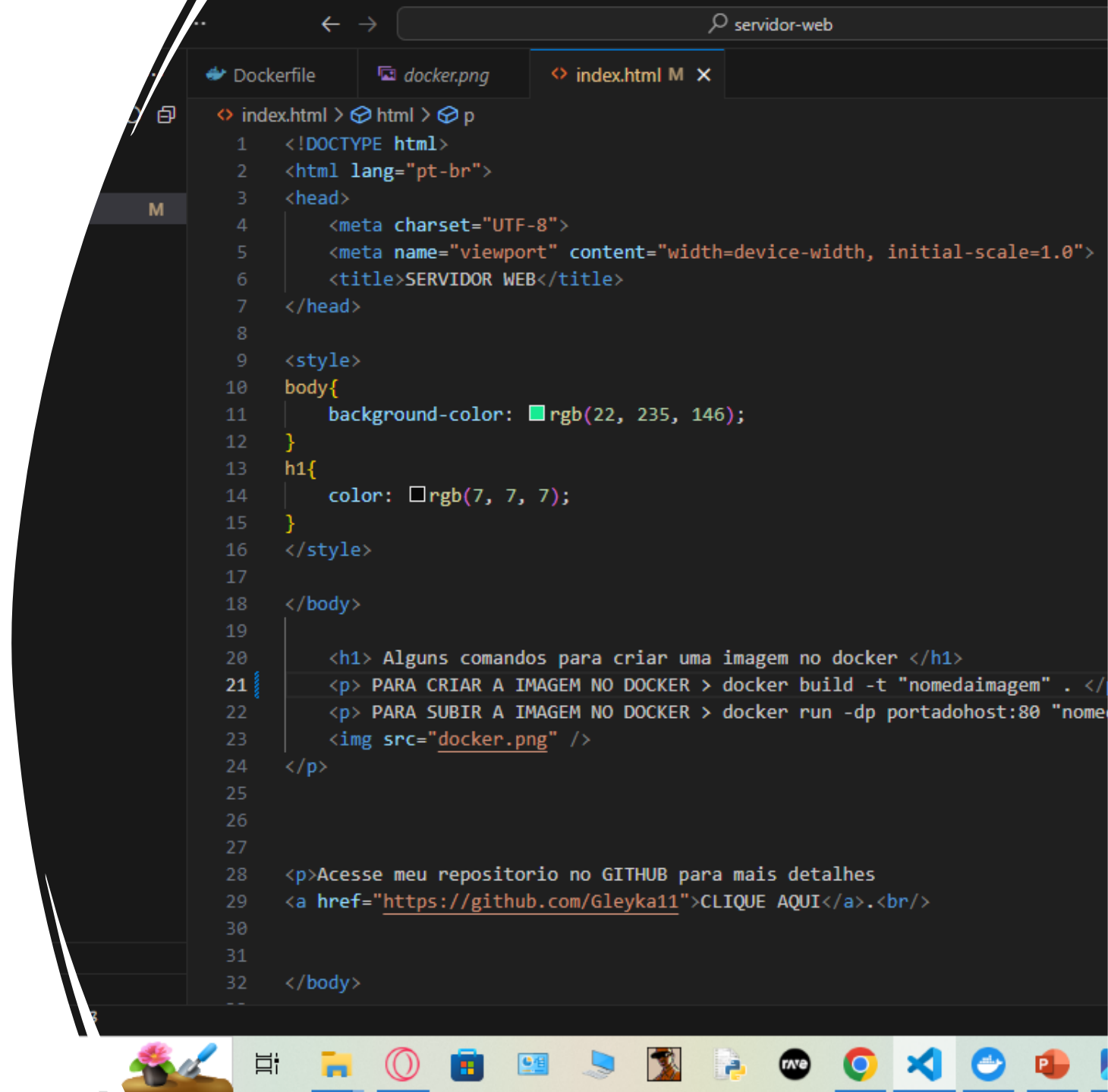
What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview
PS C:\Users\Gleyka\Desktop\servidor-web> |
```

The terminal window also shows the command `docker scout quickview` being entered.

# Agora vamos criar a pagina web

---

- Criei o index.html

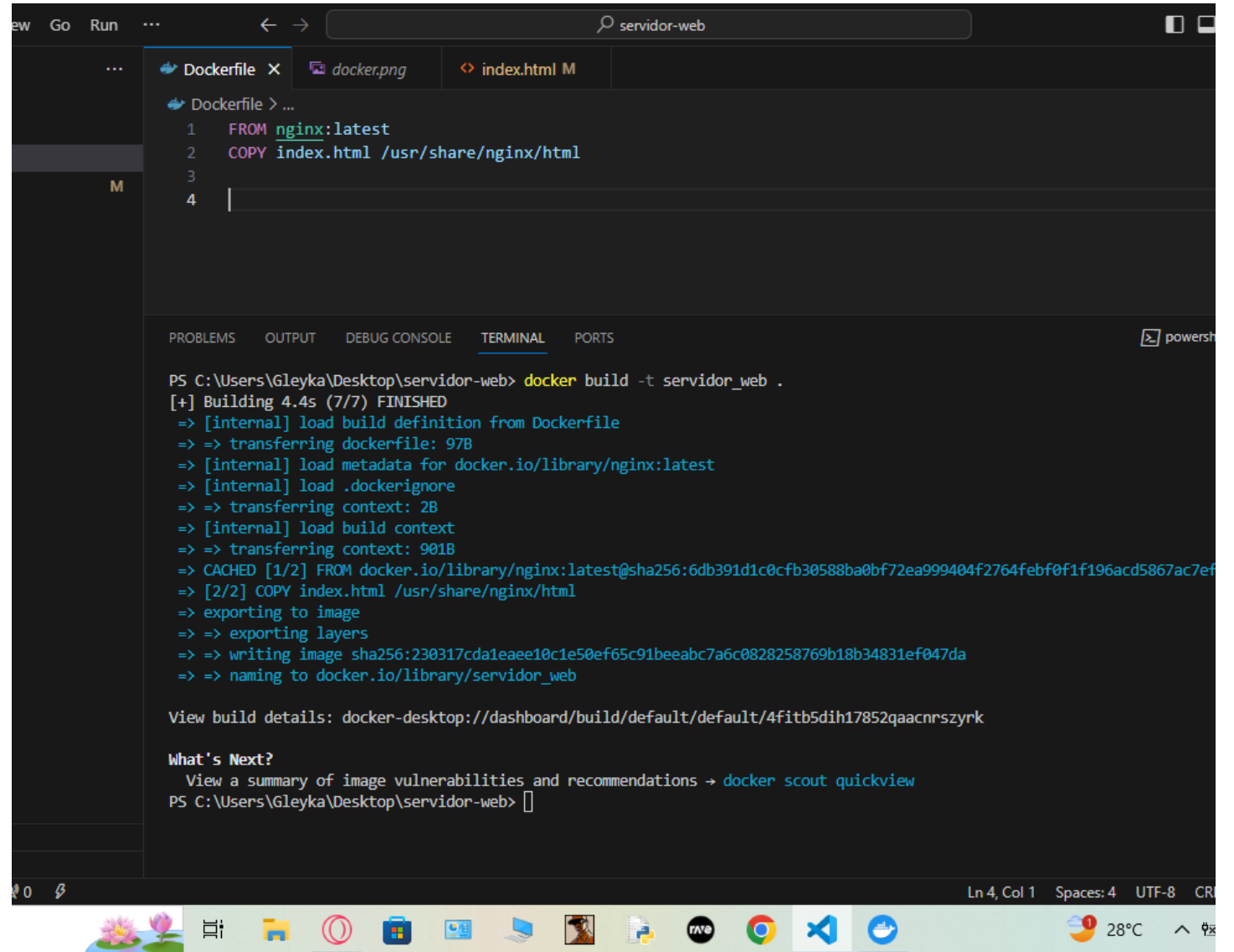


The screenshot shows a code editor with a dark theme. The file explorer on the left shows a folder named 'index.html' containing 'html' and 'p' subfolders. The main editor area displays the content of 'index.html'. The code includes HTML boilerplate with a meta charset of 'UTF-8', a viewport meta tag, and a title 'SERVIDOR WEB'. It also contains CSS for a green background and a black h1. The body contains an h1 heading, a paragraph about Docker commands, an image placeholder for 'docker.png', another paragraph about a GitHub repository, and a link to 'https://github.com/Gleyka11'. The Windows taskbar is visible at the bottom with various application icons.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>SERVIDOR WEB</title>
7 </head>
8
9 <style>
10 body{
11   background-color: #14365c;
12 }
13 h1{
14   color: black;
15 }
16 </style>
17
18 </body>
19
20   <h1> Alguns comandos para criar uma imagem no docker </h1>
21   <p> PARA CRIAR A IMAGEM NO DOCKER > docker build -t "nomedaimagem" . </p>
22   <p> PARA SUBIR A IMAGEM NO DOCKER > docker run -dp portadohost:80 "nome"
23   
24 </p>
25
26
27
28   <p>Acesse meu repositório no GITHUB para mais detalhes
29   <a href="https://github.com/Gleyka11">CLIQUE AQUI</a>.<br/>
30
31
32 </body>
```

# Criando a imagem

Docker build -t “nome da imagem”.



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'servidor-web' containing a 'Dockerfile', 'docker.png', and 'index.html'. The Dockerfile contains the following content:

```
Dockerfile > ...
1 FROM nginx:latest
2 COPY index.html /usr/share/nginx/html
3
4 |
```

The terminal window shows the output of the command `docker build -t servidor_web .`. The output indicates that the build was successful, with the image named `docker.io/library/servidor_web`.

```
PS C:\Users\Gleyka\Desktop\servidor-web> docker build -t servidor_web .
[+] Building 4.4s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 97B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 901B
=> CACHED [1/2] FROM docker.io/library/nginx:latest@sha256:6db391d1c0cfb30588ba0bf72ea999404f2764feb0f1f196acd5867ac7ef
=> [2/2] COPY index.html /usr/share/nginx/html
=> exporting to image
=> => exporting layers
=> => writing image sha256:230317cd41eae10c1e50ef65c91beeabc7a6c0828258769b18b34831ef047da
=> => naming to docker.io/library/servidor_web

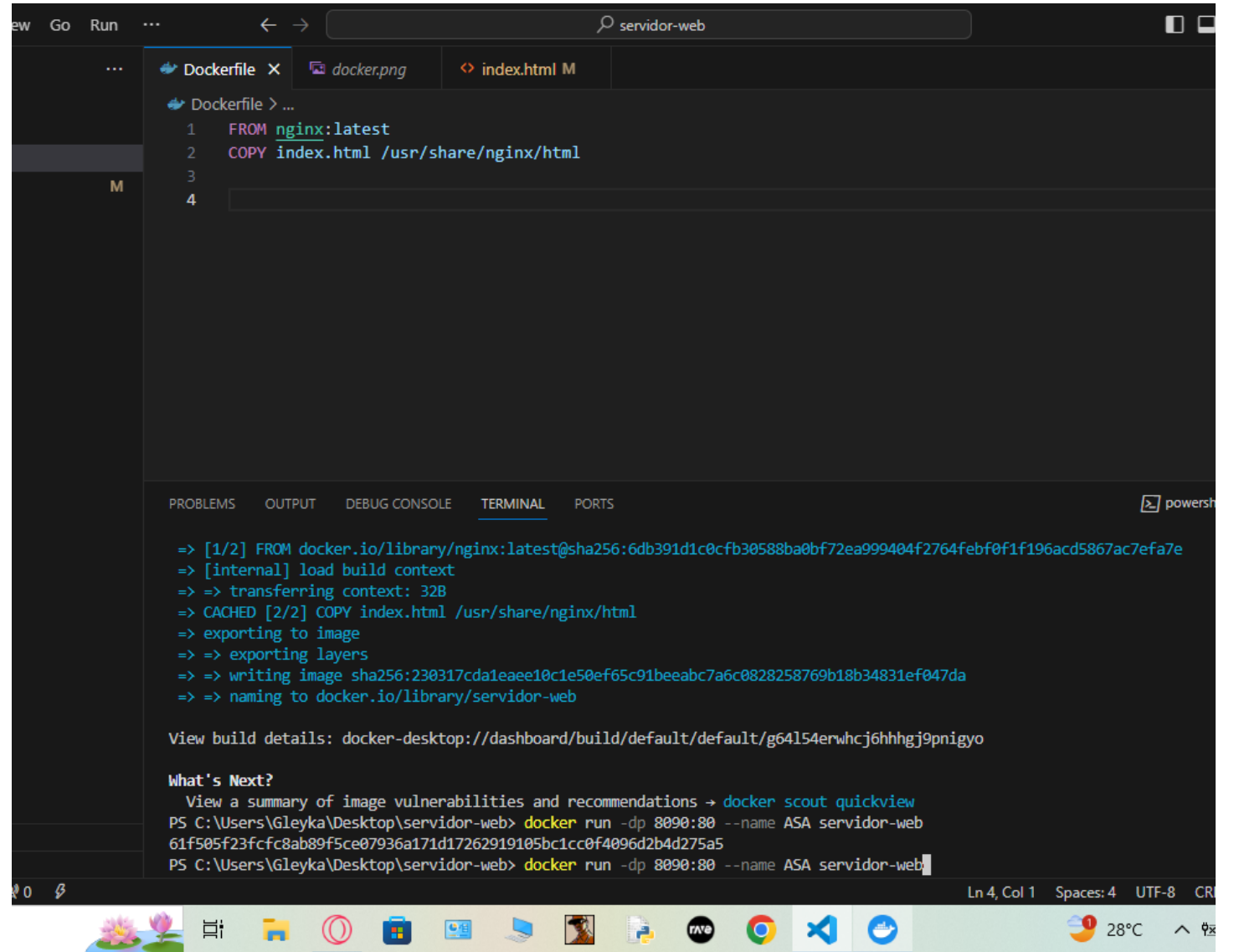
View build details: docker-desktop://dashboard/build/default/default/4fitb5dih17852qaacnrszyrk

What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview
PS C:\Users\Gleyka\Desktop\servidor-web> |
```

The terminal window also shows the command `docker scout quickview` being executed.

# Direcionando a imagem para o container

Docker run -dp 8080:80 “nome da imagem”



The screenshot shows a code editor with a Dockerfile and a terminal window displaying the build process.

**Dockerfile:**

```
1 FROM nginx:latest
2 COPY index.html /usr/share/nginx/html
3
4
```

**Terminal Output:**

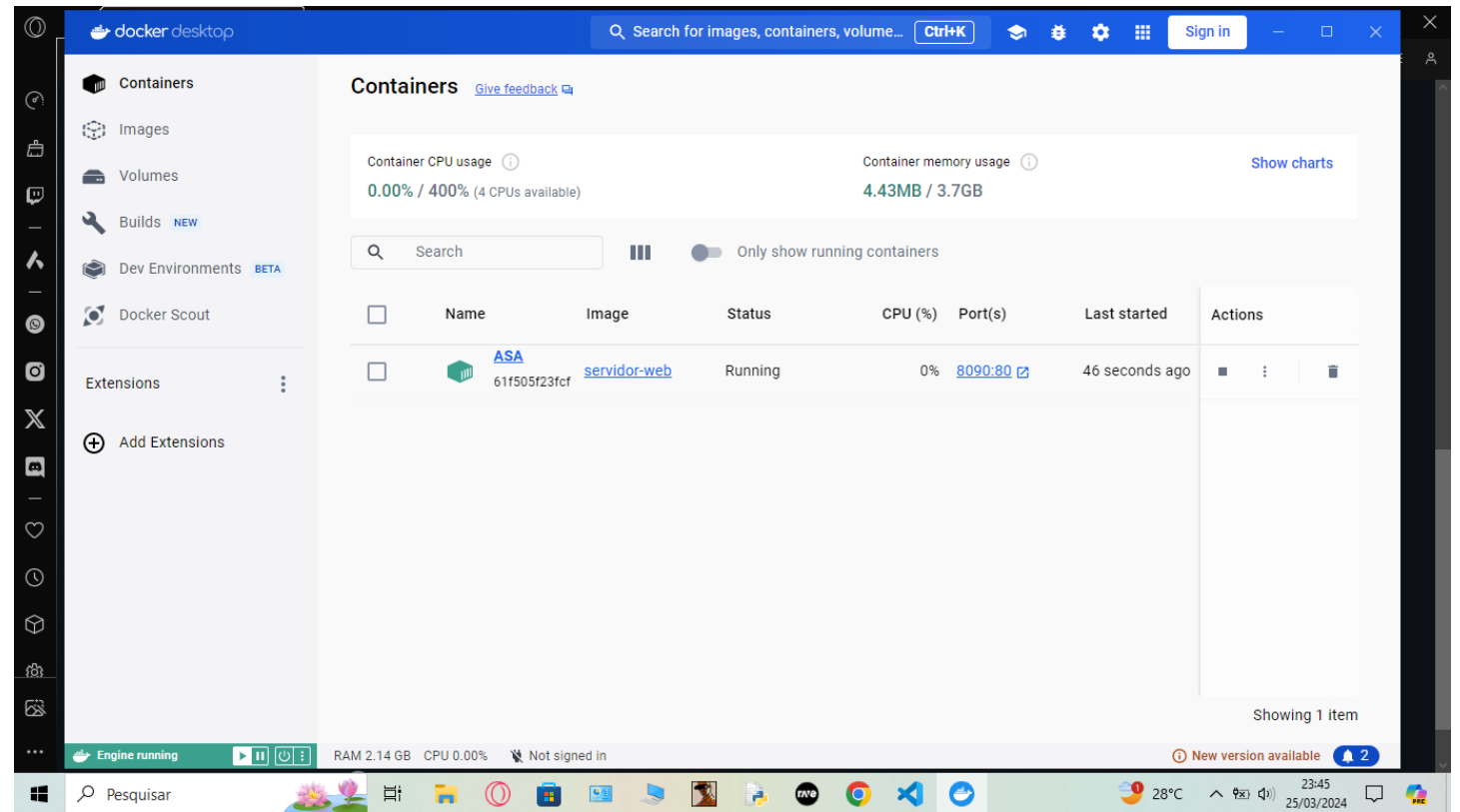
```
=> [1/2] FROM docker.io/library/nginx:latest@sha256:6db391d1c0cfb30588ba0bf72ea999404f2764febf0f1f196acd5867ac7efa7e
=> [internal] load build context
=> => transferring context: 32B
=> CACHED [2/2] COPY index.html /usr/share/nginx/html
=> exporting to image
=> => exporting layers
=> => writing image sha256:230317cda1eae10c1e50ef65c91beeabc7a6c0828258769b18b34831ef047da
=> => naming to docker.io/library/servidor-web

View build details: docker-desktop:///dashboard/build/default/default/g64154erwhcj6hhgj9pnigyo

What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview
PS C:\Users\Gleyka\Desktop\servidor-web> docker run -dp 8090:80 --name ASA servidor-web
61f505f23fcfc8ab89f5ce07936a171d17262919105bc1cc0f4096d2b4d275a5
PS C:\Users\Gleyka\Desktop\servidor-web> docker run -dp 8090:80 --name ASA servidor-web
```

The terminal output shows the successful build of the Docker image 'servidor-web' from the 'nginx:latest' base image. The build process includes loading the build context, transferring the context, copying the index.html file to the nginx directory, and exporting the image. The final image is named 'docker.io/library/servidor-web'. The terminal also shows the command to run the container with the name 'ASA' and port mapping '8090:80'.

# Imagem criada no docker



## Alguns comandos para criar uma imagem no docker

**PARA CRIAR A IMAGEM NO DOCKER**> `docker build -t "nomedaimagem"`.

**PARA SUBIR A IMAGEM NO DOCKER>** `docker run -dp portadohost:80 "nomedaimagem"` (o -d é para executar em background e o -p para publicar a porta do container para o host)



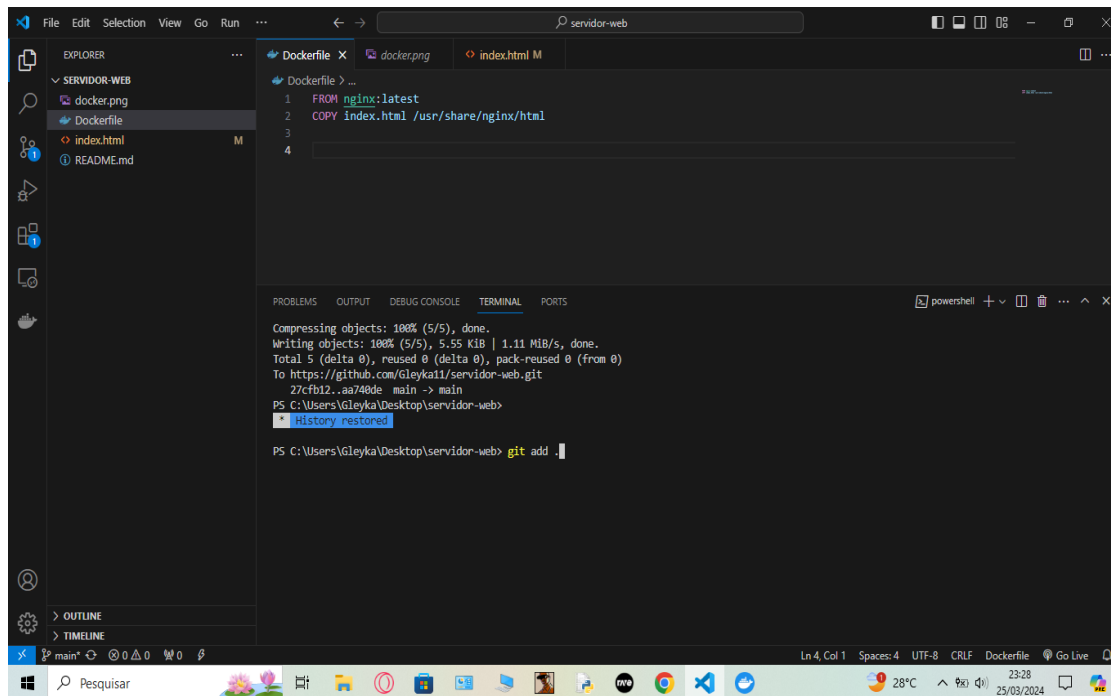
Acesse meu repositório no GITHUB para mais detalhes [CLIQUE AQUI](#).



# UPANDO PARA O GITHUB

Git add .

Git commit -m “mensagem”

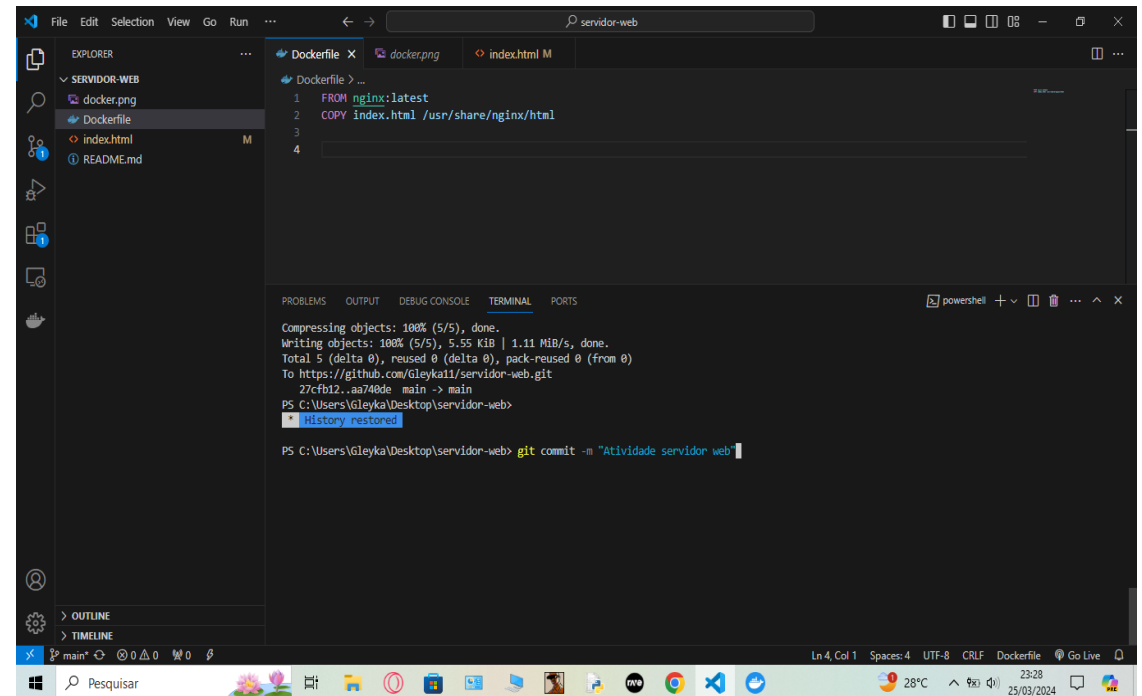


The screenshot shows the Visual Studio Code interface with a project named 'servidor-web'. The Explorer pane on the left shows files: 'docker.png', 'Dockerfile', 'index.html', and 'README.md'. The Dockerfile is open in the editor, showing the following content:

```
1 FROM nginx:latest
2 COPY index.html /usr/share/nginx/html
3
4
```

The terminal at the bottom shows the output of a previous command and the execution of 'git add .':

```
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 5.55 KiB | 1.11 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Gleyka11/servidor-web.git
27cfb12..aa740de main -> main
PS C:\Users\Gleyka\Desktop\servidor-web>
* History restored
PS C:\Users\Gleyka\Desktop\servidor-web> git add .
```



The screenshot shows the Visual Studio Code interface with the same project 'servidor-web'. The Dockerfile is still open in the editor. The terminal at the bottom shows the output of the previous 'git add .' command and the execution of 'git commit -m "Atividade servidor web"':

```
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 5.55 KiB | 1.11 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Gleyka11/servidor-web.git
27cfb12..aa740de main -> main
PS C:\Users\Gleyka\Desktop\servidor-web>
* History restored
PS C:\Users\Gleyka\Desktop\servidor-web> git commit -m "Atividade servidor web"
```

Meu repositório no GITHUB:

<https://github.com/Gleyka11/servidor-web>