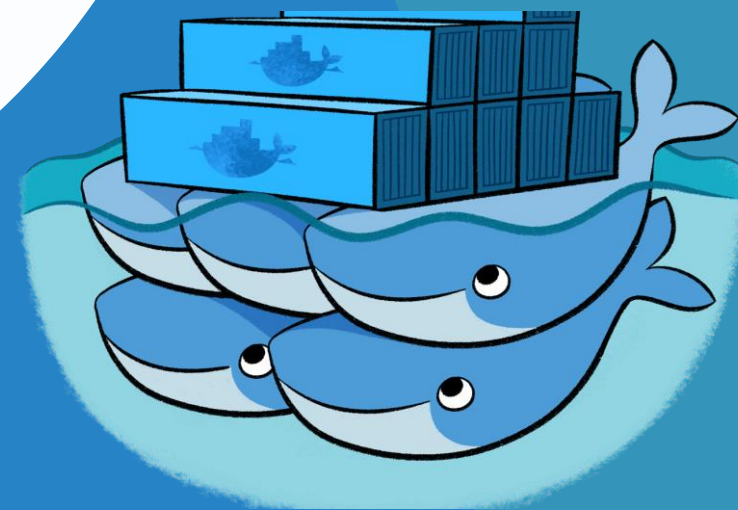




MINICURSO DE DOCKER



REQUISITOS

- Saber se virar num terminal linux;

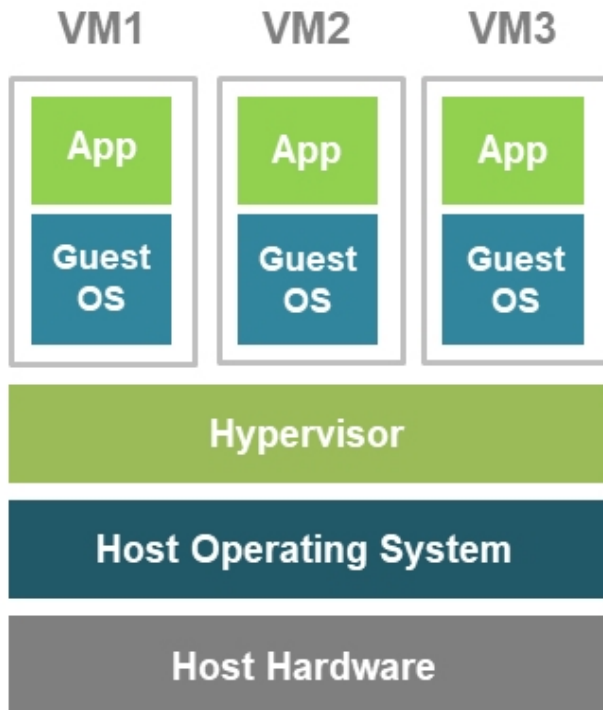
OUR GOAL

- Saber o que é:
 - Container;
 - Imagem;
- Saber os comandos básicos do Docker;
- Saber criar um container;
- Saber criar uma imagem;

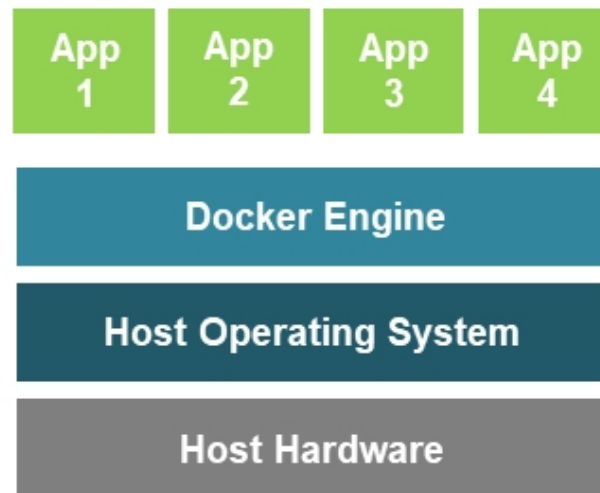


VIRTUALIZAÇÃO X CONTEINERIZAÇÃO

MÁQUINAS VIRTUAIS



CONTÊINERES



VIRTUALIZAÇÃO X CONTAINERIZAÇÃO

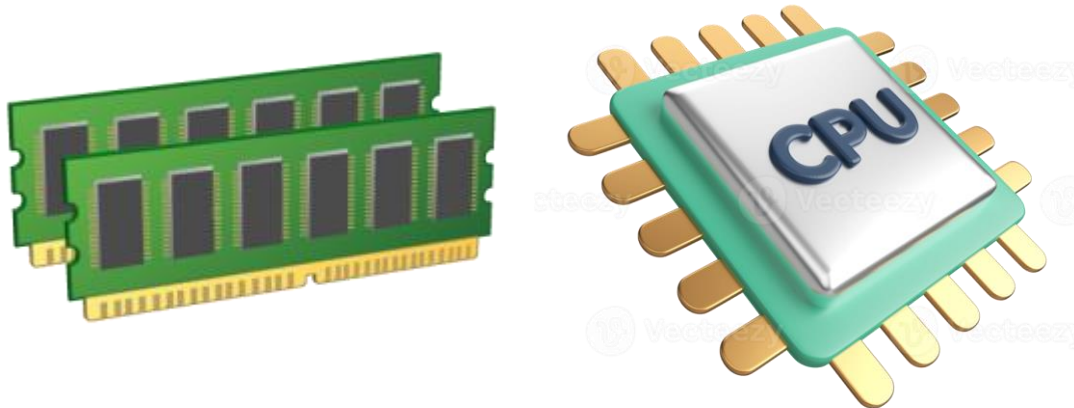
Virtualização

- Precisa de um SO;
- Recursos devem ser reservados;
- Executa em modo virtualizado;



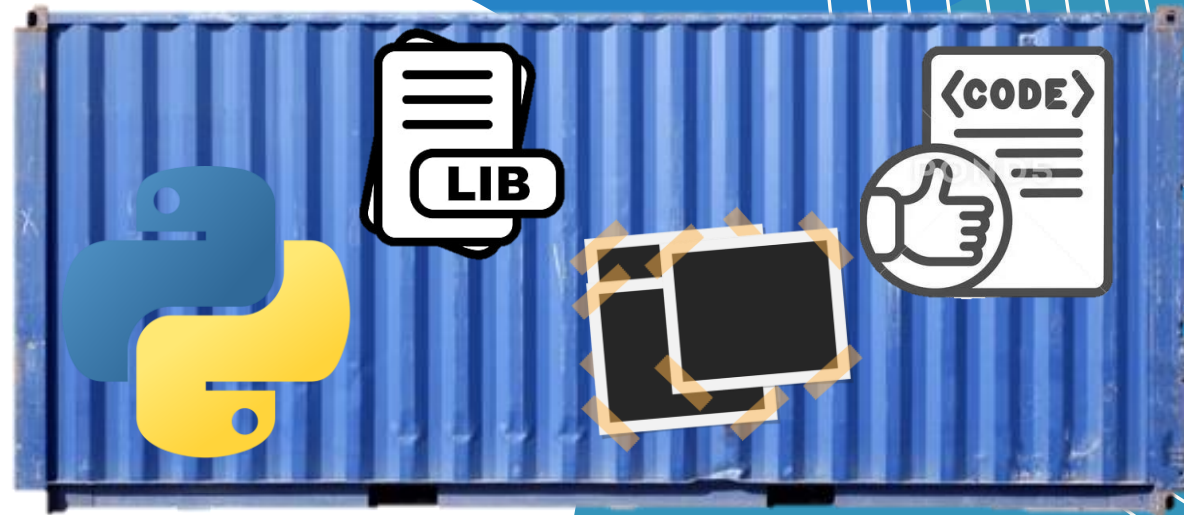
Containerização

- Usa o SO do servidor;
- Os recursos são dinâmicos;
- Não precisa executar virtualizado;



O QUE É UM CONTAINER?

- Aplicação isolada;
- Contém tudo que precisa para ser executada;

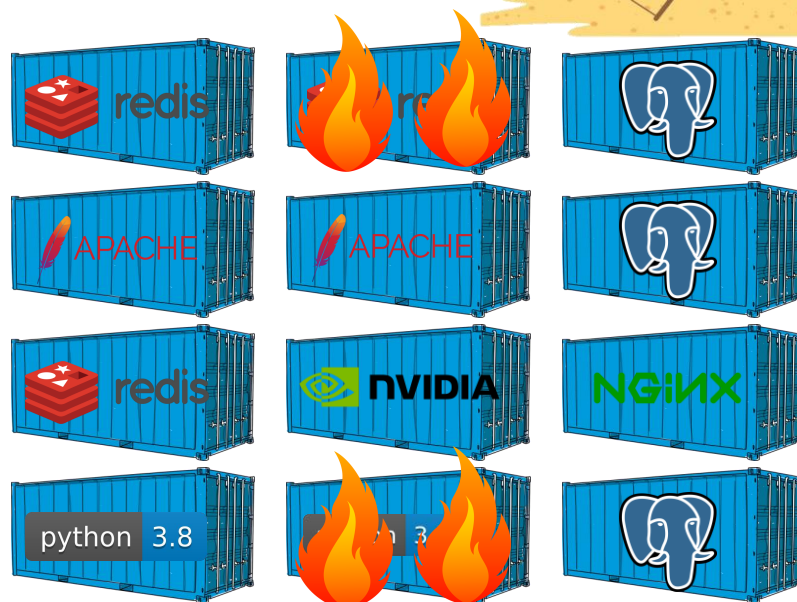


VANTAGENS

Servidor Convencional



Servidor com Docker

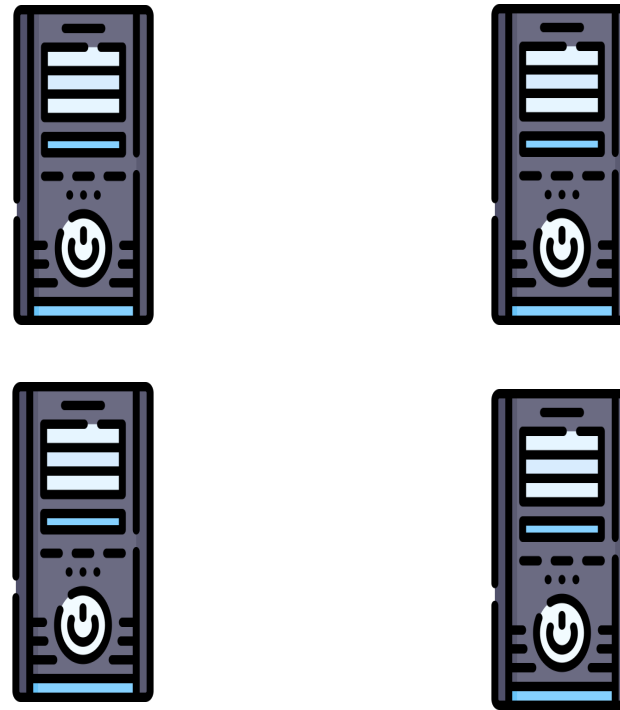


SCALE OUT

Servidor único



Múltiplos servidores



HELLO WORLD

DOCKER RUN HELLO-WORLD

```
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/
```

```
rm -rf / --no-preserve-root
```



SOBRE O COMANDO RUN

DOCKER RUN {flags} {imagem}

Flags:

- it: Interativo**
- d: Desvincular do terminal**
- p {porta}: Expor porta**
- e {variavel}={valor}: Define variável**
- name {nome}: Dar um nome ao container**

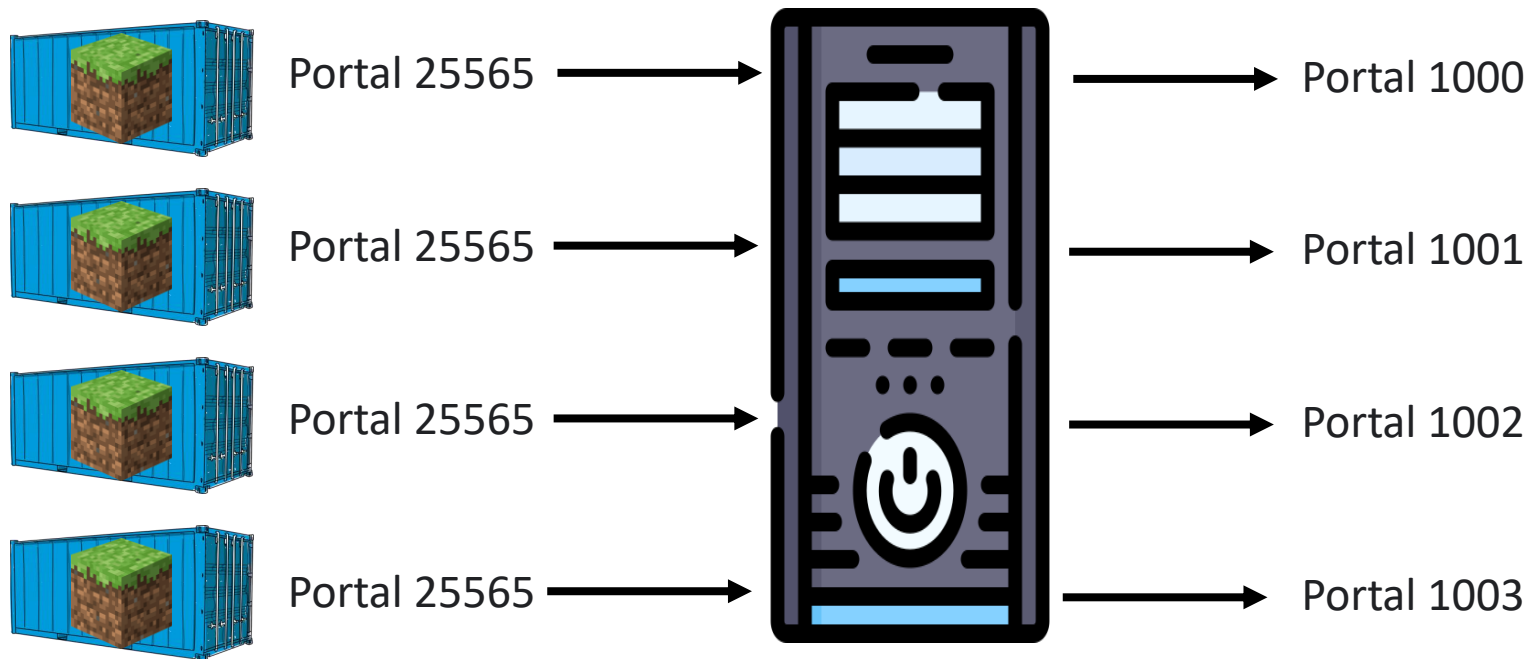
Imagens: {repositório}/{nome}:{versão/tag}

- **ubuntu**
- **library/ubuntu**
- **ubuntu:22.04**
- **library/ubuntu**



PORTAS DE UM CONTAINER

-p {porta_destino}:{porta_origem}



GERENCIANDO CONTAINERS

PRIMEIRA APLICAÇÃO DE VERDADE

DOCKER RUN

-p 80:8443

--name vscode

linuxserver/code-server

DOCKER PS -a

- Listar todos os containers

DOCKER RM {container}



DOCKER PS

```
PS C:\Users\Administrator> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
5635299266b0	vscode_custom	"/init"	15 seconds ago	Up 10 seconds	0.0.0.0:80->8443/tcp	vscode
67692ed49786	halftheopposite/tosios	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes	0.0.0.0:3001->3001/tcp	jogo

```
PS C:\Users\Administrator>
```


PRIMEIRA APLICAÇÃO DE VERDADE

DOCKER RUN -d

-p 80:8443

--name vscode

linuxserver/code-server

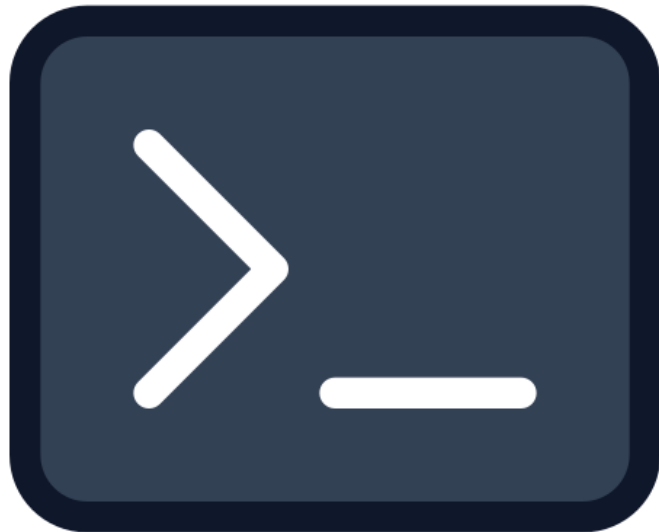
DOCKER PS -a

- **Listar todos os containers**

DOCKER EXEC -it {container} bash



ALGUNS COMANDOS



DOCKER PS

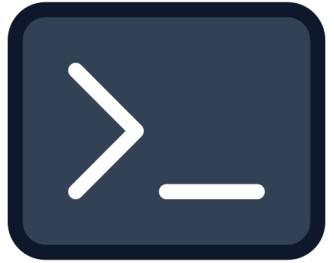
DOCKER PS -a

DOCKER STOP {container}

DOCKER START {container}

DOCKER RM {container}

CUSTOMIZANDO NOSSO CONTAINER








apt update




apt install make gcc git






crazynds/DockerCourse_2024-1

 **DockerCourse_2024-1** Public



 Pin  Unwatch 1  Fork 0  Star 0

 master  1 Branch  0 Tags


 Go to file  Add file  **Code**

Local


Codespaces

 Clone 


HTTPS SSH GitHub CLI


https://github.com/crazynds/DockerCourse_2024-1 


Clone using the web URL.


 Open with GitHub Desktop


Open with Visual Studio


 Download ZIP

 **crazynds** feat: adicionado makefile

 Makefile feat: adicionado ma

 programa.c first commit

 **README**




Add a README


Help people interested in this repository understand


Add a README


About

The necessary files to docker course ministred in UFSM-2024/1

 Activity

 0 stars

 1 watching

 0 forks

Releases

No releases published

[Create a new release](#)

Packages

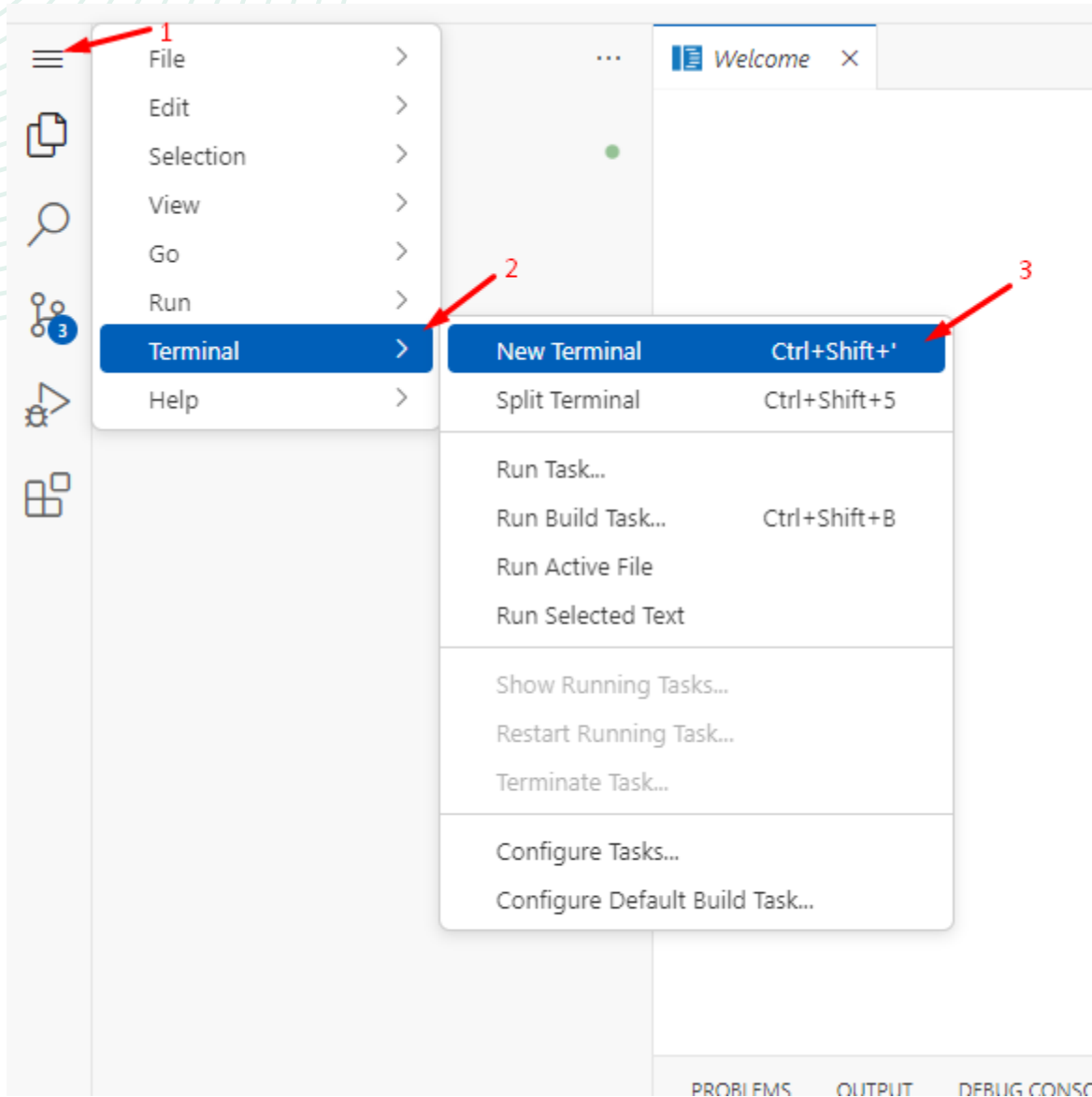
No packages published

[Publish your first package](#)

Languages

C 90.7%

Makefile 9.3%



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

```
abc@94f0ccd32248:~/workspace$ cd DockerCourse_2024-1/
abc@94f0ccd32248:~/workspace/DockerCourse_2024-1$ make
gcc -O2 -Wunused-result -c programa.c -o programa.o
programa.c: In function 'handle_client':
programa.c:128:5: warning: ignoring return value of 'read' declared with
128 |     read(client_sock, buffer, BUFFER_SIZE);
    |     ^~~~~~
programa.c:143:5: warning: ignoring return value of 'write' declared with
143 |     write(client_sock, response, strlen(response));
    |     ^~~~~~
gcc -O2 -Wunused-result -o simple_web_server programa.o
./simple_web_server
Servidor web rodando na porta 80
```



cd DockerCourse_2024-1
make

CRIANDO UMA IMAGEM

DOCKER COMMIT {container} {imagem}

- **Pare e remova o container vscode.**
 - **DOCKER STOP {container}**
 - **DOCKER RM {container}**

DOCKER IMAGES

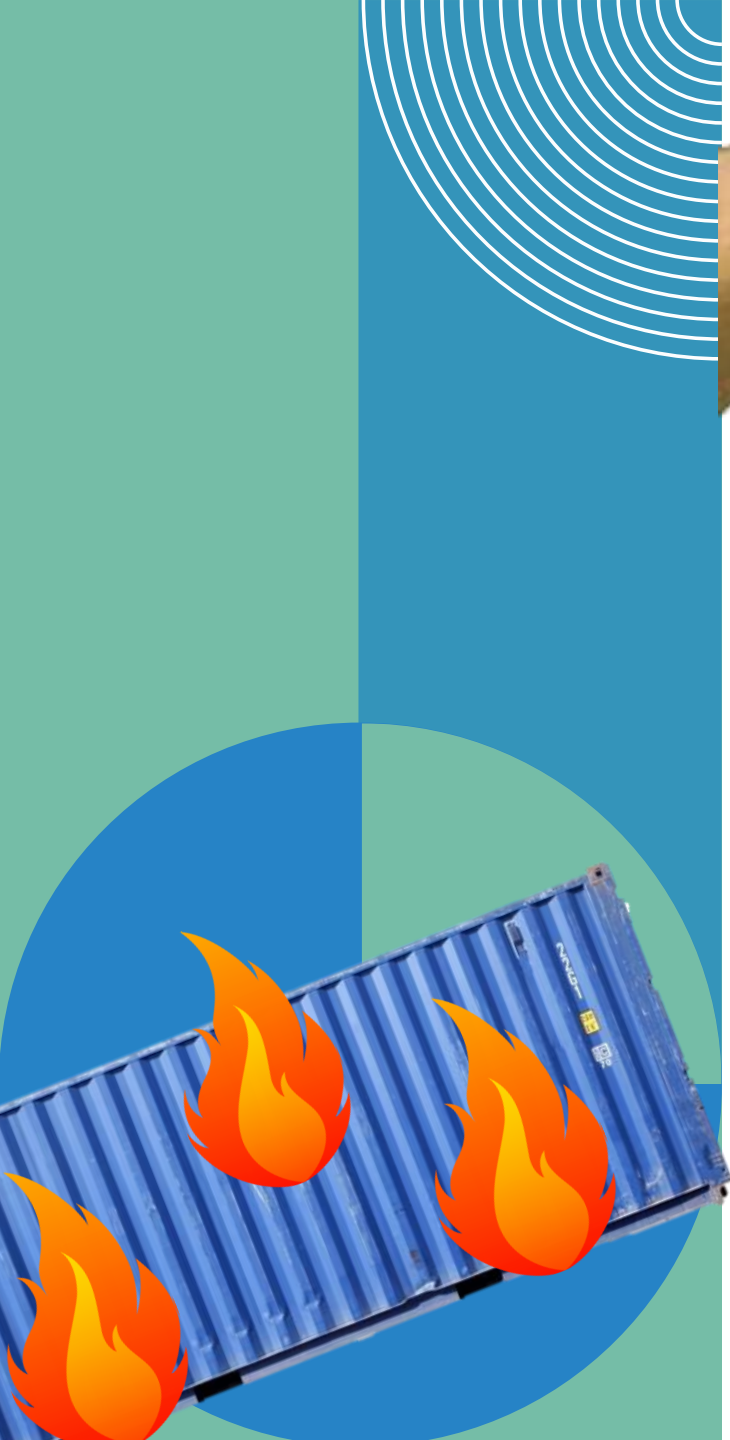




RODANDO NOSSA IMAGEM

```
DOCKER RUN -d  
-p 80:8443  
-p 81:80  
--name vscode  
{imagem}
```

**Todas as nossas modificações foram
salvas na nova imagem.**



MAIS UM EXEMPLO DE USO

HALFTHEOPPOSITE/TOSIOS

DOCKER RUN -d

-p 2000:{porta_container}

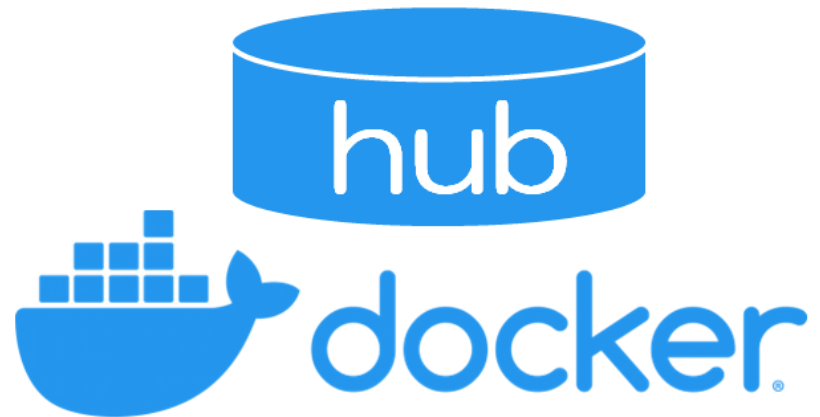
--name jogo

halftheopposite/tosios



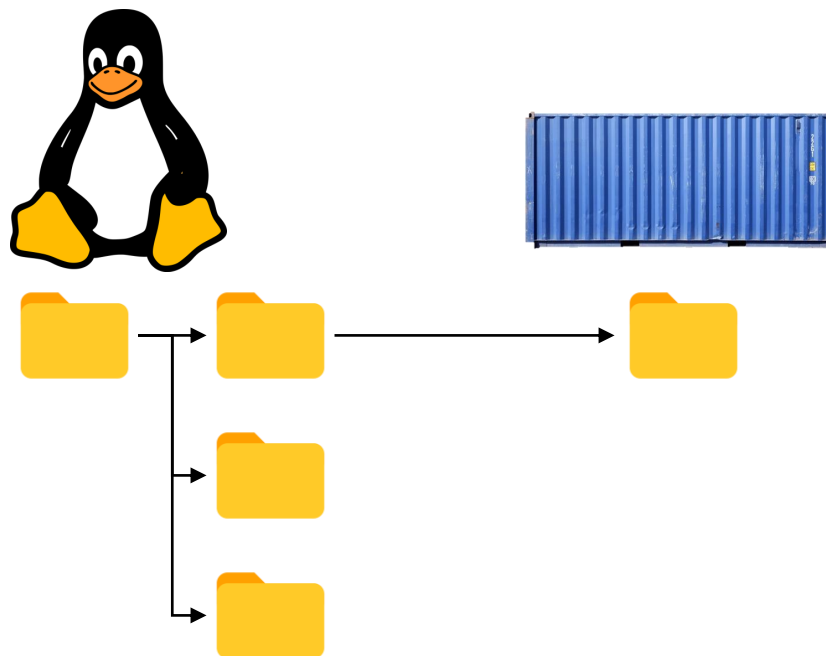
ONDE ENCONTRAR IMAGENS

- <https://hub.docker.com/>



MOUNT

Mapear pasta dentro do container.



EXEMPLOS

-v {path_origen}:{path_container}

-v ./data:/config/workspace/data

-v /home:/home

MOUNT

MKDIR data

CHMOD 0777 data

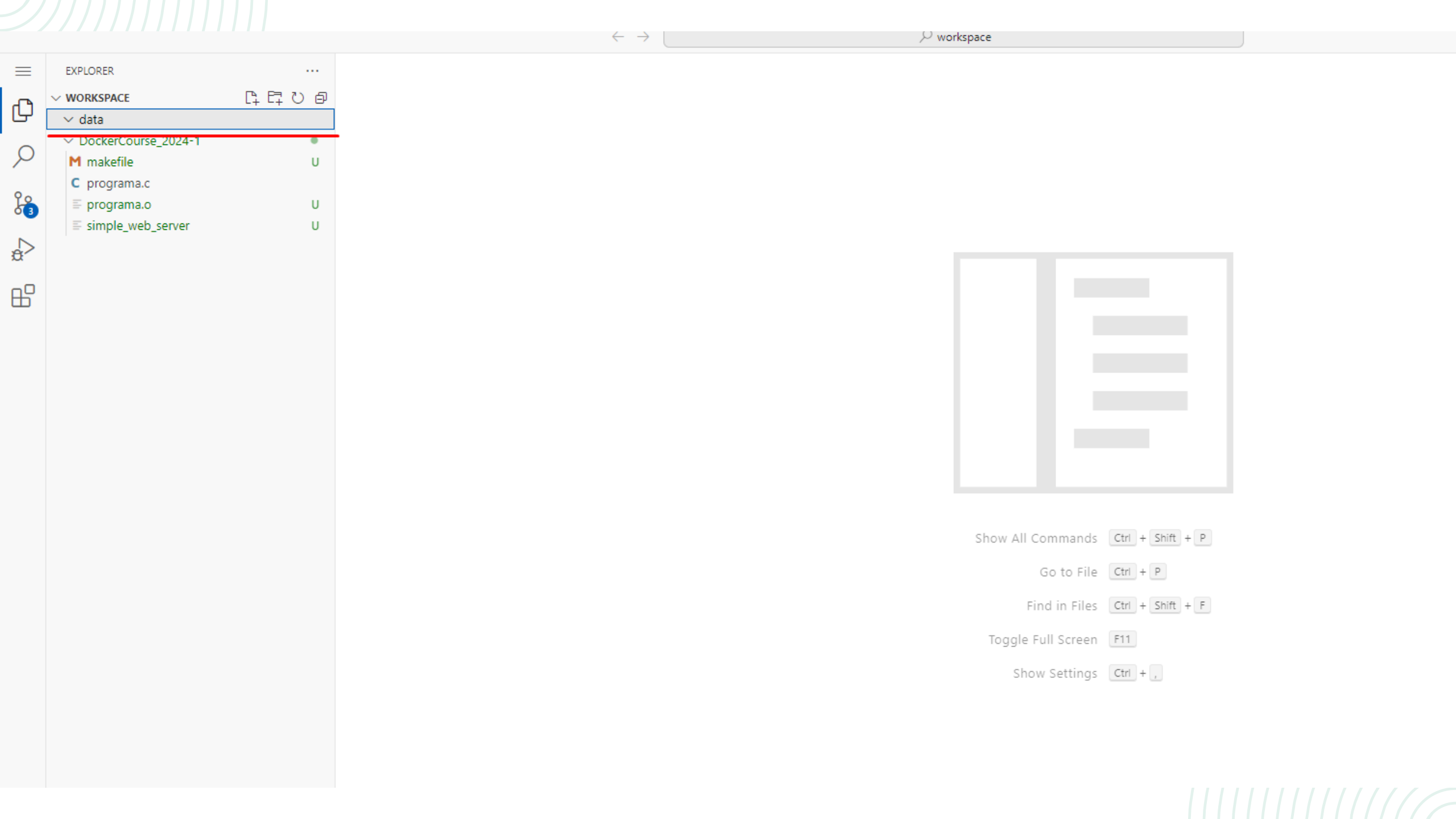
DOCKER RUN -d

-p 80:8443

-v ./data:/config/workspace/data

--name vscode

{imagem}



workspace

EXPLORER

WORKSPACE

data

DockerCourse_2024-1

makefile

programa.c

programa.o

simple_web_server

Show All Commands Ctrl + Shift + P

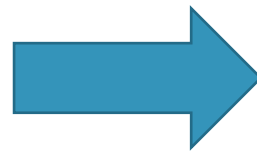
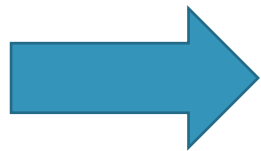
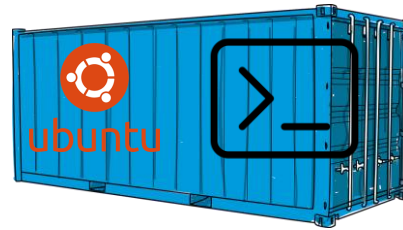
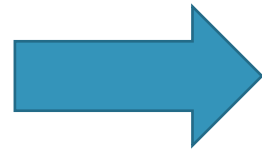
Go to File Ctrl + P

Find in Files Ctrl + Shift + F

Toggle Full Screen F11

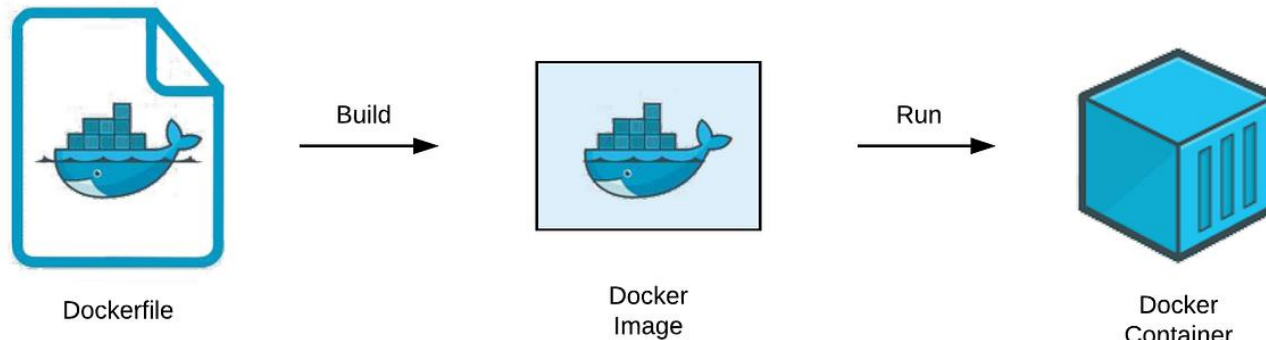
Show Settings Ctrl + ,

CRIANDO IMAGES



CRIANDO IMAGES

- Arquivo de texto;
- Contém o passo a passo da criação da imagem;



 Dockerfile X

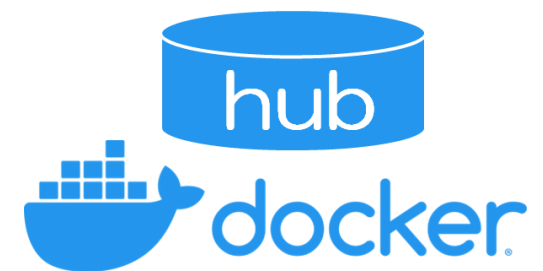
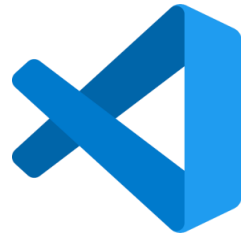
 Dockerfile

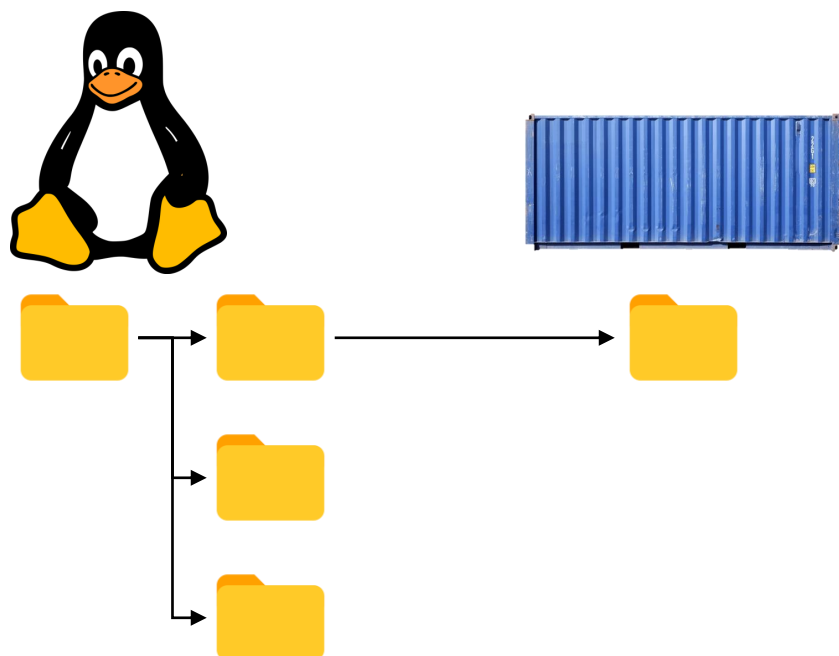
```
1  FROM ubuntu
2
3  RUN apt update && apt install -y git gcc make
4
5  COPY . /app
6
7  RUN make build
8
9  ENTRYPOINT ["/app/run.sh"]
10
11
```



COMANDO FROM

Define qual imagem base usar.





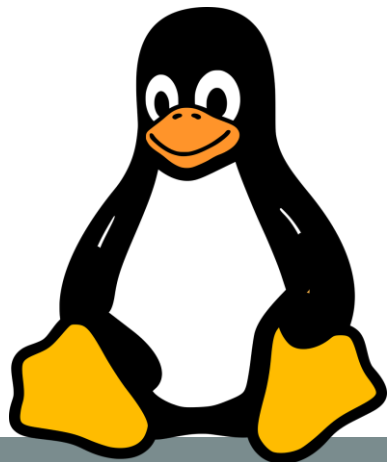
COMANDO COPY

- Copia pastas e arquivos para a imagem;



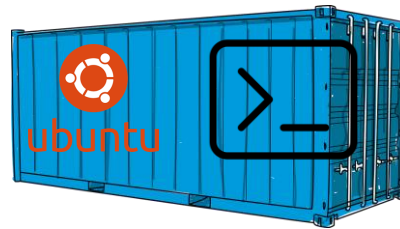
COMANDO RUN

- Executa um comando;



COMANDO ENTRYPOINT

Define qual comando vai ser executado ao
iniciar o container;



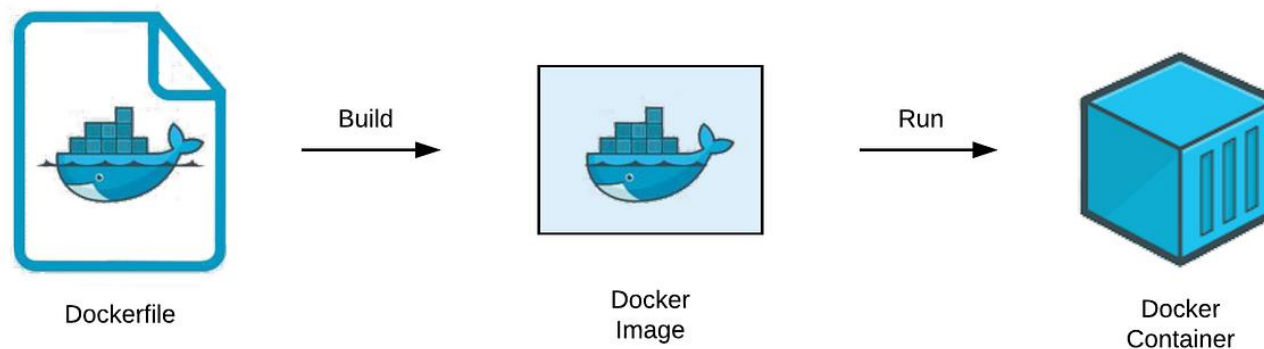
 Dockerfile X

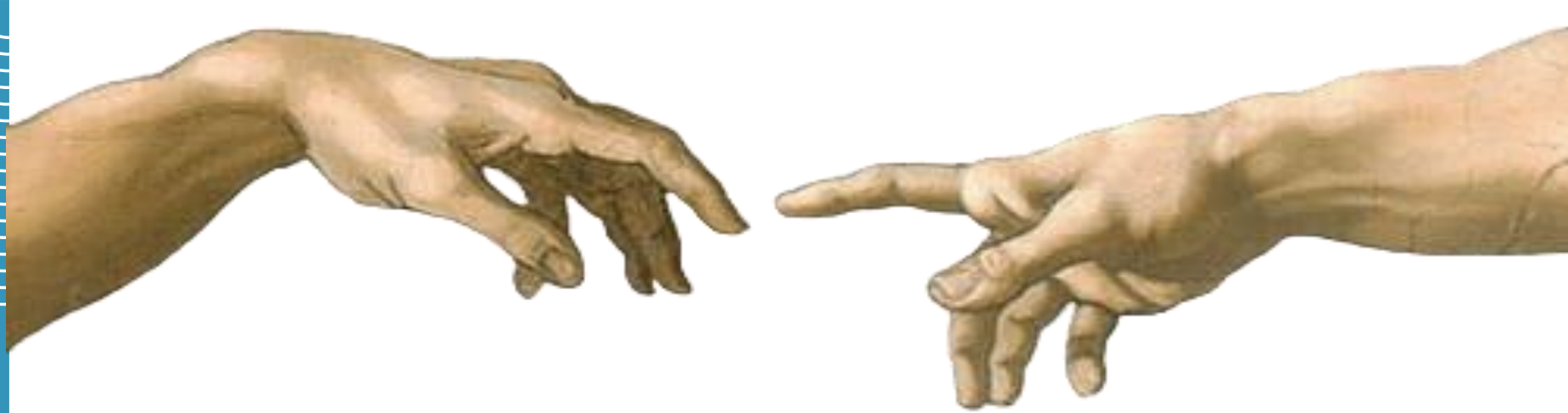
 Dockerfile

```
1  FROM ubuntu
2
3  RUN apt update && apt install -y git gcc make
4
5  COPY . /app
6
7  RUN make build
8
9  ENTRYPOINT ["/app/run.sh"]
10
11
```


PROCESSO DE BUILD

DOCKER BUILD . -t {imagem}





RODANDO NOSSA IMAGEM

```
DOCKER RUN -d  
-p 81:80  
--name vscode  
{imagem}
```

