

Fundamentos de DevOps

Laboratório 04 - Contêineres com Docker

Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.cruz@alumni.usp.br

 linkedin.com/in/guijac

Laboratório

1. Criar uma conta Docker: [Signup | Docker](#);
2. Instalar o [Docker Desktop](#) em seu computador;
3. Executar o comando `docker run hello-world`;
4. Executar uma imagem do servidor NGINX, na porta 80;
5. Testar o acesso em localhost, através do seu navegador;
6. Localizar e alterar o arquivo `index.html` do NGINX;
7. Mapear o arquivo `index.html` NGINX em seu computador, através do comando:

```
docker run -p 80:80 -d -v ${PWD}:/usr/share/nginx/html nginx
```

8. Entregar arquivos no Gitlab.

Laboratório

- Para validar a instalação do Docker, vamos executar nosso primeiro contêiner, um “Hello World”;
- Lembrem-se que o comando “docker run” baixa a imagem, caso ela não exista localmente, leia atentamente todo texto do console, que detalha o que vimos na teoria:

```
docker run hello-world
```

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Already exists
Digest: sha256:6e8b6f026e0b9c419ea0fd02d3905dd0952ad1feea67543f525c73a0a790fefb
Status: Downloaded newer image for hello-world:latest
```

```
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)

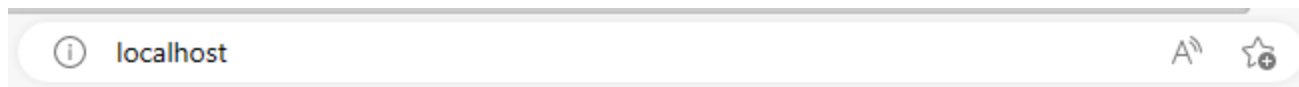
Laboratório

- Agora, iremos executar uma imagem do NGINX, um servidor leve que pode hospedar páginas HTML;
- Notem que temos um novo parâmetro no “docker run”:
 - **-p**: especifica a porta de exposição do contêiner no formato: PORTAL_LOCAL:PORTA_CONTEINER.

```
docker run -p 80:80 nginx
Unable to find image 'nginx' locally
alpine: Pulling from library/nginx
.
.
.
2023/03/08 12:56:19 [notice] 1#1: getrlimit(RLIMIT_NOFILE):
1048576:1048576
2023/03/08 12:56:19 [notice] 1#1: start worker processes
```

Laboratório

- Acesse a URL <http://localhost> e abra a página inicial do NGIX:



Welcome to nginx!

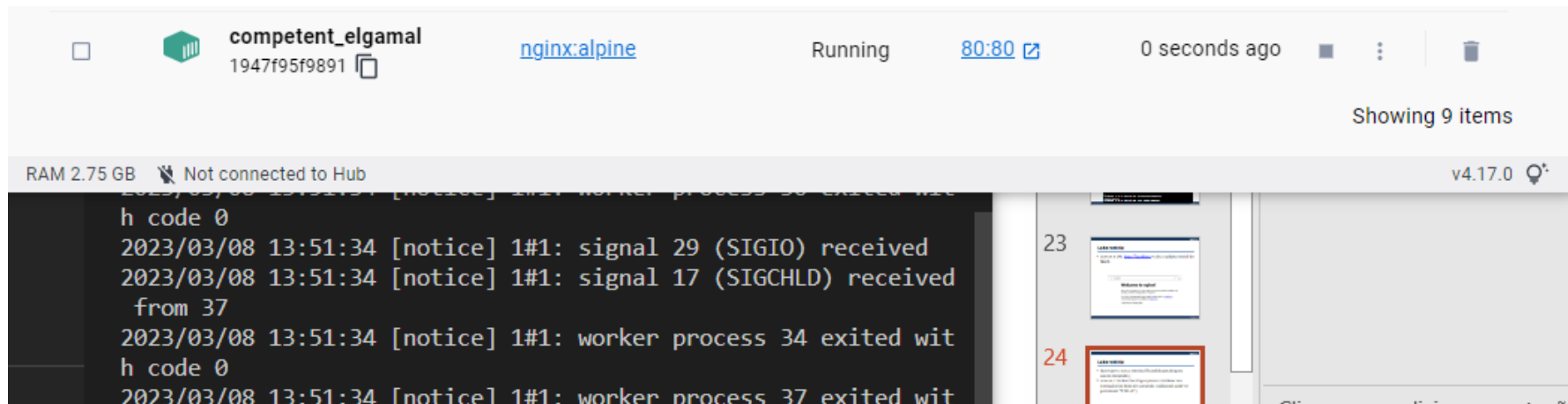
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Laboratório

- Note que o nosso terminal ficou bloqueado para novos comandos;
- Acesse o **Docker Desktop** e pare o contêiner em execução (na linha de comando tradicional pode-se pressionar “**CTRL+C**”):



Laboratório

- Agora, executaremos o contêiner em modo “**detached**”, ou seja, em background, incluindo o parâmetro “**-d**”:

```
docker run -p 80:80 -d nginx  
ed170c1b90300baf254c72eae16bd20ee610573105cd297b37b1bb048b08af33
```

- Note que neste caso o terminal fica livre para uso de outros comandos e a página do NGIX permanece no ar.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Laboratório

- Através do comando “docker ps” conseguimos ver detalhes do contêiner em execução:

```
docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
ed170c1b9030   nginx    "/docker-entrypoint...." 3 minutes ago  Up 3 minutes  0.0.0.0:80->80/tcp       relaxed_blackburn
```

- Agora iremos executar comandos Linux dentro do contêiner:

```
docker exec -it ed170c1b9030 bash
root@414e37520df8: /#
```

- Parâmetro **exec -it** indica que **nós** executaremos (modo interativo) o **bash** dentro do contêiner.

Laboratório

- Agora iremos alterar o arquivo “index.html” do NGIX, aquele que contém o “Welcome to nginx”;
- Este arquivo, por padrão, está no diretório “/usr/share/nginx/html”:

```
root@414e37520df8:/# cd /usr/share/nginx/html
root@414e37520df8:/usr/share/nginx/html# ls
50x.html  index.html
```

- Utilizaremos o “nano” para edição do arquivo:

```
apt-get update
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]Get:2 http://deb.debian.org/debian-security bullseye-
security
InRelease [48.4 kB]
.
.
.
apt-get install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Laboratório

- Vamos alterar o “Welcome to nginx” para “Welcome to Senac”;
- Em seguida pressione “CTRL+X” para sair do nano, confirme as alterações com “Y” e pressione “ENTER”:

nano index.html

```
GNU nano 7.2
<!DOCTYPE html>
<html>
<head>
<title>Welcome to Senac</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to Senac!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Laboratório

- Acesse novamente a URL <http://localhost/>, note que agora temos a página com nossas alterações:

Welcome to Senac!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Laboratório

- Saia do bash do contêiner (comando “**exit**”) e interromper a execução do contêiner (**docker stop** id-do-seu-contêiner) e, em seguida, inicie o contêiner novamente:

```
root@414e37520df8:/usr/share/nginx/html# exit
```

```
exit
```

```
docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
414e37520df8	nginx	"/docker-entrypoint...."	3 hours ago	Up 3 hours	0.0.0.0:80->80/tcp	wizardly_blackburn

```
docker stop 414e37520df8
```

```
docker ps
```

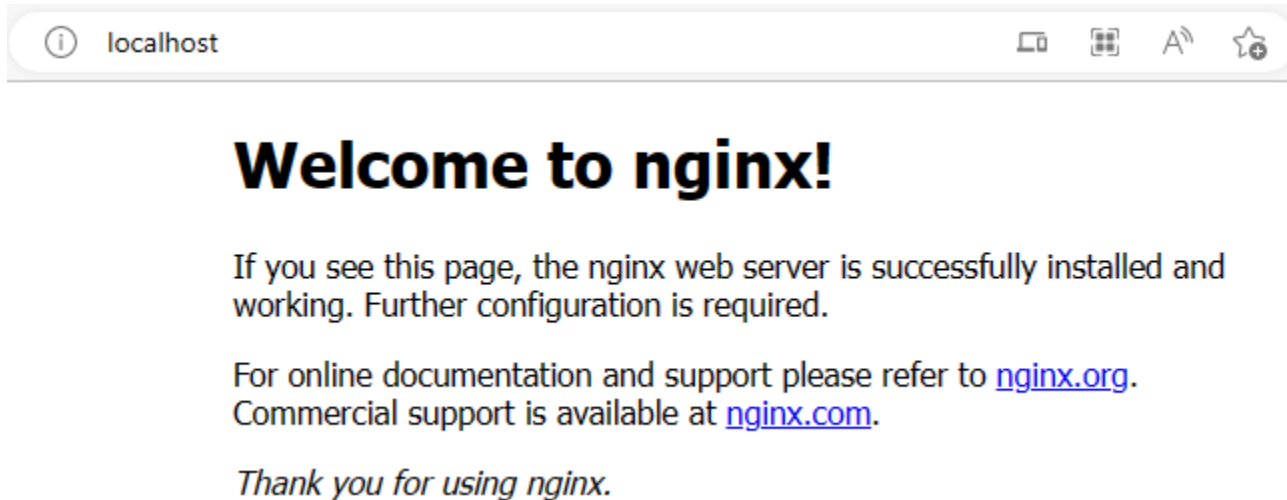
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
docker run -p 80:80 -d nginx
```

- Acesse novamente a página do NGIX: <http://localhost/>.

Laboratório

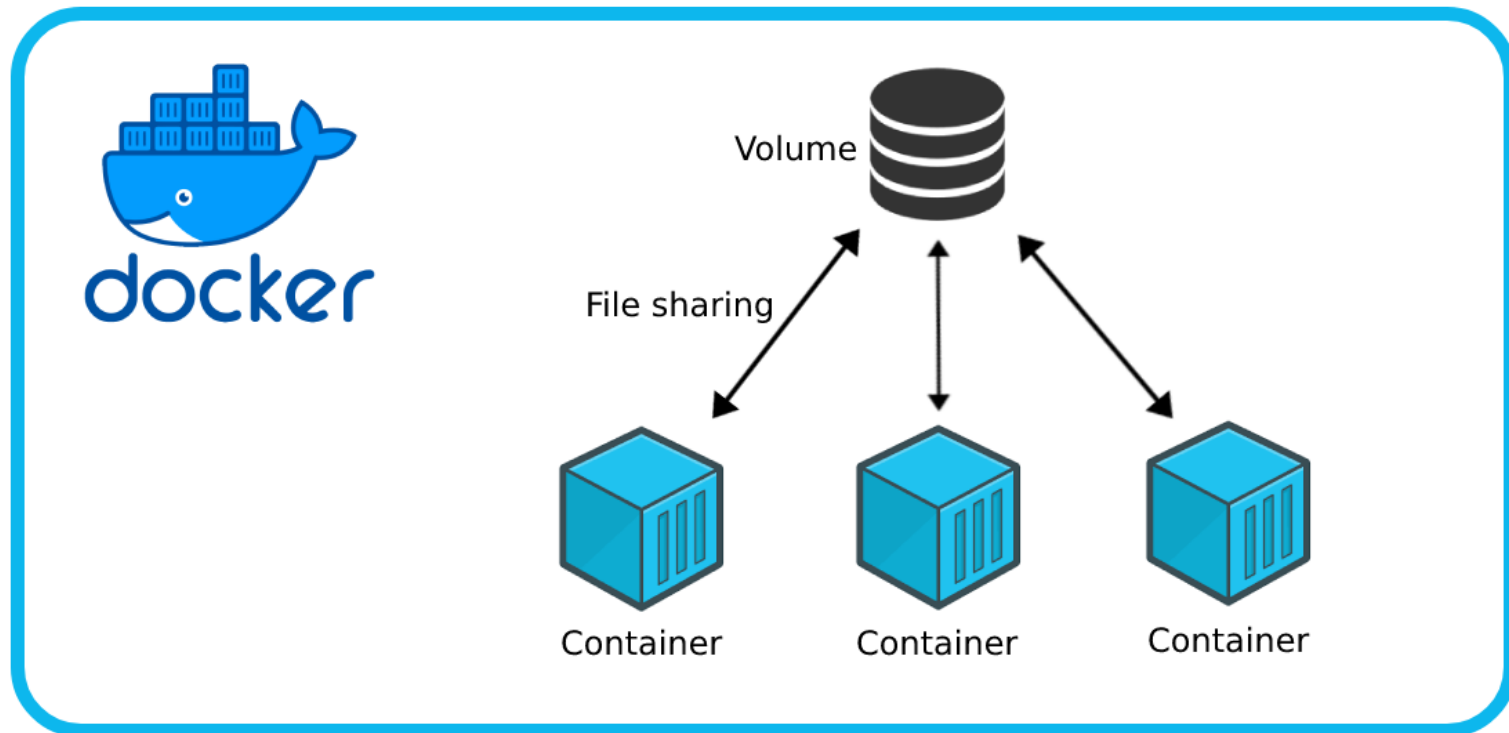
- Eis o resultado:



- Lembrem-se: um contêiner Docker executa uma imagem, que é **imutável**! Como tratar isto?

Laboratório

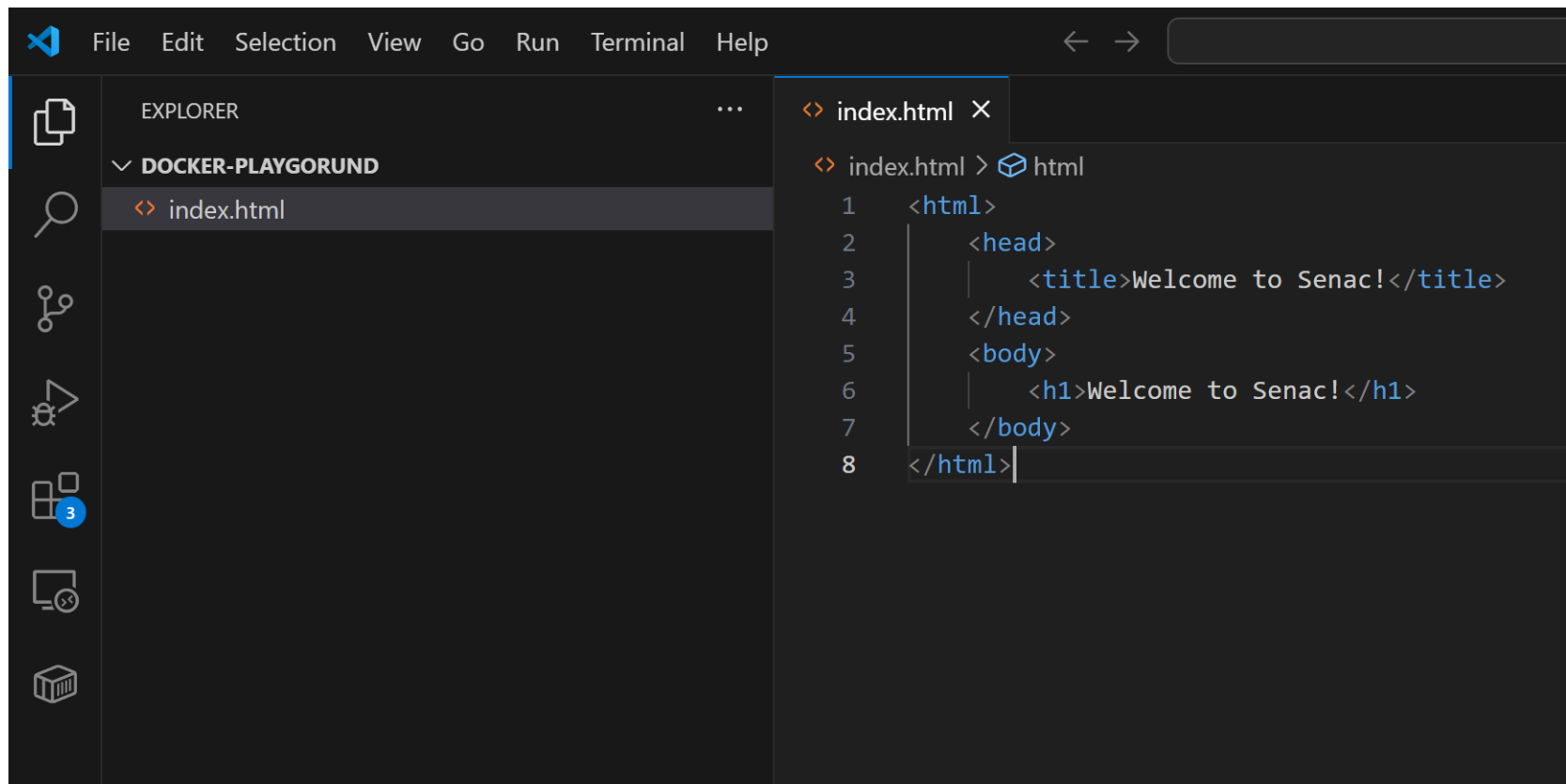
- Através de volumes!



Fonte: [Dominando Docker Volumes: Um Guia Abrangente | LinkedIn](#)

Laboratório

- Agora, iremos criar **nosso próprio index.html** (podendo ser criado a partir do index.html original do NGIX):



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a folder named 'DOCKER-PLAYGORUND' containing a file 'index.html'. The main editor area shows the content of 'index.html' with the following HTML code:

```
1 <html>
2   <head>
3     <title>Welcome to Senac!</title>
4   </head>
5   <body>
6     <h1>Welcome to Senac!</h1>
7   </body>
8 </html>
```

Laboratório

- Agora, vamos executar o “docker run” com um novo parâmetro:

```
docker run -p 80:80 -d -v ${PWD}:/usr/share/nginx/html nginx
```

- O parâmetro “-v” especifica um mapeamento de volume local para uma pasta existente no contêiner, sempre com a sintaxe -v DIRETORIO_LOCAL:DIRETORIO_REMOTO;
- A variável “\${PWD}” (Print Working Directory) contém o diretório atual de execução do terminal;
- Certifique-se de estar no diretório no qual o “index.html” foi criado.

Laboratório

- Acesse novamente o <http://localhost/>:



Laboratório

- Crie um **Dockerfile** capaz de reproduzir a página personalizada desenvolvida durante este laboratório:

```
# Imagem base do contêiner, no caso, a própria imagem do nginx
FROM nginx
# Contêiner irá conter seu index.html personalizado
COPY index.html /usr/share/nginx/html/index.html
```

- Ao finalizar, entregar no projeto **lab-04** do Gitlab:
 - **Commit** das alterações;
 - **Push** da *feature branch*;
 - Alterações na branch main, através de **Merge Request**;

Referências Bibliográficas

BOROSAN, Payamam. **Docker Handbook**. Disponível em <https://borosan.gitbook.io/docker-handbook/>. Acesso em 18 jan 2025;

DANIEL, João Francisco Lino. **MAC0475 2021 - Aula05.1 - Docker + Docker-Compose**. Disponível em <https://uclab.xyz/sistemas-complexos-2021-aula05-1>. Acesso em 18 jan 2025;

DOCKER. **Docker Docs**. Disponível em <https://docs.docker.com/>. Acesso em 18 jan 2025;

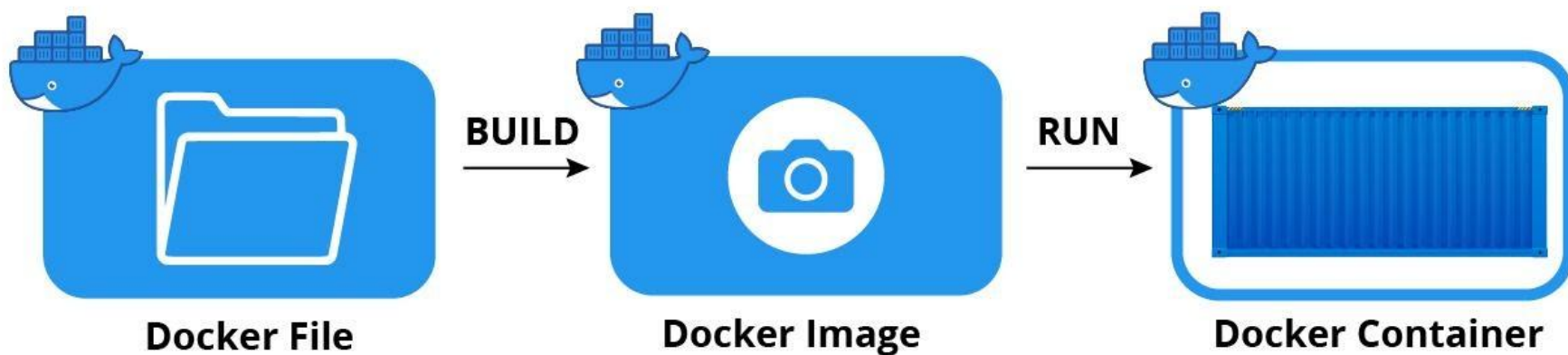
HERNANDEZ, Matt. **Using Docker in WSL 2**. Disponível em <https://code.visualstudio.com/blogs/2020/03/02/docker-in-wsl2>. Acesso em 18 jan 2025;

MICROSOFT. **Introdução aos contêineres remotos do Docker no WSL 2**. Disponível em <https://learn.microsoft.com/pt-br/windows/wsl/tutorials/wsl-containers>. Acesso em 18 jan 2025;

RED HAT. **O que é um container Linux?** Disponível em <https://www.redhat.com/pt-br/topics/containers/whats-a-linux-container>. Acesso em 18 jan 2025;

TSA CARGO. **Transporte de container: o que você precisa saber?** Disponível em <https://www.tsacargo.com.br/blog/transporte-de-container-o-que-voce-precisa-saber>. Acesso em 18 jan 2025;

Por hoje (agora sim!) é só!



Fonte: [Understanding and Building Docker Images \(jfrog.com\)](https://jfrog.com/understanding-and-building-docker-images/)

Prof. Esp. Guilherme Jorge Aragão da Cruz

✉ guilherme.cruz@alumni.usp.br

in linkedin.com/in/guijac

Licença

- Este conteúdo está licenciado sob a Licença Creative Commons Atribuição-NãoComercial-Compartilhalgual 4.0 Internacional (CC BY-NC-SA 4.0).
- Todos os direitos autorais sobre este conteúdo pertencem ao autor, e este material não pode ser usado comercialmente sem autorização expressa.
- Material elaborado no contexto da disciplina **Fundamentos de DevOps** do **Centro Universitário Senac**, para fins educacionais.
- As referências a marcas, produtos e tecnologias têm caráter exclusivamente educacional, não havendo vínculo institucional ou comercial com as organizações citadas.
- Para ver o texto completo da licença, acesse <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>.

Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.cruz@alumni.usp.br

 [linkedin.com/in/guijac](https://www.linkedin.com/in/guijac)