

## **AVALIAÇÃO PRÁTICA**

**Unidade Curricular: Tecnologias Para Desenvolvimento de Sistemas (2025)**

**Professor: Leonardo Medeiros**

**VALOR:** máximo de 100 PONTOS

**NOME:** Pedro Moraes Michalski

## **AP02 – Implementação de Sistemas em Containers RunC, LXC, Docker e Podman.**

### **Conteúdo: Tecnologias de Desenvolvimento e Operação de Sistemas**

- Sistema Operacional GNU com Linux
- Interface de Texto
- Containers: RunC, LXC, Docker e Podman
- Sistemas de Controle de Versão: Git

### **Objetivos**

- Implementar Sistemas em Containers
  - RunC
  - LXC
  - Docker
  - Podman
- Criar e compartilhar repositório do projeto Git

## **Descrição de procedimentos**

### **Parte I (RunC)**

O runc é um runtime de contêineres — basicamente, um programa que sabe criar e executar contêineres seguindo a especificação OCI (Open Container Initiative).

Para começar com os procedimentos, temos que instalar o runc, verificar a instalação.

**Comandos (Os comandos estão com sudo, por conta da permissão de administrador)**

**\$ sudo apt install runc**

**\$ runc --version**

```

pedro@Piterzin:~$ sudo apt install runc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  runc
0 upgraded, 1 newly installed, 0 to remove and 68 not upgraded.
Need to get 8043 kB of archives.
After this operation, 32.2 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.2.5-0ubuntu1~24.04.1 [8043 kB]
Fetched 8043 kB in 2s (3510 kB/s)
Selecting previously unselected package runc.
(Reading database ... 47496 files and directories currently installed.)
Preparing to unpack .../runc_1.2.5-0ubuntu1~24.04.1_amd64.deb ...
Unpacking runc (1.2.5-0ubuntu1~24.04.1) ...
Setting up runc (1.2.5-0ubuntu1~24.04.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
pedro@Piterzin:~$

```

**Imagem I – Instalação do runc**

```

pedro@Piterzin:~$ runc -version
runc version 1.2.5-0ubuntu1~24.04.1
spec: 1.2.0
go: go1.22.2
libseccomp: 2.5.5
pedro@Piterzin:~$

```

**Image II - Verificação de instalação**

Agora, devemos criar alguns diretórios, a fim de utilizarmos para trabalhar com os containers. O qual criaremos um diretório principal “runc-dire”, e dentro desse diretório criamos “rootfs” que servirá como pasta de root systemfile. Assim utilizando o comando “sudo runc spec”, para gerar o arquivo “config.json” que é baseado na especificação OCI, para enfim verificar sua criação.

#### **Comandos:**

**\$ sudo mkdir runc-dire**

**\$ cd runc-dire**

**\$ sudo mkdir rootfs**

**\$ sudo runc spec**

**\$ sudo cat config.json**

```

pedro@Piterzin:~$ sudo mkdir runc-dire
pedro@Piterzin:~$ cd runc-dire
pedro@Piterzin:~/runc-dire$ sudo mkdir rootfs
pedro@Piterzin:~/runc-dire$ sudo runc spec
pedro@Piterzin:~/runc-dire$ sudo cat config.json
{
  "ociVersion": "1.2.0",
  "process": {
    "terminal": true,
    "user": {
      "uid": 0,
      "gid": 0
    },
    "args": [
      "sh"
    ],
    "env": [
      "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
      "TERM=xterm"
    ],
    "cwd": "/",
  },
}

```

**Image III – Criação de repositórios e criação do config.json**

O proximo passo a se fazer, é baixar a imagem Alpine via Docker. Cria um container a partir da imagem alpine, sem executar. Exporta o sistema de arquivos desse container em formato .tar, extraímos para dentro de rootfs.

**Comandos:**

**\$ sudo sh -c `docker export \$(docker create alpine) | tar -C rootfs -xvf`**

```
pedro@Piterzin:~/runc-dire$ sudo sh -c 'docker export $(docker create alpine) | tar -C rootfs -xvf - # image'
.dockervenv
bin/
bin/arch
bin/ash
bin/base64
bin/bbconfig
bin/busybox
bin/cat
bin/chattr
bin/chgrp
bin/chmod
bin/chown
bin/cp
bin/date
bin/dd
bin/df
bin/dmesg
bin/dnsdomainname
bin/dumpkmap
bin/echo
bin/egrep
bin/false
bin/fatattr
bin/fdflush
bin/fgrep
bin/fsync
bin/getopt
bin/grep
bin/gunzip
bin/gzip
bin/hostname
bin/ionice
bin/iostat
bin/ipcalc
```

**Image IV – Imagem alpine via docker**

Nesse momentos rodamos o container e verificamos se está tudo certo, o runc procura no diretório atual um config.json.

**Comandos:**

**\$ sudo runc run cgl**

```
pedro@Piterzin:~/runc-dire$ sudo runc run cgl
/ # ls
bin      dev      etc      home    lib      media   mnt      opt      proc     root     run      sbin     srv      sys      tmp      usr      var
/ # pwd
/
/ # ls -la
total 64
drwxr-xr-x 19 root    root          4096 Aug 11 18:20 .
drwxr-xr-x 19 root    root          4096 Aug 11 18:20 ..
-rwxr-xr-x  1 root    root           0 Aug 11 18:20 .dockervenv
drwxr-xr-x  2 root    root          4096 Jul 15 10:42 bin
drwxr-xr-x  5 root    root          360 Aug 11 18:22 dev
drwxr-xr-x 17 root    root          4096 Aug 11 18:20 etc
drwxr-xr-x  2 root    root          4096 Jul 15 10:42 home
drwxr-xr-x  6 root    root          4096 Jul 15 10:42 lib
drwxr-xr-x  5 root    root          4096 Jul 15 10:42 media
drwxr-xr-x  2 root    root          4096 Jul 15 10:42 mnt
drwxr-xr-x  2 root    root          4096 Jul 15 10:42 opt
dr-xr-xr-x 270 root    root           0 Aug 11 18:22 proc
drwx----- 2 root    root          4096 Jul 15 10:42 root
drwxr-xr-x  3 root    root          4096 Jul 15 10:42 run
drwxr-xr-x  2 root    root          4096 Jul 15 10:42 sbin
drwxr-xr-x  2 root    root          4096 Jul 15 10:42 srv
dr-xr-xr-x 13 root    root           0 Aug 11 18:22 sys
drwxrwxrwt  2 root    root          4096 Jul 15 10:42 tmp
drwxr-xr-x  7 root    root          4096 Jul 15 10:42 usr
drwxr-xr-x 11 root    root          4096 Jul 15 10:42 var
/ # uname -a
Linux runc 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:30:46 UTC 2025 x86_64 Linux
/ #
```

**Imagem V – Run do container**

## Parte II (lxc/lxd)

Após, termos criado um diretório container, movemos o diretório “runc-dire” para dentro do container (Se possível, já comece com o runc-dire dentro desse diretório, esqueci de organizar antes). Dessa forma, começamos a instalação do lxc

```
pedro@Piterzin:~/container$ ls
runc-dire
pedro@Piterzin:~/container$ sudo apt install lxc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lxc is already the newest version (1:5.0.3-2ubuntu7.2).
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
pedro@Piterzin:~/container$
```

Imagem VI – Instalação do lxc

Nesse momento, criamos os containers usando lxc (Linux Containers) c1 e c2, porém falhou a c2 utilizando debian, então tive que utilizar uma instalação a parte. Também configurando o container c2.

#### Comandos:

\$ sudo lxc-create -n c1 -t busybox

\$ sudo lxc-create -n c2 -t debian

\$ sudo lxc-create -n container-debian -t download

```
pedro@Piterzin:~/container$ sudo lxc-create -n c1 -t busybox
pedro@Piterzin:~/container$ sudo lxc-create -n c2 -t debian
Error: amd64/ not available on mirror http://deb.debian.org/debian
lxc-create: c2: ../src/lxc/lxccontainer.c: create_run_template: 1628 Failed to create container from template
lxc-create: c2: ../src/lxc/tools/lxc_create.c: main: 317 Failed to create container c2
pedro@Piterzin:~/container$ sudo lxc-create -n container-debian -t download
lxc-create: container-debian: ../src/lxc/utills.c: get_template_path: 900 No such file or directory - bad template: dowl
oad
lxc-create: container-debian: ../src/lxc/lxccontainer.c: __lxcapi_create: 1799 Template "download" not found
lxc-create: container-debian: ../src/lxc/tools/lxc_create.c: main: 317 Failed to create container container-debian
pedro@Piterzin:~/container$ sudo lxc-create -n container-debian -t download
Downloading the image index
---
DIST      RELEASE ARCH      VARIANT BUILD
---
almalinux      8          amd64   default 20250809_23:08
almalinux      8          arm64   default 20250809_23:08
almalinux      9          amd64   default 20250809_23:08
almalinux      9          arm64   default 20250809_23:16
alpine 3.19      amd64   default 20250810_13:00
alpine 3.19      arm64   default 20250810_13:00
alpine 3.19      armhf   default 20250810_13:00
alpine 3.20      amd64   default 20250810_13:00
alpine 3.20      arm64   default 20250810_13:01
alpine 3.20      armhf   default 20250810_13:08
alpine 3.20      riscv64 default 20250810_13:15
alpine 3.21      amd64   default 20250810_13:00
alpine 3.21      arm64   default 20250810_13:00
alpine 3.21      armhf   default 20250810_13:00
alpine 3.21      riscv64 default 20250810_13:03
```

Imagem VII – Criação dos containers lxc

```

Distribution:
alpine
Release:
3.19
Architecture:
amd64

Downloading the image index
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

---
You just created an Alpinelinux 3.19 x86_64 (20250810_13:00) container.
pedro@Piterzin:~/containers$ _

```

Imagem VIII – Configuração do container.

A seguir, iniciamos os container c1 e c2, e conectamos o terminal a um processo dentro do container, basicamente entrando no container.

#### Comandos:

```
$ sudo lxc-start -n c1
```

```
$ sudo lxc-attach -n c1
```

```
$ sudo lxc-start -n c2
```

```
$ sudo lxc-attach -n c2
```

```

pedro@Piterzin:~/containers$ sudo lxc-start -n c1
pedro@Piterzin:~/containers$ sudo lxc-attach -n c1

BusyBox v1.36.1 (Ubuntu 1:1.36.1-6ubuntu3.1) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ # ^C

/ # ls
bin      etc      lib      mnt      root     selinux  tmp      var
dev      home    lib64    proc     sbin     sys      usr

/ # exit
pedro@Piterzin:~/containers$

```

Imagem IX – Run e attach do container c1

```

pedro@Piterzin:~/containers$ sudo lxc-start -n c2
pedro@Piterzin:~/containers$ sudo lxc-attach -n c2
/ # ls
bin      etc      lib      mnt      proc     run      srv      tmp      var
dev      home    media    opt      root     sbin     sys      usr

/ # pwd
/

/ # uname -a
Linux c2 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:30:46 UTC 2025 x86_64 Linux
/ # _

```

Imagem X – Run e attach do container c2

Agora, realizamos a instalação do libvirt-daemon-systems, para gerenciar máquinas virtuais e contêineres LXC pelo virt-manager (gráfico) ou virsh (CLI). Posteriormente finalizando os containers.

**Comandos:**

```
$ sudo apt install libvirt-daemon-system virt manager virtinst libvirt-clients libvirt-daemon-driver-lxc
```

```
$ sudo lxc-stop -n c1
```

```
$ sudo lxc-stop -n c2
```

```
pedro@Piterzin:~/container$ sudo apt install libvirt-daemon-system virt-manager virtinst libvirt-clients libvirt-daemon-driver-lxc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  acl alsa-topology-conf alsa-ucm-conf cpu-checker dmeventd dmidecode gir1.2-atk-1.0
  gir1.2-ayatanaappindicator3-0.1 gir1.2-freedesktop gir1.2-gdkpixbuf-2.0 gir1.2-gstreamer-1.0 gir1.2-gtk-3.0
  gir1.2-gtk-vnc-2.0 gir1.2-gtksource-4 gir1.2-harfbuzz-0.0 gir1.2-libosinfo-1.0 gir1.2-libvirt-glib-1.0
  gir1.2-pango-1.0 gir1.2-spiceclientglib-2.0 gir1.2-spiceclientgtk-3.0 gir1.2-vte-2.91 glib-networking
  glib-networking-common glib-networking-services gstreamer1.0-plugins-base gstreamer1.0-plugins-good
  gstreamer1.0-x i965-va-driver intel-media-va-driver ipxe-qemu ipxe-qemu-256k-compat-efi-roms libaa1
  libasound2-data libasound2t64 libasyncns0 libavc1394-0 libayatana-appindicator3-1 libayatana-ido3-0.4-0
  libayatana-indicator3-7 libbrlapi0.8 libburn4t64 libcacac0 libcacard0 libcdparanoia0 libdbusmenu-glib4
  libdbusmenu-gtk3-4 libdecor-0-0 libdecor-0-plugin-1-gtk libdevmapper-event1.02.1 libdv4t64 libegl-mesa0
  libfdt1 libflac12t64 libgbm1 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common
  libgl1-mesa-dri libglx-mesa0 libgstreamer-plugins-base1.0-0 libgstreamer-plugins-good1.0-0 libgtk-vnc-2.0-0
  libgtksourceview-4-0 libgtksourceview-4-common libgudev-1.0-0 libgvnc-1.0-0 libharfbuzz-gobject0
  libiec61883-0 libigdgmm12 libisoburn1t64 libisofs6t64 libjack-jackd2-0 libjson-glib-1.0-0
  libjson-glib-1.0-common liblvm2cmd2.03 libmp3lame0 libmpeg123-0t64 libncurses6 libnspr4 libnss-mymachines
```

**Imagem XI – Instalação do libvirt-daemon-systems.**

```
pedro@Piterzin:~/container$ sudo lxc-stop -n c1
pedro@Piterzin:~/container$ sudo lxc-stop -n c2
pedro@Piterzin:~/container$ _
```

**Imagem XII – Stop dos containers c1 e c2.**

Para continuar com os processos, verifique se o apt está atualizado, realizamos a instalação do snapd, e adicionamos o usuário atual a um grupo lxd, o que permitirá executar os comandos lxd.

**Comandos:**

**\$ sudo apt update**

**\$ sudo apt install snapd -y**

**\$ sudo usermod -aG lxd \$USER**

**\$ newgrp lxd**

```
pedro@Piterzin:~/container$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 126 kB in 1s (128 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
48 packages can be upgraded. Run 'apt list --upgradable' to see them.
pedro@Piterzin:~/container$ sudo apt install snapd -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  zenity | kdialog
The following packages will be upgraded:
  snapd
1 upgraded, 0 newly installed, 0 to remove and 47 not upgraded.
Need to get 22.7 MB of archives.
```

**Imagem XIII – Update do apt e instalação do snapd.**

```
pedro@Piterzin:~/container$ sudo usermod -aG lxd $USER
pedro@Piterzin:~/container$ newgrp lxd
```

**Imagem XIV – Configuração do user e criação do grupo.**

A seguir, criamos um diretório para o lxd, entramos no diretório, e iniciamos o lxd.

**Comandos:**

**\$ sudo mkdir lxd-dire**

**\$ cd lxd-dire**

**\$ sudo lxd init**

```
pedro@Piterzin:~/container$ sudo mkdir lxd-dire
pedro@Piterzin:~/container$ cd lxd-dire
pedro@Piterzin:~/container/lxd-dire$ sudo lxd init
Would you like to use LXD clustering? (yes/no) [default=no]: yes
What IP address or DNS name should be used to reach this server? [default=172.21.40.34]:
Are you joining an existing cluster? (yes/no) [default=no]:
What member name should be used to identify this server in the cluster? [default=Piterzin]:
Do you want to configure a new local storage pool? (yes/no) [default=yes]:
Name of the storage backend to use (dir, lvm, btrfs) [default=btrfs]:
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]:
Size in GiB of the new loop device (1GiB minimum) [default=30GiB]:
Do you want to configure a new remote storage pool? (yes/no) [default=no]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to configure LXD to use an existing bridge or host interface? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:
pedro@Piterzin:~/container/lxd-dire$
```

**Imagem XV – Criação do diretório lxd-dire e init do lxd.**

**PASSO IMPORTANTE!!**



A seguir, se ao dar launch em seu container, obter falha, utilize os comando lxc list, se ao listar o container e observe se existe alguma rede conectada ao IPV4 ou IPV6, caso não esteja, rode os comandos a seguir para criar uma rede para o container.

**Comandos:**

```
$ sudo lxc network create mynet ipv4.address=10.50.50.1/24 ipv4.nat=true  
ipv4.dhcp.ranges=10.50.50.2-10.50.50.100 ipv4.address=none
```

```
$ sudo lxc network attach mynet c1 eth0
```

```
$ sudo lxc restart c1
```

```
pedro@Piterzin:~/container/lxd-dire$ sudo lxc network create mynet ipv4.address=10.50.50.1/24 ipv4.nat=true ipv4.dhcp.ranges=10.50.50.2-10.50.50.100 ipv4.address=none  
Network mynet created  
pedro@Piterzin:~/container/lxd-dire$ sudo lxc network attach mynet c1 eth0  
pedro@Piterzin:~/container/lxd-dire$ sudo lxc restart c1  
pedro@Piterzin:~/container/lxd-dire$
```

**Imagem XVI – Criação da rede e conexão com o container.**

Assim, rodamos o launch do container c1, verificamos se a rede foi linkada adequadamente, utilizamos o start do container c1 e executamos o c1.

**Comandos:**

```
$ lxc launch images:debian/12 c1
```

```
$ lxc list
```

```
$ lxc start c1
```

```
$ lxc exec c1 -- bash
```

```
pedro@Piterzin:~/container/lxd-dire$ lxc launch images:debian/12 c1  
Launching c1  
Error: Failed instance creation: Failed creating instance record: Instance "c1" already exists  
pedro@Piterzin:~/container/lxd-dire$ lxc list  
+-----+-----+-----+-----+-----+-----+-----+  
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS | LOCATION |  
+-----+-----+-----+-----+-----+-----+-----+  
| c1 | RUNNING | | fd42:627c:191f:31ee:216:3eff:fe82:7a62 (eth0) | CONTAINER | 0 | Piterzin |  
+-----+-----+-----+-----+-----+-----+-----+  
| cont1 | RUNNING | | | CONTAINER | 0 | Piterzin |  
+-----+-----+-----+-----+-----+-----+-----+  
pedro@Piterzin:~/container/lxd-dire$ lxc start c1  
Error: The instance is already running  
pedro@Piterzin:~/container/lxd-dire$ lxc exec c1 --bash  
Error: unknown flag: --bash  
pedro@Piterzin:~/container/lxd-dire$ lxc exec c1 -- bash  
root@c1:~# ls  
root@c1:~# mkdir testando  
root@c1:~# ls  
testando  
root@c1:~# cd testando  
root@c1:~/testando# touch principal.py  
root@c1:~/testando# ls -a  
. . . principal.py  
root@c1:~/testando# ls -la  
total 0  
drwxr-xr-x 1 root root 24 Aug 11 19:14 .  
drwx----- 1 root root 54 Aug 11 19:13 ..  
-rw-r--r-- 1 root root 0 Aug 11 19:14 principal.py  
root@c1:~/testando# exit  
exit  
pedro@Piterzin:~/container/lxd-dire$ _
```

**Imagem XVII – Criação da rede e conexão com o container.**

## Parte III (Docker)

Para começar a parte de docker, devemos criar um novo diretório na nossa pasta container, criando alguns arquivos. OBS: Nos prints está errado, utilize "cd docker-dire" antes de criar os outros arquivos.



**Comandos:**

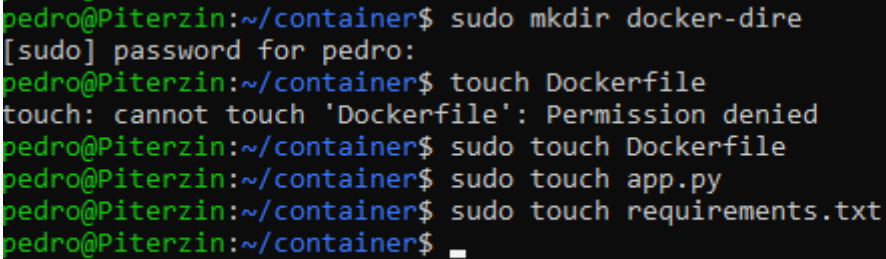
```
$ sudo mkdir docker-dire
```

```
$ cd docker-dire
```

```
$ sudo touch Dockerfil
```

```
$ sudo touch app.py
```

```
$ sudo touch requirements.txt
```



```
pedro@Piterzin:~/container$ sudo mkdir docker-dire
[sudo] password for pedro:
pedro@Piterzin:~/container$ touch Dockerfile
touch: cannot touch 'Dockerfile': Permission denied
pedro@Piterzin:~/container$ sudo touch Dockerfile
pedro@Piterzin:~/container$ sudo touch app.py
pedro@Piterzin:~/container$ sudo touch requirements.txt
pedro@Piterzin:~/container$
```

**Imagem XVIII – Criação do repositório e criação dos arquivos.**

A seguir, será necessário conseguir algumas informações, por meio da documentação do docker disponível na página <https://www.geeksforgeeks.org/devops/dockerize-your-flask-app/>. A primeira configuração que deve conseguir, são as congurações do app.py. Utilize o nano e cole o conteúdo da imagem XIX no app.py.

**Comandos:**

```
$ sudo nano app.py
```

```
$ code . (Se quiser abrir pelo VsCode)
```

**Step 1: Create a Flask App**

Inside the project folder, create app.py:



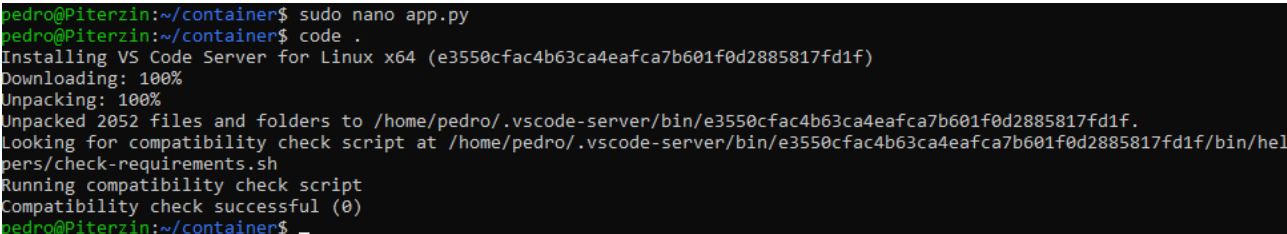
```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Welcome to Flask with Docker!"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000, debug=True)
```

**Imagem XIX – Conteúdo do app.py.**



```
pedro@Piterzin:~/container$ sudo nano app.py
pedro@Piterzin:~/container$ code .
Installing VS Code Server for Linux x64 (e3550cfac4b63ca4eafca7b601f0d2885817fd1f)
Downloading: 100%
Unpacking: 100%
Unpacked 2052 files and folders to /home/pedro/.vscode-server/bin/e3550cfac4b63ca4eafca7b601f0d2885817fd1f.
Looking for compatibility check script at /home/pedro/.vscode-server/bin/e3550cfac4b63ca4eafca7b601f0d2885817fd1f/bin/hel
pers/check-requirements.sh
Running compatibility check script
Compatibility check successful (0)
pedro@Piterzin:~/container$
```

**Imagem XX – Nano do app.py.**

Também na documentação do site disponibilizado, devemos conseguir algumas informações para o Dockerfile. Edite o arquivo do Dockerfile para colocar o seguinte conteúdo disponível na imagem XXI.

```
# Use an official lightweight Python image
FROM python:3.9-slim

# Set the working directory
WORKDIR /app

# Copy project files into the container
COPY . /app

# Install dependencies
RUN pip install -r requirements.txt

# Expose port 5000 for Flask
EXPOSE 5000

# Command to run the app
CMD ["python", "app.py"]
```

#### Imagem XXI – Conteúdo para o Dockerfile.

O próximo passo para prosseguir é iniciar um ambiente virtual, e pegar os requirements para o docker. Para com todas as informações preenchidas poder finalmente rodar o build do docker. Com a build executada com sucesso, rode o docker.

#### Comandos:

**\$ source venv/bin/activate**

**\$ pip freeze > requirements.txt**

**\$ pip install -r requirements.txt**

**\$ docker build -t flask-docker-app .**

**\$ docker run -p 5000:5000 flask-docker-app**

```
pedro@Piterzin:~/container/docker-dire$ source venv/bin/activate
(venv) pedro@Piterzin:~/container/docker-dire$ pip freeze > requirements.txt
(venv) pedro@Piterzin:~/container/docker-dire$ pip install -r requirements.txt
(venv) pedro@Piterzin:~/container/docker-dire$ docker build -t flask-docker-app .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 10.94MB
Step 1/6 : FROM python:3.9-slim
--> 563a905f7a66
Step 2/6 : WORKDIR /app
--> Using cache
--> 3e9f785fc835
Step 3/6 : COPY . /app
--> 3851ce3f4b62
Step 4/6 : RUN pip install -r requirements.txt
--> Running in cbd93f0d951a
```

#### Imagem XXII – Comandos para configuração do docker.

## Parte IV (Docker-rootless)

A seguir, será necessário utilizar alguns comandos, pegos na documentação do docker-rootless disponível na página <https://docs.docker.com/engine/security/rootless/>. Assim, instalando algumas dependências e excluindo algumas configurações, que será necessário para execução do docker-rootless.

### Comandos:

```
$ sudo apt-get install -y dbus-user-session
```

```
$ sudo apt-get install -y uidmap
```

```
$ sudo apt-get install -y systemd-container
```

```
$ sudo systemctl disable docker.service
```

```
$ sudo systemctl disable docker.socket
```

```
$ sudo systemctl stop docker.service
```

```
$ sudo systemctl stop docker.socket
```

```
$ sudo systemctl disable --now docker.service docker.socket
```

```
pedro@Piterzin:~/container/docker-dire$ sudo apt-get install -y dbus-user-session
[sudo] password for pedro:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
dbus-user-session is already the newest version (1.14.10-4ubuntu4.1).
dbus-user-session set to manually installed.
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
pedro@Piterzin:~/container/docker-dire$ sudo apt-get install -y uidmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
uidmap is already the newest version (1:4.13+dfsg1-4ubuntu3.2).
uidmap set to manually installed.
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
pedro@Piterzin:~/container/docker-dire$ sudo apt-get install -y systemd-container
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
systemd-container is already the newest version (255.4-1ubuntu8.10).
systemd-container set to manually installed.
The following packages were automatically installed and are no longer required:
  libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
pedro@Piterzin:~/container/docker-dire$
```

Imagem XXII – Comandos para configuração do docker.

```
pedro@Piterzin:~/container/docker-dire$ sudo systemctl disable docker.service
pedro@Piterzin:~/container/docker-dire$ sudo systemctl disable docker.socket
pedro@Piterzin:~/container/docker-dire$ sudo systemctl stop docker.service
pedro@Piterzin:~/container/docker-dire$ sudo systemctl stop docker.socket
```

Imagem XXIII – Comandos para configuração do docker.

```
pedro@Piterzin:~/container/docker-dire$ sudo systemctl disable --now docker.service docker.socket
pedro@Piterzin:~/container/docker-dire$
```

#### Imagem XXIV – Comandos para configuração do docker.

Próximo passo para o processo, é instalar os pacotes de scripts do docker rootless

```
pedro@Piterzin:~/container/docker-dire$ sudo apt install docker-ce-rootless-extras
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package docker-ce-rootless-extras
```

#### Imagem XXV – Comando para instalação de pacote.

```
pedro@Piterzin:~/container/docker-dire$ curl -fsSL https://get.docker.com/rootless | sh
# Installing stable version 28.3.3
# Executing docker rootless install script, commit: bedc5d6b3e782a5e50d3d2a870f5e1f1b5a38d5c
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 78.0M  100 78.0M    0     0  10.2M      0  0:00:07  0:00:07 --:--:-- 10.5M
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 16.6M  100 16.6M    0     0  9710k      0  0:00:01  0:00:01 --:--:-- 9708k
+ PATH=/home/pedro/bin:/home/pedro/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/wsl/lib:/mnt/c/Program Files/WindowsApps/MicrosoftCorporatio
nII.WindowsSubsystemForLinux_2.5.10.0_x64__8wekyb3d8bbwe:/mnt/c/Program Files (x86)/Common Files/Oracl
e/Java/java8path:/mnt/c/Program Files (x86)/Common Files/Oracle/Java/javapath:/mnt/c/Windows/system32:/
mnt/c/Windows:/mnt/c/Windows/System32/Wbem:/mnt/c/Windows/System32/WindowsPowerShell/v1.0:/mnt/c/Win
dows/System32/OpenSSH:/mnt/c/Program Files (x86)/NVIDIA Corporation/PhysX/Common:/mnt/c/Program Files
/NVIDIA Corporation/NVIDIA app/NvDLISR:/mnt/c/Program Files/NVIDIA Corporation/NVIDIA NvDLISR:/mnt/c/P
rogram Files/nodejs:/mnt/c/Users/Fabiano/AppData/Local/Microsoft/WindowsApps:/mnt/c/Users/Fabiano/App
Data/Local/Programs/Microsoft VS Code/bin:/mnt/c/msys64/mingw64/bin:/mnt/c/Users/Fabiano/AppData/Roami
ng/npm:/mnt/c/Program Files/Multipass/bin:/snap/bin /home/pedro/bin/dockerd-rootless-setuptool.sh inst
all
[INFO] Creating /home/pedro/.config/systemd/user/docker.service
[INFO] starting systemd service docker.service
```

#### Imagem XXVI – Comando para instalação de pacote.

Agora, adicionamos os diretórios onde o Docker Rootless instalou os binários ao PATH. E fazemos com que comandos como docker e dockerd-rootless.sh sejam encontrados pelo shell sem precisar digitar o caminho completo. Também, configuramos o Docker Rootless para rodar como um serviço systemd de usuário.

#### Comandos:

```
$ export PATH=$HOME/bin:$HOME/.local/bin:$PATH
```

```
$ dockerd-rootless-setuptool.sh install
```

```

pedro@Piterzin:~/container/docker-dire$ export PATH=$HOME/bin:$HOME/.local/bin:$PATH
pedro@Piterzin:~/container/docker-dire$ dockerd-rootless-setuptool.sh install
[WARNING] File already exists, skipping: /home/pedro/.config/systemd/user/docker.service
+ systemctl --user --no-pager --full status docker.service
• docker.service - Docker Application Container Engine (Rootless)
   Loaded: loaded (/home/pedro/.config/systemd/user/docker.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-08-11 23:53:26 -03; 56s ago
     Docs: https://docs.docker.com/go/rootless/
   Main PID: 3090 (rootlesskit)
    Tasks: 54
   Memory: 39.6M (peak: 51.6M)
      CPU: 697ms
   CGroup: /user.slice/user-1000.slice/user@1000.service/app.slice/docker.service
           └─3090 rootlesskit --state-dir=/run/user/1000/dockerd-rootless --net=vpnet --mtu=1500 --
slirp4netns-sandbox=auto --slirp4netns-seccomp=auto --disable-host-loopback --port-driver=builtin --co
py-up=/etc --copy-up=/run --propagation=rslave /home/pedro/bin/dockerd-rootless.sh
           └─3098 /proc/self/exe --state-dir=/run/user/1000/dockerd-rootless --net=vpnet --mtu=1500
--slirp4netns-sandbox=auto --slirp4netns-seccomp=auto --disable-host-loopback --port-driver=builtin -
--copy-up=/etc --copy-up=/run --propagation=rslave /home/pedro/bin/dockerd-rootless.sh
           └─3115 vpnetkit --ethernet /run/user/1000/dockerd-rootless/vpnetkit-ethernet.sock --mtu 1500 |
--host-ip 0.0.0.0
           └─3130 dockerd
           └─3152 containerd --config /run/user/1000/docker/containerd/containerd.toml

Aug 11 23:53:29 Piterzin systemd[663]: /home/pedro/.config/systemd/user/docker.service:7: Invalid envi
ronment assignment, ignoring: Files/NVIDIA

```

Imagem XXVII – Configuração do Docker rootless.

A seguir, criamos e executamos um container, o comando também permite remover o container ao sair "--rm".

```

pedro@Piterzin:~/container/docker-dire$ docker run -it --rm alpine sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
9824c27679d3: Pull complete
Digest: sha256:4bcff63911fcb4448bd4fdac207030997caf25e9bea4045fa6c8c44de311d1
Status: Downloaded newer image for alpine:latest
/ # ls
bin    etc    lib    mnt    proc   run    srv    tmp    var
dev    home   media  opt    root  /sbin  sys    usr
/ # uname-a
sh: uname-a: not found
/ # uname -a
Linux 90b069d279d3 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:30:46 UTC 202
5 x86_64 Linux
/ #

```

Imagem XXVIII – Comando para executar o container.

## Parte V (Podman)

Instala o Podman, que é um engine de contêineres compatível com Docker CLI, mas sem daemon e projetado para rodar rootless por padrão.

Permite criar e rodar contêineres OCI (mesmo formato que o Docker) sem precisar de dockerd. Também executamos o container utilizando o podman.

**Comandos:**

\$ sudo apt install podman

\$ podman run --rm docker.io/library/python:latest



```

pedro@Piterzin:~/container/docker-dire$ sudo apt install podman
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
 libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri libglapi-mesa libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 aardvark-dns buildah catatonit common containernetworking-plugins fuse-overlayfs
 golang-github-containers-common golang-github-containers-image libgpgme11t64 netavark passt
 slirp4netns
Suggested packages:
 containers-storage docker-compose
The following NEW packages will be installed:
 aardvark-dns buildah catatonit common containernetworking-plugins fuse-overlayfs
 golang-github-containers-common golang-github-containers-image libgpgme11t64 netavark passt podman
 slirp4netns
0 upgraded, 13 newly installed, 0 to remove and 30 not upgraded.
Need to get 32.1 MB of archives.
After this operation, 130 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 netavark amd64 1.4.0-4 [1666 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 aardvark-dns amd64 1.4.0-5 [881 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 containernetworking-plugins amd64

```

**Imagem XXIX – Instalação do podman.**

```

pedro@Piterzin:~/container/docker-dire$ podman run -it --rm docker.io/library/python:latest
WARN[0000] "/" is not a shared mount, this could cause issues or missing mounts with rootless containe
rs
Trying to pull docker.io/library/python:latest...
Getting image source signatures
Copying blob fb5f775313f0 done |
Copying blob ebed137c7c18 done |
Copying blob c2e76af9483f done |
Copying blob 873a4c802874 done |
Copying blob 37f838b71c6b done |
Copying blob 66d01326c435 done |
Copying blob 505bcd5de71d done |
Copying config 3e7f48ebe9 done |
Writing manifest to image destination
Python 3.13.6 (main, Aug 7 2025, 17:29:12) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> _

```

**Imagem XXX – Executar container com podman.**

## Parte VI (Git)

utilizamos o codeberg e git para versionar os códigos, para isso, criamos uma conta no codeberg, criamos um repositório sem nenhuma das opções marcadas, somente com o nome. Após a criação do repositório, utilizamos os seguintes comandos para enviar os arquivos para o repositório, fazendo pasta por pasta

**Comandos:**

**\$ touch README.md**

**\$ sudo touch README.md**

**\$ sudo git init**

**\$ sudo git switch -c main**

**\$ git config --global --add safe.directory (sua pasta, no meu caso é / home/pedro/container)**

**\$ git add README.md**

**\$ sudo git commit -m "first commit"**

**\$ sudo git push -u origin main**

Agora, faça para cada pasta.

**RUNC**

```
$ sudo git add runc-dire
$ sudo git commit -m "commit runc"
$ sudo git push -u origin main
```

## LXD

```
$ sudo git add lxd-dire
$ sudo git commit -m "commit lxd"
$ sudo git push -u origin main
```

## DOCKER

```
$ sudo git add docker-dire
$ sudo git commit -m "commit docker"
$ sudo git push -u origin main
```

```
pedro@Piterzin:~/container$ touch README.md
touch: cannot touch 'README.md': Permission denied
pedro@Piterzin:~/container$ sudo touch README.md
pedro@Piterzin:~/container$ git init
/home/pedro/container/.git: Permission denied
pedro@Piterzin:~/container$ sudo git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/pedro/container/.git/
pedro@Piterzin:~/container$ sudo git switch -c main
Switched to a new branch 'main'
```

Imagem XXXI – Primeiro commit.

```
pedro@Piterzin:~/container$ git config --global --add safe.directory /home/pedro/container
pedro@Piterzin:~/container$ git add README.md
fatal: Unable to create '/home/pedro/container/.git/index.lock': Permission denied
pedro@Piterzin:~/container$ sudo git add README.md
pedro@Piterzin:~/container$ _
```

Imagem XXXII – Configurações e add



```

pedro@Piterzin:~/container$ sudo git commit -m "first commit"
[main (root-commit) 4e943ec] first commit
Committer: root <root@Piterzin.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md

```

Imagem XXXIII – Commit.

```

pedro@Piterzin:~/container$ sudo git push -u origin main
Username for 'https://codeberg.org': Piterzin
Password for 'https://Piterzin@codeberg.org':
branch 'main' set up to track 'origin/main'.
Everything up-to-date
pedro@Piterzin:~/container$ _

```

Imagem XXXIV – Push –u origin main.

```

pedro@Piterzin:~/container$ sudo git add runc-dire
pedro@Piterzin:~/container$ git commit -m "commit runc"
fatal: Unable to create '/home/pedro/container/.git/index.lock': Permission denied
pedro@Piterzin:~/container$ sudo git commit -m "commit runc"
[main a554e6a] commit runc
Committer: root <root@Piterzin.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

426 files changed, 7274 insertions(+)
create mode 100644 runc-dire/config.json
create mode 100755 runc-dire/rootfs/.dockerenv
create mode 120000 runc-dire/rootfs/bin/arch
create mode 120000 runc-dire/rootfs/bin/ash
create mode 120000 runc-dire/rootfs/bin/base64
create mode 120000 runc-dire/rootfs/bin/bbconfig
create mode 100755 runc-dire/rootfs/bin/busybox
create mode 120000 runc-dire/rootfs/bin/cat

```

Imagem XXXV – Add e commit runc.

```
pedro@Piterzin:~/container$ sudo git push -u origin main
Username for 'https://codeberg.org': Piterzin
Password for 'https://Piterzin@codeberg.org':
Enumerating objects: 158, done.
Counting objects: 100% (158/158), done.
Delta compression using up to 12 threads
Compressing objects: 100% (126/126), done.
Writing objects: 100% (157/157), 3.49 MiB | 1.74 MiB/s, done.
Total 157 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://codeberg.org/Piterzin/container-to-tds.git
   4e943ec..a554e6a  main -> main
branch 'main' set up to track 'origin/main'.
pedro@Piterzin:~/container$
```

Imagem XXXVI – Push –u origin main.

```
pedro@Piterzin:~/container$ sudo git add lxd-dire
pedro@Piterzin:~/container$ sudo git commit -m "commit lxd"
[main 394549d] commit lxd
Committer: root <root@Piterzin.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

426 files changed, 7274 insertions(+)
create mode 100644 lxd-dire/config.json
create mode 100755 lxd-dire/rootfs/.dockerenv
create mode 120000 lxd-dire/rootfs/bin/arch
create mode 120000 lxd-dire/rootfs/bin/ash
create mode 120000 lxd-dire/rootfs/bin/base64
create mode 120000 lxd-dire/rootfs/bin/bbconfig
create mode 100755 lxd-dire/rootfs/bin/busybox
create mode 120000 lxd-dire/rootfs/bin/cat
create mode 120000 lxd-dire/rootfs/bin/chattr
create mode 120000 lxd-dire/rootfs/bin/chgrp
```

Imagem XXXVII – Add e commit lxd.

```
pedro@Piterzin:~/container$ sudo git push -u origin main
Username for 'https://codeberg.org': Piterzin
Password for 'https://Piterzin@codeberg.org':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 275 bytes | 275.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://codeberg.org/Piterzin/container-to-tds.git
   a554e6a..394549d  main -> main
branch 'main' set up to track 'origin/main'.
pedro@Piterzin:~/container$
```

Imagem XXXVIII – Push –u origin main.

```
pedro@Piterzin:~/container$ sudo git add docker-dire
pedro@Piterzin:~/container$ sudo git commit -m "commit docker"
[main a02d055] commit docker
Committer: root <root@Piterzin.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1190 files changed, 188475 insertions(+)
create mode 100644 docker-dire/Dockerfile
create mode 100755 docker-dire/app.py
create mode 100644 docker-dire/requirements
create mode 100644 docker-dire/requirements.txt
create mode 100644 docker-dire/venv/bin/Activate.ps1
create mode 100644 docker-dire/venv/bin/activate
create mode 100644 docker-dire/venv/bin/activate.csh
create mode 100644 docker-dire/venv/bin/activate.fish
create mode 100755 docker-dire/venv/bin/flask
create mode 100755 docker-dire/venv/bin/pip
```

Imagem XXXIX – Add e commit docker.

```

pedro@Piterzin:~/container$ sudo git push -u origin main
Username for 'https://codeberg.org': Piterzin
Password for 'https://Piterzin@codeberg.org':
Enumerating objects: 1321, done.
Counting objects: 100% (1321/1321), done.
Delta compression using up to 12 threads
Compressing objects: 100% (1284/1284), done.
Writing objects: 100% (1320/1320), 5.16 MiB | 2.20 MiB/s, done.
Total 1320 (delta 33), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (33/33), done.
To https://codeberg.org/Piterzin/container-to-tds.git
   394549d..a02d055  main -> main
branch 'main' set up to track 'origin/main'.
pedro@Piterzin:~/container$

```

**Imagem XXXX – Push –u origin main.**

Para finalizar, falta apenas configurar o usuário.

**Comandos:**

**\$ git config --global user.name "Piterzin"**

**\$ git config --global user.email "manopar24@gmail.com"**

**\$ sudo git push**

```

pedro@Piterzin:~/container$ git config --global user.name "Piterzin"
pedro@Piterzin:~/container$ git config --global user.email "Piterzin"
pedro@Piterzin:~/container$ git config --global user.email "manopar24@gmail.com"
pedro@Piterzin:~/container$ git push
Username for 'https://codeberg.org': Piterzin
Password for 'https://Piterzin@codeberg.org':
error: update_ref failed for ref 'refs/remotes/origin/main': cannot lock ref 'refs/remote
': Unable to create '/home/pedro/container/.git/refs/remotes/origin/main.lock': Permissio
Everything up-to-date
pedro@Piterzin:~/container$ sudo git push
Username for 'https://codeberg.org': Piterzin
Password for 'https://Piterzin@codeberg.org':
Everything up-to-date
pedro@Piterzin:~/container$

```

## Parte VII (Referências)

- GEEKSFORGEES. Dockerize your Flask App. Disponível em: <https://www.geeksforgeeks.org/devops/dockerize-your-flask-app/>. Acesso em: 12 ago. 2025.
- DOCKER. Rootless mode. Disponível em: <https://docs.docker.com/engine/security/rootless/>. Acesso em: 12 ago. 2025.