

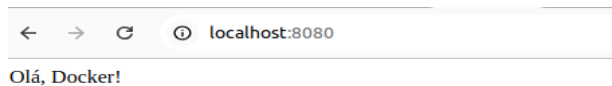


Universidade Senac Tecnologia
Pelotas - Rio Grande Do Sul

Computação em Nuvem - Docker

Disciplina de Computação em Nuvem no semestre de 2024/2

Por Débora Sieburger Carvalho



Primeiramente, após adentrar o sistema operacional "*linux ubuntu*", instalei o Google Chrome e comecei a instalar o **Docker** através da documentação descrita no site oficial. Código de instalação:

```
root@UbuntuDeb: /home/deb# for pkg in docker.io docker-doc docker-com
pose docker-compose-v2 podman-docker containerd runc; do sudo apt-get
remove $pkg; done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker.io' is not installed, so not removed
The following package was automatically installed and is no longer r
equired:
  libreoffice-ogltrans
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 284 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker-doc' is not installed, so not removed
The following package was automatically installed and is no longer r
equired:
  libreoffice-ogltrans
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 284 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker-compose' is not installed, so not removed
The following package was automatically installed and is no longer r
equired:
  libreoffice-ogltrans
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 284 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker-compose-v2' is not installed, so not removed
The following package was automatically installed and is no longer r
```

```
for pkg in docker.io docker-doc
docker-compose
docker-compose-v2
podman-docker containerd runc;
do sudo apt-get remove $pkg;
done
```

Dessa forma a gente tira arquivos conflitantes com a instalação do docker.



Instalando usando o apt repositório:

- `sudo apt-get update`
- `sudo apt-get install ca-certificates curl`
- `sudo install -m 0755 -d /etc/apt/keyrings`
- `sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc`
- `sudo chmod a+r /etc/apt/keyrings/docker.asc`

Instalando usando o apt repositório:

- `sudo apt-get update`
- `sudo apt-get install ca-certificates curl`
- `sudo install -m 0755 -d /etc/apt/keyrings`
- `sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc`
- `sudo chmod a+r /etc/apt/keyrings/docker.asc`

Adicionando ao repositório: • `echo \ "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \ $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \`

- `sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`
- `sudo apt-get update`

Para instalar a versão mais recente, executei • `sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin`

Verificando se a instalação do Docker Engine foi bem-sucedida executando a hello-world imagem: • `sudo service docker start`

- `sudo docker run hello-world`
- `sudo service docker status`
- `sudo service docker start`
- `sudo service docker stop`

Inicializando quando iniciar a máquina: ◦ `sudo systemctl enable docker.service`

◦ `sudo systemctl enable containerd.service`

Parar a inicialização: ◦ `sudo systemctl disable docker.service`



○ sudo systemctl disable containerd.service

● docker version

Verificando os docker ativos: ● docker ps

docker container ls -a

```
root@UbuntuDeb:/home/deb# sudo service docker status
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-09-03 10:23:39 -03; 3min ago
     TriggeredBy: ● docker.socket
        Docs: https://docs.docker.com
       Main PID: 10437 (dockerd)
          Tasks: 12
         Memory: 43.1M
            CPU: 5.865s
        CGroup: /system.slice/docker.service
               └─10437 /usr/bin/dockerd -H fd:// --containerd=/run/containerd.sock

set 03 10:23:31 UbuntuDeb dockerd[10437]: time="2024-09-03T10:23:31.123Z" level=info msg="Starting Docker Engine"
set 03 10:23:31 UbuntuDeb dockerd[10437]: time="2024-09-03T10:23:31.123Z" level=info msg="Docker Engine"
set 03 10:23:32 UbuntuDeb dockerd[10437]: time="2024-09-03T10:23:32.123Z" level=info msg="Docker Engine"
set 03 10:23:37 UbuntuDeb dockerd[10437]: time="2024-09-03T10:23:37.123Z" level=info msg="Docker Engine"
set 03 10:23:38 UbuntuDeb dockerd[10437]: time="2024-09-03T10:23:38.123Z" level=info msg="Docker Engine"
set 03 10:23:38 UbuntuDeb dockerd[10437]: time="2024-09-03T10:23:38.123Z" level=info msg="Docker Engine"
set 03 10:23:39 UbuntuDeb dockerd[10437]: time="2024-09-03T10:23:39.123Z" level=info msg="Docker Engine"
set 03 10:23:39 UbuntuDeb systemd[1]: Started Docker Application Container Engine.
set 03 10:26:15 UbuntuDeb dockerd[10437]: time="2024-09-03T10:26:15.123Z" level=info msg="Docker Engine"
set 03 10:26:35 UbuntuDeb dockerd[10437]: time="2024-09-03T10:26:35.123Z" level=info msg="Docker Engine"

lines 1-22/22 (END)
```

```
root@UbuntuDeb:/home/deb# docker version
Client: Docker Engine - Community
 Version: 27.2.0
  API version: 1.47
 Go version: go1.21.13
 Git commit: 3ab4256
  Built: Tue Aug 27 14:15:13 2024
 OS/Arch: linux/amd64
 Context: default

Server: Docker Engine - Community
 Engine:
  Version: 27.2.0
  API version: 1.47 (minimum version 1.24)
  Go version: go1.21.13
  Git commit: 3ab5c7d
  Built: Tue Aug 27 14:15:13 2024
 OS/Arch: linux/amd64
 Experimental: false
 containerd:
  Version: 1.7.21
  GitCommit: 472731909fa34bd7bc9c087e4c27943f9835f111
 runc:
  Version: 1.1.13
  GitCommit: v1.1.13-0-g58aa920
 docker-init:
  Version: 0.19.0
  GitCommit: de40ad0
root@UbuntuDeb:/home/deb# S
```

Criei um diretório e comecei a instalação do Flask.

Baixei o python pelo diretório root:

sudo apt install python3-venv

Criei um dockerfile e um requirements.txt

```
(venv) root@UbuntuDeb:/home/deb/Documents/dir_trabalho# pip show werkzeug
Name: Werkzeug
Version: 3.0.4
Summary: The comprehensive WSGI web application library.
Home-page:
Author:
Author-email:
License:
Location: /home/deb/Documents/dir_trabalho/venv/lib/python3.10/site-packages
Requires: MarkupSafe
Required-by: Flask
(venv) root@UbuntuDeb:/home/deb/Documents/dir_trabalho#
```

```
python3 -m venv venv
sudo apt install python3.10-venv
source venv/bin/activate
Instalar dependências no python:
pip3 install --no-cache-dir -r requirements.txt
pip install --upgrade flask
python app.py
pip freeze > requirements.txt
pip show werkzeug
deactivate
```

Após isso criei a imagem docker a partir do código:

docker build -t minha-imagem:v1 .



- Dockerfile

```
Dockerfile x
Dockerfile > ...
1 FROM ubuntu:22.04
2 RUN apt-get update && \
3 apt-get install python3.11 python3.11-dev python3-pip -y
4 WORKDIR /app
5 COPY . .
6 RUN pip3 install --no-cache-dir -r requirements.txt
7 EXPOSE 8080
8 ENV LOGOMARCA="iagem(link)"
9 ENV FOTO="iagem(link)"
10 ENV NOME="DEB"
11 ENV IDADE="DEB"
12 ENV EMAIL="DEB"
13 ENV PROFISSAO="DEB"
14 ENV SITE="DEB"
15 CMD [ "python3", "app.py" ]
```

Criei um arquivo requirements.txt e outro app.py

`python3 -m venv venv`

- `sudo apt install python3.10-venv source venv/bin/activate`
- `pip3 install --no-cache-dir -r requirements.txt`
- `pip install --upgrade flask`
- `python app.py pip freeze > requirements.txt`
- `pip show werkzeug deactivate`

- Image dir

- `docker run nome-da-imagem`

`docker build -t minha-imagem:v1 .` (o comando tem o ponto (.) de pasta local)
`docker images docker run -it -p 8080:8080 minha-imagem:v1`

`docker run -it -p 8080:8080 image_dir:v1`