



INSTITUTO FEDERAL

Ceará
Campus Fortaleza

Telemática

Administração de Serviços de Rede

Professor: Ricardo Duarte Taveira

Alunos: Luan Victor Santiago Sobrinho e

Denilson Barbosa de Oliveira

Atividade 08 | Proxy Reverso - Nginx

Passo 1: Instalação do Docker e Docker Compose

```
luanv3@luanv3-VirtualBox:~$ # Add Docker's official GPG key:
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

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
    $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
Atingido:1 http://archive.ubuntu.com/ubuntu noble InRelease
Atingido:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Atingido:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Atingido:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Lendo listas de pacotes... Pronto
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
ca-certificates já é a versão mais recente (20240203).
curl já é a versão mais recente (8.5.0-2ubuntu10.6).
0 pacotes atualizados, 0 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
Obter:1 https://download.docker.com/linux/ubuntu noble InRelease [48,8 kB]
Atingido:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Atingido:3 http://archive.ubuntu.com/ubuntu noble InRelease
Atingido:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Obter:5 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [18,9 kB]
Atingido:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Baixados 67,7 kB em 1s (49,5 kB/s)
Lendo listas de pacotes... Pronto
luanv3@luanv3-VirtualBox:~$
```

```
luanv3@luanv3-VirtualBox:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
Os pacotes adicionais seguintes serão instalados:
  docker-ce-rootless-extras libslirp0 pigz slirp4netns
Pacotes sugeridos:
  aufs-tools cgroupfs-mount | cgroup-lite
Os NOVOS pacotes a seguir serão instalados:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libslirp0 pigz slirp4netns
0 pacotes atualizados, 9 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
É preciso baixar 127 MB/127 MB de arquivos.
Depois desta operação, 458 MB adicionais de espaço em disco serão usados.
Você quer continuar? [S/n] S
Obter:1 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.25-1 [29,6 MB]
Obter:2 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.20.0-1-ubuntu.24.04-noble [33,2 MB]
Obter:3 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.5.1-1-ubuntu.24.04-noble [15,2 MB]
Obter:4 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.5.1-1-ubuntu.24.04-noble [26,1 MB]
Obter:5 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:27.5.1-1-ubuntu.24.04-noble [9.601 kB]
Obter:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.32.4-1-ubuntu.24.04-noble [12,8 MB]
Baixados 127 MB em 22s (5.863 kB/s)
A seleccionar pacote anteriormente não seleccionado pigz.
(Lendo banco de dados ... 154774 ficheiros e directórios atualmente instalados.)
A preparar para desempacotar .../0-pigz_2.8-1_amd64.deb ...
A descompactar pigz (2.8-1) ...
A seleccionar pacote anteriormente não seleccionado containerd.io.
A preparar para desempacotar .../1-containerd.io_1.7.25-1_amd64.deb ...
A descompactar containerd.io (1.7.25-1) ...
A seleccionar pacote anteriormente não seleccionado docker-buildx-plugin.
A preparar para desempacotar .../2-docker-buildx-plugin_0.20.0-1-ubuntu.24.04-noble_amd64.deb ...
A descompactar docker-buildx-plugin (0.20.0-1-ubuntu.24.04-noble) ...
A seleccionar pacote anteriormente não seleccionado docker-ce-cli.
A preparar para desempacotar .../3-docker-ce-cli_5:27.5.1-1-ubuntu.24.04-noble_amd64.deb ...
A descompactar docker-ce-cli (5:27.5.1-1-ubuntu.24.04-noble) ...
A seleccionar pacote anteriormente não seleccionado docker-ce.
A preparar para desempacotar .../4-docker-ce_5:27.5.1-1-ubuntu.24.04-noble_amd64.deb ...
A descompactar docker-ce (5:27.5.1-1-ubuntu.24.04-noble) ...
A seleccionar pacote anteriormente não seleccionado docker-ce-rootless-extras.
A preparar para desempacotar .../5-docker-ce-rootless-extras_5:27.5.1-1-ubuntu.24.04-noble_amd64.deb ...
A descompactar docker-ce-rootless-extras (5:27.5.1-1-ubuntu.24.04-noble) ...
```

```
luanv3@luanv3-VirtualBox:~$ docker --version
Docker version 27.5.1, build 9f9e405
luanv3@luanv3-VirtualBox:~$ docker compose version
Docker Compose version v2.33.0
luanv3@luanv3-VirtualBox:~$
```

Passo 2: Ativar o serviço do docker e configurar para inicializar com o sistema

```
luanv3@luanv3-VirtualBox:~$ docker --version
Docker version 27.5.1, build 9f9e405
luanv3@luanv3-VirtualBox:~$ docker compose version
Docker Compose version v2.33.0
luanv3@luanv3-VirtualBox:~$ sudo systemctl enable --now docker docker.socket containerd
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
luanv3@luanv3-VirtualBox:~$
```

Passo 3: Criar diretório do projeto

3. Acessar diretório

Reaproveitamento da atividade 06

3.1. Criar o diretório `flask_app` e o arquivo da aplicação web `app.py`.

3.2. Colar o seguinte código:

```
import time

import redis
from flask import Flask

app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)

def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)

@app.route('/')
def hello():
    count = get_hit_count()
    return f'Hello World! I have been seen {count} times.\n'
```

Esse código cria uma aplicação web simples utilizando o framework Flask e o banco de dados em memória Redis para contar o número de vezes que a página principal da aplicação foi acessada.

3.3. Crie outro arquivo chamado `requirements.txt` e liste as dependências de Python necessárias para o projeto:

- flask, para criar a aplicação web.
- redis, para interagir com o banco de dados Redis.

3.4. Criar um Dockerfile:

```
# syntax=docker/dockerfile:1
FROM python:3.10-alpine
WORKDIR /code
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
EXPOSE 5000
COPY . .
CMD ["flask", "run", "--debug"]
```

Esse Dockerfile é um script de construção para criar uma imagem Docker para rodar uma aplicação Flask com Python.

Passo 4: Configuração do NGINX

4. Criar a pasta nginx

4.1. Criar o arquivo nginx.conf para configuração do proxy reverso:

```
server {
    listen 80;

    location / {
        root /usr/share/nginx/html;
        index index.html;
    }

    location /teste/ {
        root /usr/share/nginx/html; # Caminho onde seu index está
        try_files /teste.html =404; # Seu index personalizado
    }

    location /api/ {
        proxy_pass http://meu_flask:5000/;
        rewrite ^/api(/.*)$ $1 break;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Este arquivo configura um servidor NGINX que serve arquivos estáticos e atua como proxy reverso.

- Requisições para / e /teste/ entregam arquivos locais.
- Requisições para /api/ são encaminhadas a um serviço Flask.
- Cabeçalhos são ajustados para manter informações do cliente.

Passo 5: Definir os serviços no arquivo Compose

5. Criar o arquivo `compose.yaml`

5.1. Configurar os serviços:

```
services:

  nginx:
    image: nginx
    container_name: meu_proxy
    ports:
      - "8080:80"
    depends_on:
      - web
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/conf.d/default.conf:ro
      - ./site_html:/usr/share/nginx/html:ro
    networks:
      - minha_rede

  web:
    build: ./flask_app
    container_name: meu_flask
    depends_on:
      - redis
    expose:
      - "5000"
    networks:
      - minha_rede

  redis:
    image: "redis:alpine"
    networks:
      - minha_rede

networks:
  minha_rede:
    driver: bridge
```

O docker compose define três serviços:

- O NGINX (`meu_proxy`) atua como proxy reverso e servidor de arquivos estáticos, redirecionando requisições para os outros serviços.
- O serviço web (Flask) (`meu_flask`) será construído a partir do `Dockerfile` localizado no diretório `flask_app`.
- O Redis (`redis`) usa a imagem `redis:alpine`, que será baixada se ainda não estiver no cache.

Os containers Flask e Redis rodam em paralelo, permitindo que a aplicação Flask interaja com o Redis para armazenar e recuperar dados. Todos os serviços estão na mesma rede `minha_rede`, garantindo comunicação interna eficiente.

Passo 6: Criar e executar os serviços com o Compose

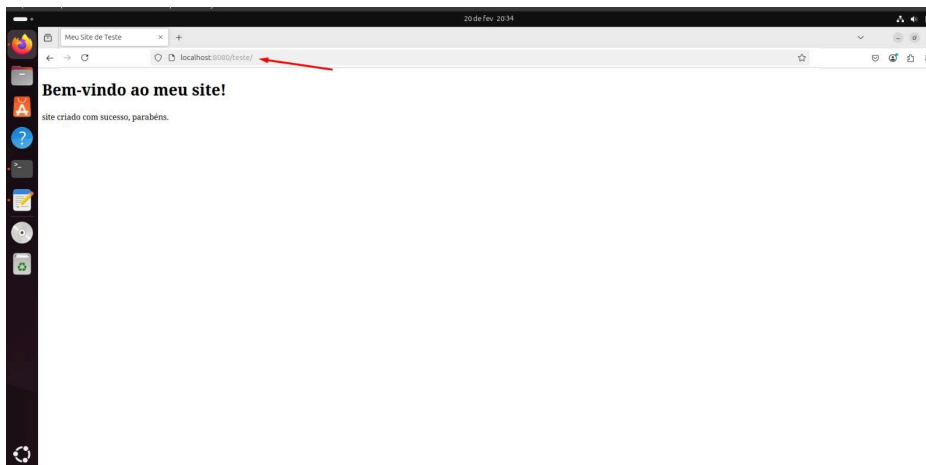
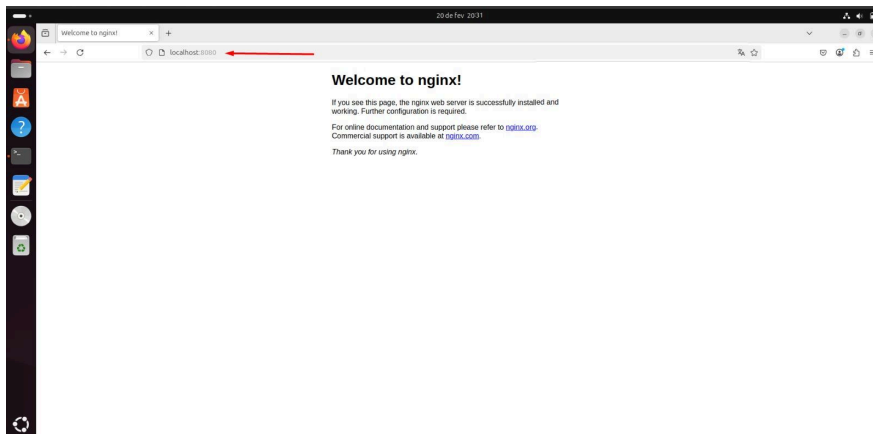
6. Iniciar os serviços com o comando `docker compose up`:

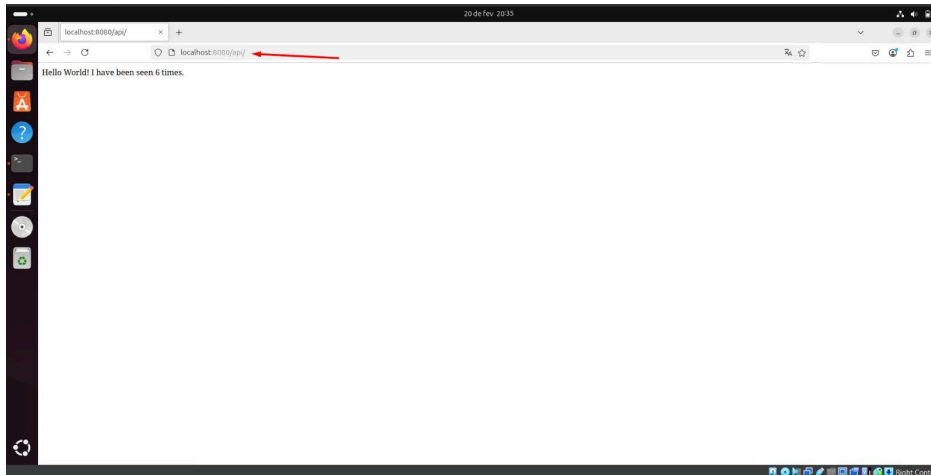
```
root@luanv3-VirtualBox:/home/luanv3/projeto_proxy#
root@luanv3-VirtualBox:/home/luanv3/projeto_proxy#
root@luanv3-VirtualBox:/home/luanv3/projeto_proxy# docker compose up -d --build
[+] Building 3.3s (14/14) FINISHED                                docker:default
=> [web internal] load build definition from Dockerfile           0.1s
=> => transferring dockerfile: 332B                               0.0s
=> [web] resolve image config for docker-image://docker.io/docker/docker 1.5s
=> CACHED [web] docker-image://docker.io/docker/dockerfile:1@sha256:93bf 0.0s
=> [web internal] load metadata for docker.io/library/python:3.10-alpine 0.6s
=> [web internal] load .dockerignore                               0.0s
=> => transferring context: 28B                                     0.0s
=> [web 1/6] FROM docker.io/library/python:3.10-alpine@sha256:4c4097e466 0.0s
=> [web internal] load build context                               0.0s
=> => transferring context: 93B                                     0.0s
=> CACHED [web 2/6] WORKDIR /code                                 0.0s
=> CACHED [web 3/6] RUN apk add --no-cache gcc musl-dev linux-headers 0.0s
=> CACHED [web 4/6] COPY requirements.txt requirements.txt        0.0s
=> CACHED [web 5/6] RUN pip install -r requirements.txt           0.0s
=> CACHED [web 6/6] COPY . .                                      0.0s
=> [web] exporting to image                                       0.0s
=> => exporting layers                                             0.0s
=> => writing image sha256:20be8cbb9a9667158066f6965af584aee1e95c4be0a83 0.0s
=> => naming to docker.io/library/projeto_proxy-web              0.0s
=> [web] resolving provenance for metadata file                   0.1s
[+] Running 3/4
✔ web                               Built
0.0s Container projeto_proxy-redis-1 S... 0.9s
✔ Container projeto_proxy-redis-1 Starting
[+] Running 4/4U_proxy              Created 0.0s
✔ web                               Built
✔ Container projeto_proxy-redis-1 Started
✔ Container meu_flask               Started
✔ Container meu_proxy               Started
root@luanv3-VirtualBox:/home/luanv3/projeto_proxy#
```

6.1. Verificar os containers:

```
root@luanv3-VirtualBox:/home/luanv3/projeto_proxy# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
7b5d25af2f87   nginx         "/docker-entrypoint..." 28 hours ago   Up About a minute   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   meu_proxy
871cf1acd8aa   projeto_proxy-web "flask run --debug"       28 hours ago   Up About a minute   5000/tcp                               meu_flask
f97d6d7ddf1c   redis:alpine  "docker-entrypoint.s..." 28 hours ago   Up About a minute   6379/tcp                               projeto_proxy-redis-1
root@luanv3-VirtualBox:/home/luanv3/projeto_proxy#
```

6.2. Testar os serviços com o localhost:





6.3. Testar o serviço acessando <http://192.168.18.53:8000>(ip da minha máquina onde o serviço está rodando)

