

# Ambiente Docker

## Primeiro Passo - Instalações

`sudo apt update`

`sudo apt install -y docker.io docker-compose`

`sudo systemctl start docker`

`sudo systemctl enable docker`

```
root@GabrielPagliari:~# sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1528 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [262 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1941 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [331 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [860 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [167 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.2 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7588 B]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:21 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1736 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [319 kB]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2000 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [340 kB]
Get:25 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1088 kB]
Get:26 http://archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [251 kB]
```

```
root@GabrielPagliari:~# sudo apt update
Setting up pigz (2.6-1) ...
Setting up containerd (1.7.12-0ubuntu2~22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up python3-lib2to3 (3.10.8-1~22.04) ...
Setting up python3-websocket (1.2.3-1) ...
Setting up python3-dockerpty (0.4.1-2) ...
Setting up python3-distutils (3.10.8-1~22.04) ...
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up python3-setuptools (59.6.0-1.2ubuntu0.22.04.1) ...
Setting up docker.io (24.0.7-0ubuntu2~22.04.1) ...
Adding group 'docker' (GID 117) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up python3-jsonschema (3.2.0-0ubuntu2) ...
Setting up python3-requests (2.25.1+dfsg-2ubuntu0.1) ...
Setting up python3-docker (5.0.3-1) ...
Setting up docker-compose (1.29.2-1) ...
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
root@GabrielPagliari:~# sudo systemctl start docker
```

## Passo 2 - Iniciar o Swarm

`sudo docker swarm init`

```
root@GabrielPagliari: ~  
Error response from daemon: could not choose an IP address to advertise since this system has multiple address  
es on different interfaces (10.255.255.254 on lo and 172.27.9.231 on eth0) - specify one with --advertise-addr  
root@GabrielPagliari:~# sudo docker swarm --172.27.9.231  
unknown flag: --172.27.9.231  
See 'docker swarm --help'.  
  
Usage:  docker swarm COMMAND  
  
Manage Swarm  
  
Commands:  
  init      Initialize a swarm  
  join      Join a swarm as a node and/or manager  
  
Run 'docker swarm COMMAND --help' for more information on a command.  
  
root@GabrielPagliari:~# sudo docker swarm init --advertise-addr 172.27.9.231  
Swarm initialized: current node (j6306actocnx68bgeqmh62rmv) is now a manager.  
  
To add a worker to this swarm, run the following command:  
  
    docker swarm join --token SWMTKN-1-2pop2uf3bjs1nkt6443y862jq90qy6qld4hugimj8w2jpkfsna-0vjt5g1pek2gbtkcpbdk  
olw0b 172.27.9.231:2377  
  
To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.  
root@GabrielPagliari:~#
```

## Passo 3- Criar o projeto e ir na pasta

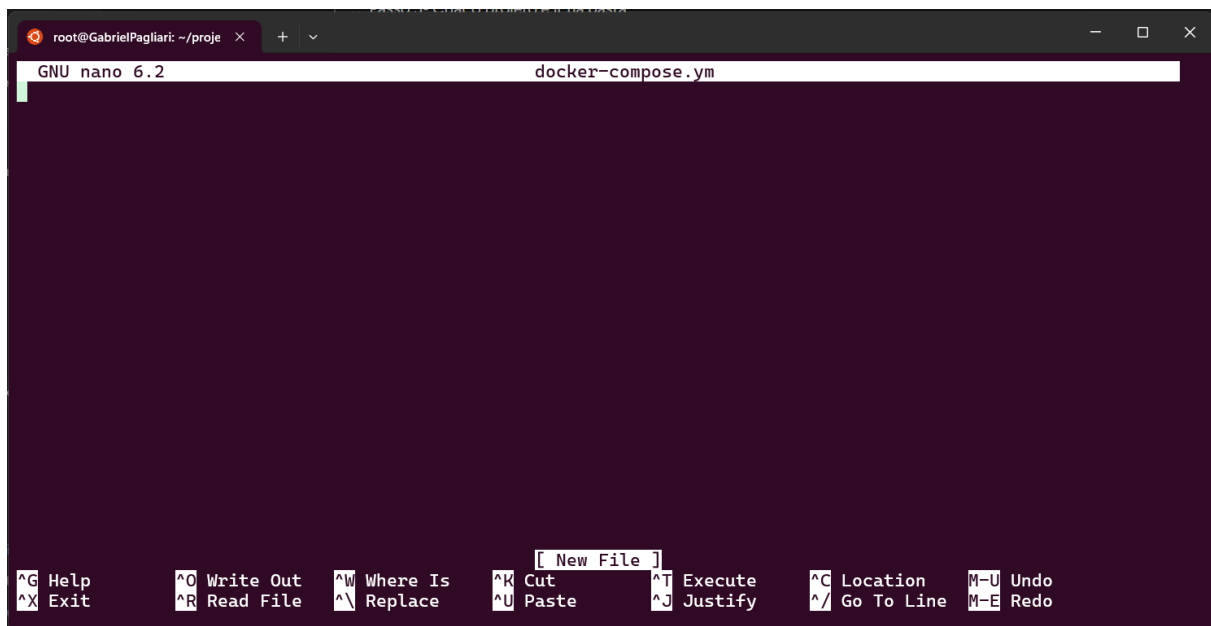
`mkdir projeto_final`

`cd projeto_final`

```
root@GabrielPagliari: ~/proje  
root@GabrielPagliari:~# sudo docker swarm --172.27.9.231  
unknown flag: --172.27.9.231  
See 'docker swarm --help'.  
  
Usage:  docker swarm COMMAND  
  
Manage Swarm  
  
Commands:  
  init      Initialize a swarm  
  join      Join a swarm as a node and/or manager  
  
Run 'docker swarm COMMAND --help' for more information on a command.  
  
root@GabrielPagliari:~# sudo docker swarm init --advertise-addr 172.27.9.231  
Swarm initialized: current node (j6306actocnx68bgeqmh62rmv) is now a manager.  
  
To add a worker to this swarm, run the following command:  
  
    docker swarm join --token SWMTKN-1-2pop2uf3bjs1nkt6443y862jq90qy6qld4hugimj8w2jpkfsna-0vjt5g1pek2gbtkcpbdk  
olw0b 172.27.9.231:2377  
  
To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.  
  
root@GabrielPagliari:~# mkdir projeto_final  
root@GabrielPagliari:~# cd projeto_final  
root@GabrielPagliari:~/projeto_final#
```

## Passo 4- Criar o arquivo docker-compose.yml:

`nano docker-compose.ym`



Passo 5- Configure dessa forma:

version: '3.8'

services:

wordpress:

image: wordpress:latest

ports:

- "8080:80"

environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: wordpress

WORDPRESS\_DB\_PASSWORD: wordpress\_password

WORDPRESS\_DB\_NAME: wordpress

volumes:

- wordpress\_data:/var/www/html

deploy:

replicas: 1

restart\_policy:

condition: on-failure

db:

image: mysql:5.7

environment:

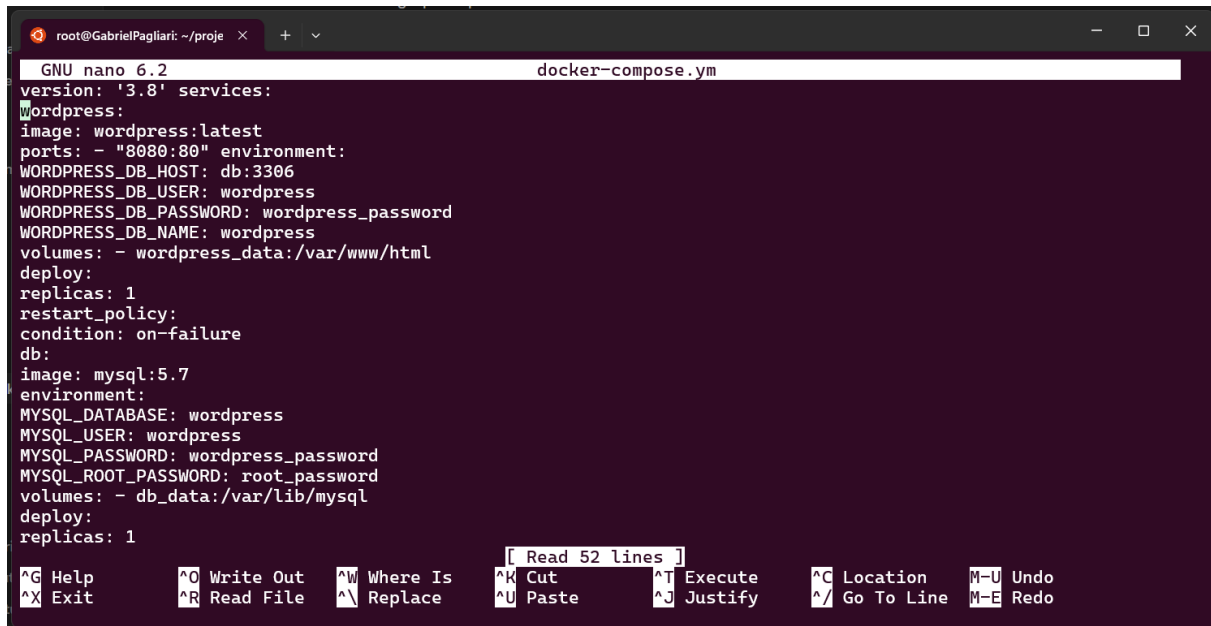
MYSQL\_DATABASE: wordpress

MYSQL\_USER: wordpress

MYSQL\_PASSWORD: wordpress\_password

MYSQL\_ROOT\_PASSWORD: root\_password  
volumes:  
- db\_data:/var/lib/mysql  
deploy:  
replicas: 1  
restart\_policy:  
condition: on-failure  
  
redis:  
image: redis:latest  
ports:  
- "6379:6379"  
volumes:  
- redis\_data:/data  
deploy:  
replicas: 1  
restart\_policy:  
condition: on-failure  
  
prometheus:  
image: prom/prometheus:latest  
ports:  
- "9090:9090"  
volumes:  
- ./prometheus.yml:/etc/prometheus/prometheus.yml  
deploy:  
replicas: 1  
restart\_policy:  
condition: on-failure  
  
grafana:  
image: grafana/grafana:latest  
ports:  
- "3000:3000"  
environment:  
- GF\_SECURITY\_ADMIN\_PASSWORD=admin  
volumes:  
- grafana\_data:/var/lib/grafana  
deploy:  
replicas: 1

restart\_policy:  
condition: on-failure  
  
volumes:  
wordpress\_data:  
db\_data:  
redis\_data:  
grafana\_data:



```
root@GabrielPagliari: ~/proje
GNU nano 6.2 docker-compose.yml
version: '3.8' services:
wordpress:
image: wordpress:latest
ports: - "8080:80" environment:
WORDPRESS_DB_HOST: db:3306
WORDPRESS_DB_USER: wordpress
WORDPRESS_DB_PASSWORD: wordpress_password
WORDPRESS_DB_NAME: wordpress
volumes: - wordpress_data:/var/www/html
deploy:
replicas: 1
restart_policy:
condition: on-failure
db:
image: mysql:5.7
environment:
MYSQL_DATABASE: wordpress
MYSQL_USER: wordpress
MYSQL_PASSWORD: wordpress_password
MYSQL_ROOT_PASSWORD: root_password
volumes: - db_data:/var/lib/mysql
deploy:
replicas: 1
[ Read 52 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
```

## Passo 6- Criar e configurar o Prometheus:

nano prometheus.yml

global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: 'prometheus'  
static\_configs:
  - targets: ['localhost:9090']
- job\_name: 'wordpress'  
static\_configs:
  - targets: ['wordpress:80']
- job\_name: 'mysql'  
static\_configs:

- targets: ['db:3306']
- job\_name: 'redis'  
static\_configs:
  - targets: ['redis:6379']
- job\_name: 'docker'  
static\_configs:
  - targets: ['docker:9323']

```

GNU nano 6.2 prometheus.yml
global:
scrape_interval: 15s
scrape_configs: - job_name: 'prometheus' static_configs: - targets: ['localhost:9090']
- job_name: 'wordpress' static_configs: - targets: ['wordpress:80']
- job_name: 'mysql' static_configs: - targets: ['db:3306']
- job_name: 'redis' static_configs: - targets: ['redis:6379']
- job_name: 'docker' static_configs: - targets: ['docker:9323']
metrics_path:

```

## Passo 7- Iniciar o Docker

sudo systemctl start docker

## Passo 8- Iniciar o Docker Swarm

sudo docker swarm init

```
root@GabrielPagliari: ~/proje
Commands:
  init      Initialize a swarm
  join      Join a swarm as a node and/or manager

Run 'docker swarm COMMAND --help' for more information on a command.

root@GabrielPagliari:~# sudo docker swarm init --advertise-addr 172.27.9.231
Swarm initialized: current node (j6306actocnx68bgeqmh62rmv) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-2pop2uf3bjs1nkt6443y862jq90qy6qld4hugimj8w2jpkfsna-0vjt5g1pek2gbtkcpbdk
    olw0b 172.27.9.231:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

root@GabrielPagliari:~# mkdir projeto_final
root@GabrielPagliari:~# cd projeto_final
root@GabrielPagliari:~/projeto_final# nano docker-compose.yml
root@GabrielPagliari:~/projeto_final# nano docker-compose.yml
root@GabrielPagliari:~/projeto_final# nano prometheus.yml
root@GabrielPagliari:~/projeto_final# sudo systemctl start docker
root@GabrielPagliari:~/projeto_final# sudo docker swarm init
Error response from daemon: This node is already part of a swarm. Use "docker swarm leave" to leave this swarm
and join another one.
root@GabrielPagliari:~/projeto_final#
```

Passo 9- Colocar os serviços:

`sudo docker stack deploy -c docker-compose.yml swarmservice`

```
root@GabrielPagliari: ~/proje
condition: on-failure
environment:
  WORDPRESS_DB_HOST: db:3306
  WORDPRESS_DB_NAME: wordpress
  WORDPRESS_DB_PASSWORD: wordpress_password
  WORDPRESS_DB_USER: wordpress
image: wordpress:latest
ports:
  - published: 8080
    target: 80
volumes:
  - wordpress_data:/var/www/html:rw
version: '3.8'
volumes:
  db_data: {}
  grafana_data: {}
  redis_data: {}
  wordpress_data: {}

root@GabrielPagliari:~/projeto_final# sudo docker stack deploy -c docker-compose.yml swarmservice
Creating network swarmservice_default
Creating service swarmservice_prometheus
Creating service swarmservice_grafana
Creating service swarmservice_wordpress
Creating service swarmservice_db
Creating service swarmservice_redis
root@GabrielPagliari:~/projeto_final#
```

Passo 10- Verificando:

`sudo docker stack services swarmservice`

```
Windows PowerShell x root@GabrielPagliari: ~ + v
root@GabrielPagliari:~# sudo docker swarm init
Error response from daemon: This node is already part of a swarm. Use "docker swarm leave" to leave this swarm
and join another one.
root@GabrielPagliari:~# sudo docker swarm init --advertise-addr 172.27.9.231
Error response from daemon: This node is already part of a swarm. Use "docker swarm leave" to leave this swarm
and join another one.
root@GabrielPagliari:~# nano prometheus.yml
root@GabrielPagliari:~# sudo docker service update --config-rm prometheus_config --config-add source=prometheu
s.yml,target=/etc/prometheus/prometheus.yml swarmservice_prometheus
config not found: prometheus.yml
root@GabrielPagliari:~# sudo docker stack services swarmservice
ID                NAME                MODE                REPLICAS            IMAGE                PORTS
6l81jvkujtq       swarmservice_db      replicated          1/1                 mysql:5.7
rljl7wm2s2fc       swarmservice_grafana replicated          1/1                 grafana/grafana:latest *:3000->3000/tcp
sukvsrsz32al       swarmservice_prometheus replicated          1/1                 prom/prometheus:latest *:9090->9090/tcp
f6czs3cgutuo       swarmservice_redis    replicated          1/1                 redis:latest          *:6379->6379/tcp
6w33mbaalndq       swarmservice_wordpress replicated          1/1                 wordpress:latest       *:8080->80/tcp
root@GabrielPagliari:~#
```