

Sistemas Operacionais de Redes II

Avaliação 9

Amanda Sá de Carvalho

Docker Compose

Requisitos

Instalar o Docker Compose através do Docker Desktop

Install on Ubuntu

This page contains information on how to install, launch and upgrade Docker Desktop on an Ubuntu distribution.

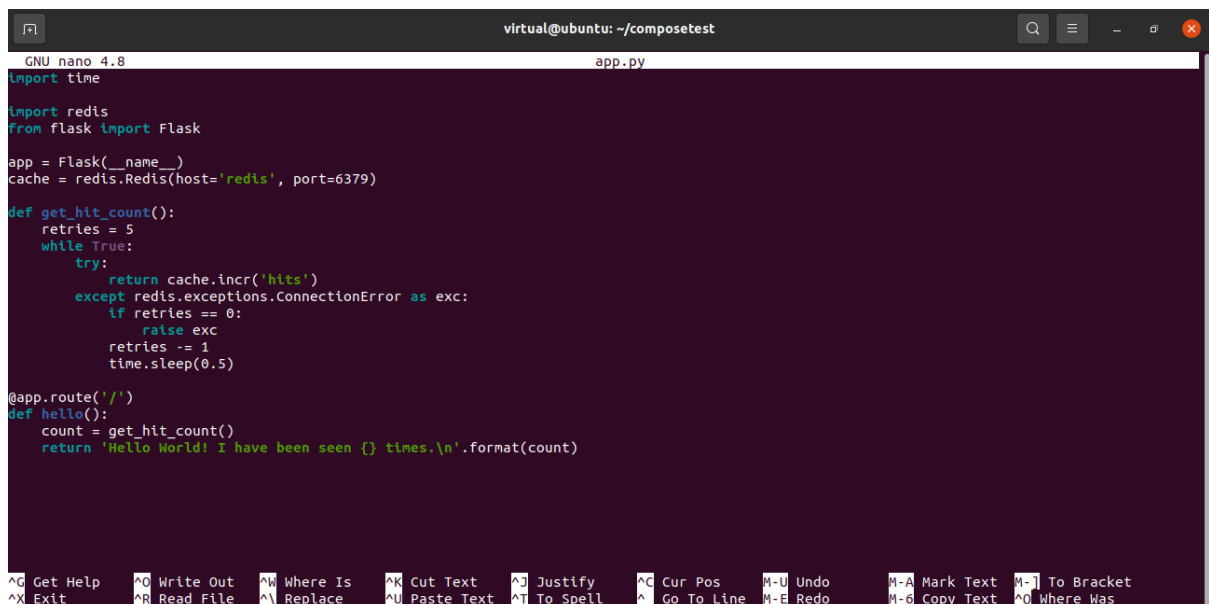
DEB package

Aplicação de Exemplo com Docker Compose

Criar o diretório do nosso exemplo

```
virtual@ubuntu:~$ mkdir composetest
virtual@ubuntu:~$ cd composetest/
virtual@ubuntu:~/composetest$
```

Criar um arquivo app.py para ser nossa aplicação teste



```
virtual@ubuntu: ~/composetest
GNU nano 4.8 app.py
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 'Hello World! I have been seen {} times.\n'.format(count)
```

Get Help Exit Write Out Read File Where Is Replace Cut Text Paste Text Justify To Spell Cur Pos Go To Line Undo Redo Mark Text Copy Text To Bracket Where Was

Criar um requirements.txt

```
virtual@ubuntu: ~/composetest
GNU nano 4.8 requirements.txt
flask
redis
```

Criar o Dockerfile do exemplo

```
virtual@ubuntu: ~/composetest
GNU nano 4.8 Dockerfile Modified
# syntax=docker/dockerfile:1
FROM python:3.7-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"]
```

Criar o arquivo docker-compose.yml

```
virtual@ubuntu: ~/composetest
GNU nano 4.8 docker-compose.yml
version: "3.9"
services:
  web:
    build: .
    ports:
      - "8000:5000"
  redis:
    image: "redis:alpine"
```

Rodar o comando docker compose up para subir nossa aplicação

```
virtual@ubuntu:~/composetest$ docker compose up
[+] Running 7/7
  :: redis Pulled                                4.6s
  :: c158987b0551 Pull complete                  1.2s
  :: 1a990ecc86f0 Pull complete                  1.3s
  :: f2520a938316 Pull complete                  1.6s
  :: f7ed7b580c2c Pull complete                  2.3s
  :: d48b7a41c3f1 Pull complete                  2.3s
  :: b7e40f3e68f9 Pull complete                  2.4s
[+] Building 26.5s (14/15)
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 320B                                                0.0s
=> [internal] load .dockerignore                                                    0.0s
=> => transferring context: 2B                                                      0.0s
=> resolve image config for docker.io/docker/dockerfile:1                        1.0s
=> docker-image://docker.io/docker/dockerfile:1@sha256:9ba7531bd80fb0a858632727cf7a112fbfd19b17e94c4e84ced81e24ef1a0dbc 1.7s
=> => resolve docker.io/docker/dockerfile:1@sha256:9ba7531bd80fb0a858632727cf7a112fbfd19b17e94c4e84ced81e24ef1a0dbc 0.0s
=> => sha256:1328b32c40fca9bcf9d70d8eccb72eb873d1124d72dadce04db8badbe7b08546 9.94MB / 9.94MB 1.4s
=> => sha256:9ba7531bd80fb0a858632727cf7a112fbfd19b17e94c4e84ced81e24ef1a0dbc 2.00kB / 2.00kB 0.0s
=> => sha256:ad87fb03593d1b71f9a1cfc1406c4aafcb253b1dabebf569768d6e6166836f34 528B / 528B 0.0s
=> => sha256:1e8a16826fd1c80a63fa6817a9c7284c94e40cdd14a9b0d0d3722356efa47bd 2.37kB / 2.37kB 0.0s
=> => extracting sha256:1328b32c40fca9bcf9d70d8eccb72eb873d1124d72dadce04db8badbe7b08546 0.3s
=> [internal] load .dockerignore                                                    0.0s
=> [internal] load build definition from Dockerfile                                0.0s
=> [internal] load metadata for docker.io/library/python:3.7-alpine                1.5s
=> [1/6] FROM docker.io/library/python:3.7-alpine@sha256:f1878142fbb04f418c3f67f78708559887ed5487dbe9bfc191e726548c3f89e 21.2s
=> => resolve docker.io/library/python:3.7-alpine@sha256:f1878142fbb04f418c3f67f78708559887ed5487dbe9bfc191e726548c3f89e 0.0s
=> => sha256:23de60bccb155d303f6e3517724b5370da3ee7fe4654053eb20bca6c6535dfe 7.71kB / 7.71kB 0.0s
=> => sha256:7ec3a91b66d9050989fca52618b7d400610c8c2cf9a8bbcb86bc403e0e01f220 622.90kB / 622.90kB 0.3s
=> => sha256:806cc3ffa52139640aa9976c06a5bedda84b66bfd9ac5298615b121d72a1e082 11.05MB / 11.05MB 1.9s
=> => sha256:9a88ff2414d3de106670f5faah6ceh9034b667fc2ca491528e520eb706f0fc6e 229B / 229B 0.4s
```

Verificar a aplicação rodando na porta 8000

```
← → ↻ 127.0.0.1:8000 ☆ 🖨 ☰
Hello World! I have been seen 3 times.
```

Podemos ver com o comando docker image ls nossas imagens de teste rodando

```
virtual@ubuntu:~$ docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
composetest-web     latest      8072eb349a13  3 minutes ago  215MB
redis               alpine      8ace02fae412  17 hours ago   29.9MB
mysql/mysql-server  latest      3f3946d5572f  2 months ago   517MB
hello-world         latest      feb5d9fea6a5  14 months ago  13.3kB
```

Entendendo o Docker Compose

Com esse exemplo simples podemos ver como o docker compose funciona, ele é um orquestrador de containers, no nosso caso só foi levantado 2 containers com o docker compose, mas é possível instanciar inúmeros containers ao mesmo tempo.

Todos eles são configurados dentro do arquivo docker-compose que é escrito em YAML, assim é possível que eles sejam configurados e integrados em um só lugar.

O docker compose pode ser usado em diversas aplicações web, como uma que usa um servidor e um banco de dados, ou então uma API em Python usando Flask, etc.

O docker compose tem uma sintaxe fácil de entender e uma documentação ampla, assim sendo fácil o seu aprendizado e implementação nas aplicações web atualmente.

Portanto o aprendizado de docker compose é essencial para quem se interessa pela web e deseja facilitar ainda mais suas aplicações e torná-las flexíveis e extensíveis.