

Para crear el apache:

- sudo apt update
- sudo apt install apache2
-

verificar si apache está activo: sudo systemctl status apache2

Para crear el Docker compose:

- Se crea un 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)
```

En este ejemplo, redis es el nombre del host del contenedor de redis en la red de la aplicación. Utilizamos el puerto por defecto para Redis, 6379.

- se crea un archivo requirements.txt

flask

redis

- Ahora creamos la dockerfile, la cual llamamos Dockerfile

```
# 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"]
```

Esto le dice a Docker que:

Construya una imagen empezando por la imagen de Python 3.7.

Establecer el directorio de trabajo a /code.

Establecer las variables de entorno utilizadas por el comando flask.

Instalar gcc y otras dependencias

Copiar requirements.txt e instalar las dependencias de Python.

Añadir metadatos a la imagen para describir que el contenedor está escuchando en el puerto 5000

Copiar el directorio actual . en el proyecto al workdir . en la imagen.

Establecer el comando por defecto para el contenedor a flask run.

- creamos un archivo de docker-compose.yml

```
version: "3.9"

services:

  web:

    build: .

    ports:

      - "8000:5000"

  redis:

    image: "redis:alpine"
```

Este archivo compose define dos servicios: el **web** y el **redis**.

El servicio **web** utiliza una imagen construida a partir del Dockerfile en el directorio actual. Luego vincula el contenedor y la máquina anfitriona al puerto expuesto, 8000.

El servicio **redis** utiliza una imagen pública de Redis extraída del registro de Docker Hub.

Instalar el nginx:

- sudo apt update
- sudo apt install nginx

El proxy:

Esta da permisos de lectura y escritura

- sudo chmod a+rw

Esta hace la config del server

- nano cd /etc/nginx/sites-available/ejemplo.com.conf
/etc/nginx/sites-enabled/ejemplo.com.conf

recarga nginx para aplicar cualquier cambio hecho

- `sudo systemctl reload nginx`