*Pràctica: “Crear docker-compose i dockerfile”*

**Índex**

[1- Crear un fitxer app.py: 2](#_heading=h.gjdgxs)

[2- Crear un fitxer requirements.txt: 2](#_heading=h.30j0zll)

[3- Crear un fitxer Dockerfile: 3](#_heading=h.1fob9te)

[4- Definir els serveis en un fitxer docker-compose.yml: 3](#_heading=h.3znysh7)

[5- Construir i fer córrer la app amb Compose: 5](#_heading=h.2et92p0)

1. **Crear un fitxer 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 exec

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)

1. **Crear un fitxer requirements.txt:**

Dins del fitxer afegeix aquestes dues llibreries.

flask redis

1. **Crear un fitxer Dockerfile:**

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

COPY . .

CMD ["flask", "run"]

* Què fa el fitxer Dockerfile?
  + Construeix una imatge començant amb un **python:3.7-alpine**
  + Defineix el directori de treball /code
  + Defineix les variables d’entorn definides per la comanda flask
  + Instal·la gcc per tal que els paquets python com MarkupSafe i SQLAlchemy puguin ser compilats
  + Copia el fitxer requirements.txt
  + Instal·la les dependències de python
  + Copia el directori actual al directori de treball . de la imatge
  + Defineix la comanda per defecte pel contenidor: flask run

Tasques:

* 1).- Genera la imatge a partir del Dockerfile.
* 2).- Connecta al contenidor per bash i fes un print-screen de com s’han copiat els fitxers al contenidor.
* 3).- Guarda la imatge a Docker Hub

1. **Definir els serveis en un fitxer docker-compose.yml:**

services:

web:

build: .

ports:

- "5000:5000"

redis:

image: "redis:alpine"

Aquest fitxer defineix dos serveis:

* Servei **web**: Fa servir una imatge que serà construïda a partir del Dockerfile. Després defineix el port de treball i que exposa que serà el mateix i correspon al port per defecte d’un Flask Web Server
* Servei **redis**: Fa servir una imatge pública de redis que es descarregarà del repositori públic dockerhub

1. **Construir i fer córrer la app amb Docker Compose:**

Executar la comanda docker compose up -d

Accedir a IP\_Docker:5000

Refrescar la pàgina i observar què passa

1. **Editar el fitxer docker-compose:**

services:

web:

build: .

ports:

- "5000:5000"

volumes:

.:/code

environment:

FLASK\_ENV: development

redis:

image: "redis:alpine"

* + Aquí hem creat un *bind mount* on fem que la carpeta de treball actual del nostre host sigui la carpeta /code del contenidor, de tal forma que tot el que hi hagi a la nostra carpeta de treball, estarà automàticament a la carpeta /code del contenidor.
  + També definim que flask treballi en mode desenvolupador i d’aquesta manera els canvis del host es guardaran automàticament dins del directori del contenidor.

1. **Re-construir:**

* Tornar a executar *docker compose up -d* i accedeix novament a la web
* Refrescar la pàgina
* Prova de canviar el fitxer app.py del teu host:Canvia el missatge del Hello from Docker i posa qualsevol altra cosa (return 'Hello from Docker! I have been seen {} times.\n'.format(count) )

Mira si el canvi ha tingut efecte sense tornar a construir la imatge.