Práctica 2: Contenedores Docker y Docker Compose

Autor: Manuel Díaz-Meco
Fecha: 3 de octubre de 2024

Introducción

El objetivo de esta práctica es trabajar con Docker para la creación de imágenes personalizadas, su publicación en Docker Hub, y el despliegue de múltiples contenedores utilizando Docker-compose. Además, se realizará una prueba opcional de limitación de CPU y la subida del contenedor con los resultados obtenidos a Docker Hub.

Apartado 1: Crear una imagen Docker personalizada con Apache

En este apartado se crea una imagen que contiene Apache y un archivo personalizado `index.html` para ser servido en el puerto 8080. Estos han sido los comandos utilizados para construir y ejecutar el contenedor:

Comandos utilizados:

```
docker build -t pr2_pruebacpd .
docker run --name pr2_pruebaCPD -p8080:80 -d pr2_pruebacpd
```

Donde el archivo `index.html` ha sido el siguiente, siguiendo el guión:

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prueba CPD</title>
</head>
<body>
    <h1>Prueba inicial CPD, Manuel Diaz-Meco</h1>
</body>
</html>
```

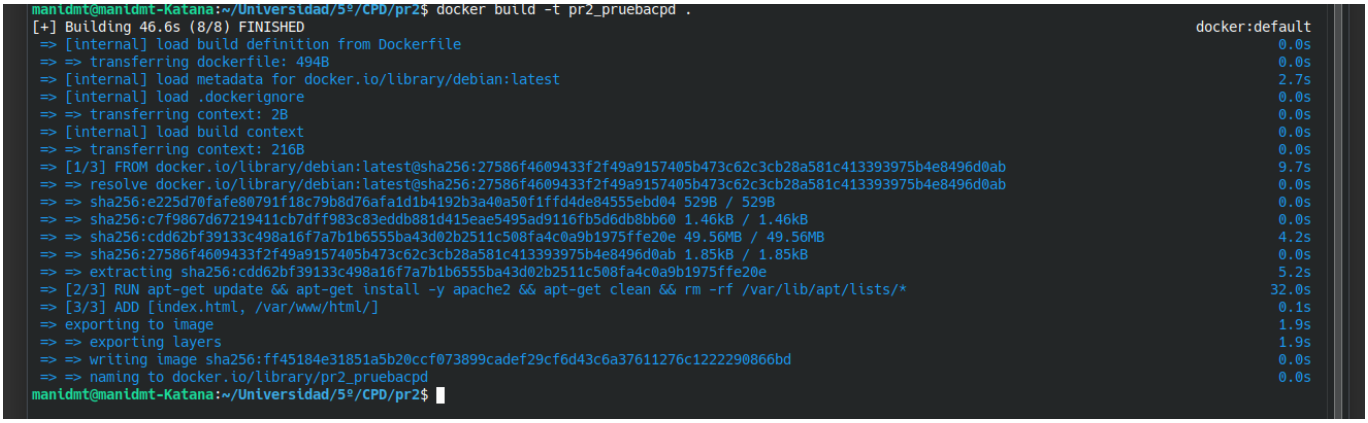
Y el archivo `Dockerfile` ha sido:

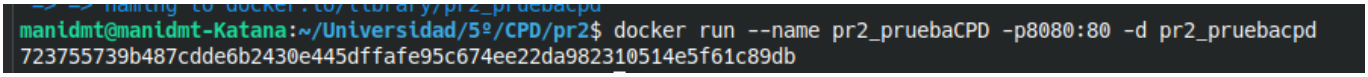
Dockerfile

```
FROM debian
# Parece ser que MAINTAINER está obsoleto
```

```
LABEL maintainer.name = "Manuel"
LABEL maintainer.email =  "<manidmt5@correo.ugr.es>"
# Instalamos apache2
RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf
/var/lib/apt/lists/*
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
EXPOSE 80
ADD ["index.html", "/var/www/html/"]
ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

Capturas de pantalla:





Apartado 2: Publicar la imagen en Docker Hub

En este apartado se publica la imagen construida en Docker Hub para que pueda ser utilizada desde cualquier lugar. El comando utilizado para la publicación fue el siguiente:

Comando utilizado:

```
docker push manidmt/pr2_pruebaCPD:1.0
```

Captura de pantalla:

```
manidmt@manidmt-Katana:~/Universidad/5º/CPD/pr2$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to
create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using
 SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: manidmt
Password:
WARNING! Your password will be stored unencrypted in /home/manidmt/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
manidmt@manidmt-Katana:~/Universidad/5º/CPD/pr2$ cd Imagen/
manidmt@manidmt-Katana:~/Universidad/5º/CPD/pr2/Imagen$ docker build -t manidmt/cpd1:1.0 .
[+] Building 8.5s (9/9) FINISHED                                                                                                          docker:default
 => [internal] load build definition from Dockerfile                                                                                               0.0s
 => => transferring dockerfile: 102B                                                                                                               0.0s
 => [internal] load metadata for docker.io/library/alpine:3.20                                                                                     1.4s
 => [auth] library/alpine:pull token for registry-1.docker.io                                                                                      0.0s
 => [internal] load .dockerignore                                                                                                                  0.0s
 => => transferring context: 2B                                                                                                                    0.0s
 => [1/4] FROM docker.io/library/alpine:3.20@sha256:beefdbd8a1da6d2915566fde36db9db0b524eb737fc57cd1367effd16dc0d06d                                0.0s
 => [2/4] RUN apk update                                                                                                                           1.7s
 => [3/4] RUN apk add curl                                                                                                                         2.2s
 => [4/4] RUN apk add vim                                                                                                                          2.4s
 => exporting to image                                                                                                                             0.7s
 => => exporting layers                                                                                                                            0.7s
 => => writing image sha256:a831722512e07df33af69232777c664f8c5aed56e19ecc93539e64f35fadccc5                                                       0.0s
 => => naming to docker.io/manidmt/cpd1:1.0                                                                                                        0.0s
manidmt@manidmt-Katana:~/Universidad/5º/CPD/pr2/Imagen$ docker push manidmt/cpd1:1.0
The push refers to repository [docker.io/manidmt/cpd1]
3e42144e6453: Pushed
62ffed2d1a43: Pushed
7a9450f7e1ba: Pushed
63ca1fbb43ae: Mounted from library/alpine
1.0: digest: sha256:61de14132e7bf2c6706f03dcea7006155d67c1279d90835efe45d0b6cbd5bdd6 size: 1161
manidmt@manidmt-Katana:~/Universidad/5º/CPD/pr2/Imagen$
```

Apartado 3: Desplegar un servidor Wordpress con Docker-compose

En este apartado se ha creado la correspondiente imagen para crear un servidor Wordpress, que luego se edita para que salga mi nombre, a través de `docker-compose` El archivo `docker-compose.yml` utilizado es el siguiente:

docker-compose.yml:

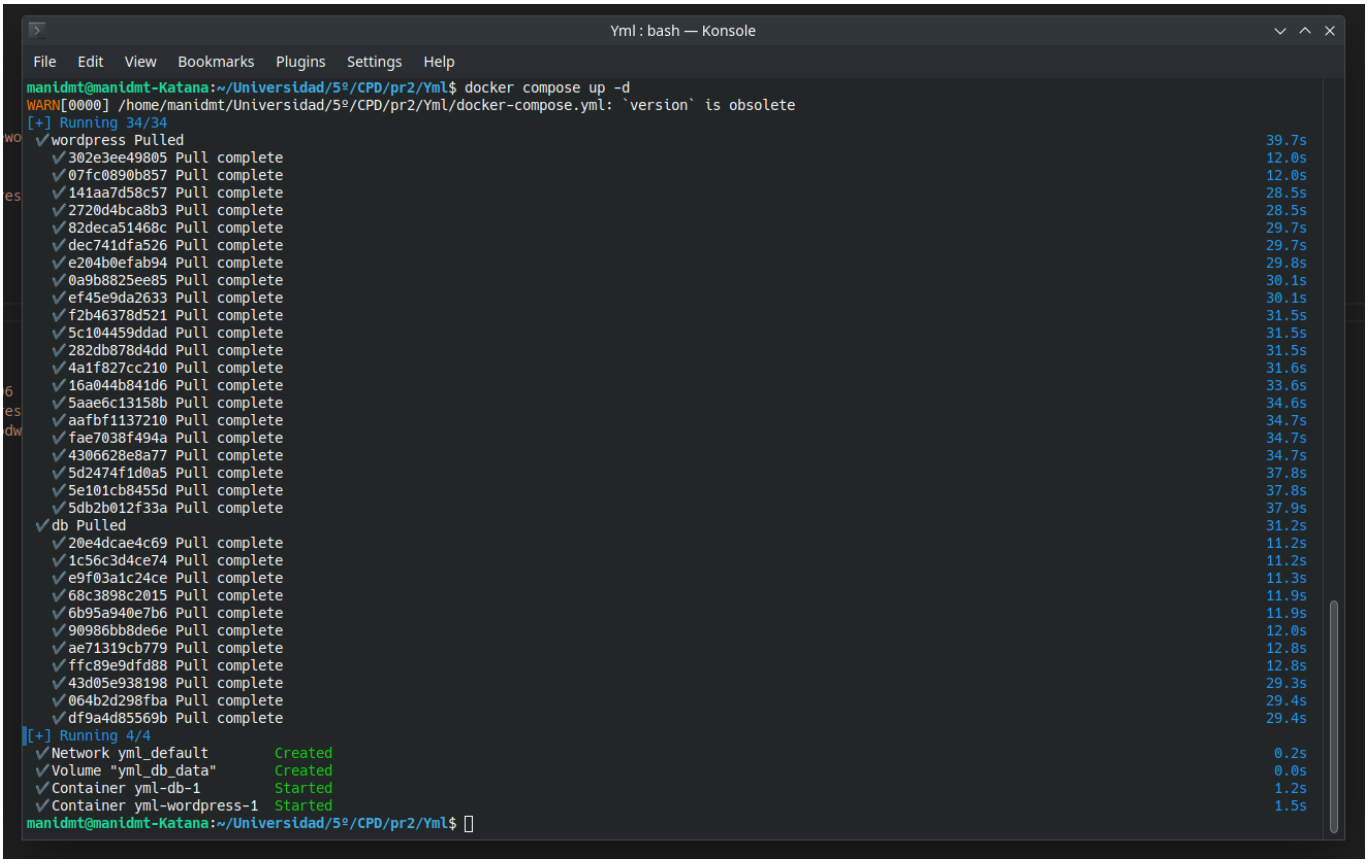
```yaml
version: '3'
services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
```

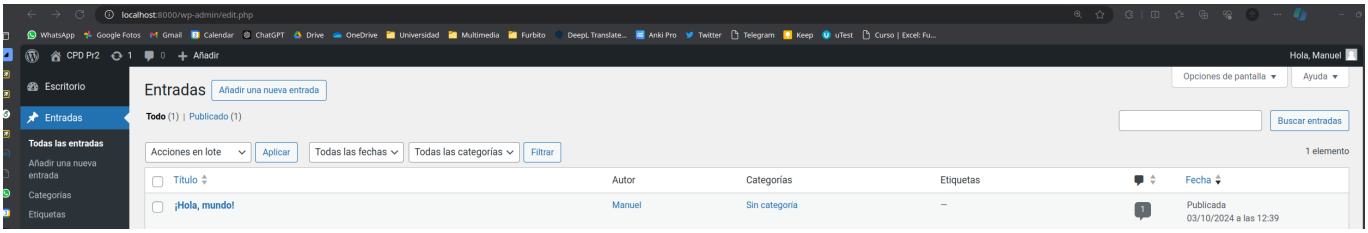
```
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
volumes:
  db_data:
```

Comando para ejecutar:

```
docker-compose up -d
```

Capturas de pantalla:





Apartado 4: Limitar el uso de CPU y realizar un benchmark

En este apartado opcional, se limitó el uso de CPU y se ejecutó el benchmark de CPU usando la herramienta `sysbench`, previamente instalada en el contenedor. A continuación se muestran los comandos utilizados y los resultados obtenidos.

Comando para ejecutar el benchmark sin limitación de CPU:

```
sysbench --test=cpu --cpu-max-prime=20000 --threads=6 run
```

Comando para ejecutar el benchmark con limitación de CPU (0.5 CPUs):

```
docker run --cpus=".5" ubuntu /bin/bash
sysbench --test=cpu --cpu-max-prime=20000 --threads=6 run
```

Capturas de pantalla:

```
root@bdcdf9948ea9:/# sysbench --test=cpu --cpu-max-prime=20000 --threads=6 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 6
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  2368.29

General statistics:
    total time:                          10.0021s
    total number of events:              23695

Latency (ms):
         min:                                    2.17
         avg:                                    2.53
         max:                                    7.40
         95th percentile:                        2.52
         sum:                                59993.66

Threads fairness:
    events (avg/stddev):           3949.1667/2.27
    execution time (avg/stddev):   9.9989/0.00
```

```
root@a0e3c82713a7:/# sysbench --test=cpu --cpu-max-prime=20000 --threads=6 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 6
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:    81.54

General statistics:
    total time:                          10.0607s
    total number of events:              821

Latency (ms):
         min:                                    3.77
         avg:                                   73.26
         max:                                  200.36
         95th percentile:                      104.84
         sum:                                60150.54

Threads fairness:
    events (avg/stddev):           136.8333/5.11
    execution time (avg/stddev):   10.0251/0.07

root@a0e3c82713a7:/#
```

```
manidmt@manidmt-Katana:~$ docker push manidmt/cpd_bechhalfcpu:1.0
The push refers to repository [docker.io/manidmt/cpd_bechhalfcpu]
24c299b80e91: Pushed
b15b682e901d: Mounted from library/ubuntu
1.0: digest: sha256:8ea741251e1349a12f2ab5e8824ec30b0705db70dff333d164f44b7790da0faa size: 741
manidmt@manidmt-Katana:~$
```

```
manidmt@manidmt-Katana:~/Universidad/5º/CPD/pr2/Yml$ lscpu
Architecture:             x86_64
  CPU op-mode(s):         32-bit, 64-bit
  Address sizes:          39 bits physical, 48 bits virtual
  Byte Order:             Little Endian
CPU(s):                   20
  On-line CPU(s) list:    0-19
Vendor ID:                GenuineIntel
  Model name:             13th Gen Intel(R) Core(TM) i7-13700H
    CPU family:           6
    Model:                186
    Thread(s) per core:   2
    Core(s) per socket:   14
    Socket(s):            1
    Stepping:             2
    CPU(s) scaling MHz:   10%
    CPU max MHz:          5000,0000
    CPU min MHz:          400,0000
    BogoMIPS:             5836,80
```
