**TALLER 18**

**5.**

docker-compose version 1.27.4, build 40524192

**9.**

- Se crea una imagen con python 3.7.

- Se selecciona /code como directorio de trabajo.

- Se seleccionan las variables de configuración usadas para flask run.

- Instalar gcc y dependencias.

- Se copia requirements.txt requirements.txt

- Se expone el puerto por el cual el contenedor escuchará.

- Se copia el directorio . actual al workdir . en la imagen

- Se escoge el comando por defecto del contenedor a flask run.

**10.**

- Se elige la versión 3.8 de Compose File.

- Se elegirán los servicios.

- web: el servicio web usa una imagen que se construye del Dockerfile del directorio actual. Luego enlaza el contenedor y la host machine al puerto expuesto, 5000.

- redis: el servicio redis usa una imagen pública Redis del registro de Docker Hub.

**12.**

El mensaje de la página se actualiza contando el número de visitas.

**13.**

REPOSITORY TAG IMAGE ID CREATED SIZE

test\_web latest 513b832cdde0 5 hours ago 195MB

<none> <none> 0e71dd56f1cd 5 hours ago 195MB

<none> <none> 3511c3dcab51 5 hours ago 195MB

<none> <none> 3124bb92316a 5 hours ago 195MB

python 3.7-alpine 6b73b71fd64e 3 days ago 41.1MB

redis alpine c1949ec48c51 10 days ago 31.2MB

hello-world latest bf756fb1ae65 10 months ago 13.3kB

**14.**

El modo de desarrollo funciona para facilitar la modificación del código principal sin tener que hacer rebuild a la imagen. Particularmente, da la instrucción de ejecutar flask run en modo desarrollador y recargar el código ante cambios.

**18.**

Name Command State Ports

------------------------------------------------------------------------------

test\_redis\_1 docker-entrypoint.sh redis ... Up 6379/tcp

test\_web\_1 flask run Up 0.0.0.0:5000->5000/tcp