Contenedores con Docker. Imágenes de aplicaciones y Dockerfile

1. **Uso de nginx**. Se trata de descargar la imagen del servicio **nginx** (servidor web con soporte para proxy inverso) desde el Docker Hub para ello:

   a) Descarga la imagen con docker pull nginx
   b) Ejecuta el comando:

   ```
   docker run -it -d --rm -p 80:80 --name web nginx
   ```

   En este caso se mapea el puerto 80 del contenedor al puerto 80 de la maquina anfitrión y se le pasa el nombre "web" al nuevo contenedor. Además en este caso se pasa el parámetro –rm que borra el contenedor una vez que se para

   c) Ejecuta el comando `docker ps` para comprobar que el nuevo contenedor está en ejecución
   d) Vamos a probar el acceso al servidor nginx y para ello en la maquina anfitrión tecleamos http://localhost en un navegador cualquiera
   e) Sabiendo que el directorio de publicación web del servidor nginx es `/usr/share/nginx/html/` se pide crear una página web de inicio y subirla al contendor usando el mapeo de carpetas mediante el parámetro –v ya visto en prácticas anteriores
   f) Lanza el nuevo contenedor con los parámetros adecuados y verifica que la nueva página es ya visible desde el navegador

2. **Creación y uso de Dockerfile**. Vamos a crear un fichero de nombre "Dockerfile" que nos permita automatizar la creación de una imagen personalizada de nginx que llevara una página web personalizada. Para ello:

   a) Creamos un fichero Dockerfile con este contenido:

   ```
   FROM nginx:latest
   COPY ./index.html /usr/share/nginx/html/index.html
   ```

   b) Creamos la imagen con:

   ```
   $ docker build -t webserver .
   ```

   Y veremos una salida similar a esta:

```
Sending build context to Docker daemon  3.072kB
Step 1/2 : FROM nginx:latest
latest: Pulling from library/nginx
bf5952930446: Pull complete
ba755a256dfe: Pull complete
c57dd87d0b93: Pull complete
d7fbf29df889: Pull complete
1f1070938ccd: Pull complete
Digest: sha256:36b74457bccb56fbf8b05f79c85569501b721d4db813b684391d63e02287c0b2
Status: Downloaded newer image for nginx:latest
 ---> 08393e824c32
Step 2/2 : COPY ./index.html /usr/share/nginx/html/index.html
 ---> fe915b3e4179
Successfully built fe915b3e4179
Successfully tagged webserver:latest
```

c) Verificamos que tenemos la imagen ya descargada y lanzamos de nuevo un contenedor a partir de la nueva imagen:

```
docker run -it --rm -d -p 80:80 --name web webserver
```

d) Desde un navegador probamos que el acceso a http://localhost muestra la página copiada

e) Finalmente vamos a etiquetar la imagen con el nombre de nuestro dockerid para que podamos subirla al DockerHub, para ello haremos:

- `docker login` (caso de no estar logeado)
- `docker tag webserver <dockerid>/webserver` (se cambia el nombre a la imagen poniéndole una nueva etiqueta)
- `docker push <dockerid>/webserver` (se sube la imagen al DockerHub)

f) Comprobar en el DockerHub que la nueva imagen ya está visible

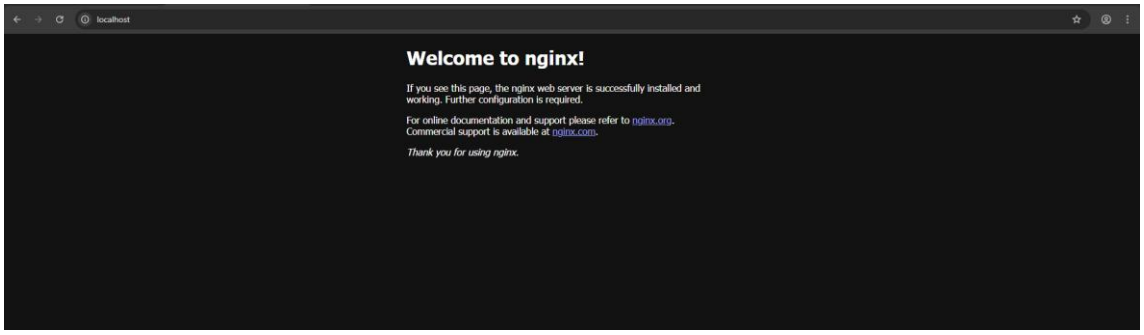
1.- Primero lanzamos el contenedor de nginx en el puerto 80.

```
H:\>docker run -it -d --rm -p 80:80 --name web nginx
```

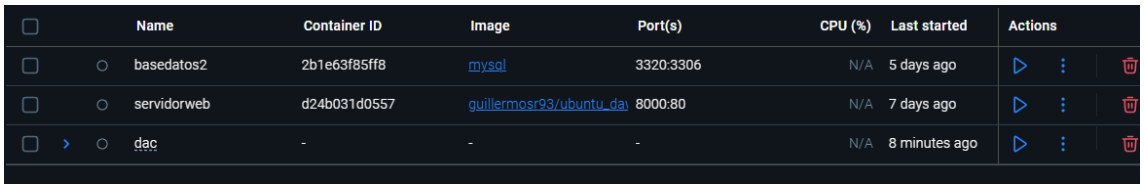
Con docker ps comprobamos que esté funcionando.

```
H:\>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                    NAMES
dc36ae7978ef   nginx          "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp       web
344b3e134469   mariadb:11.2   "docker-entrypoint.s…"   3 weeks ago     Up 4 minutes   0.0.0.0:3310->3306/tcp   mariadb
```

Ahora si vamos a localhost:80 nos aparecerá la página de inicio de nginx.



2.- Vamos a crear un volumen para enlazar una carpeta en nuestro pc, para ello detenemos el contenedor, como tenemos el comando –rm al pararlo se borra



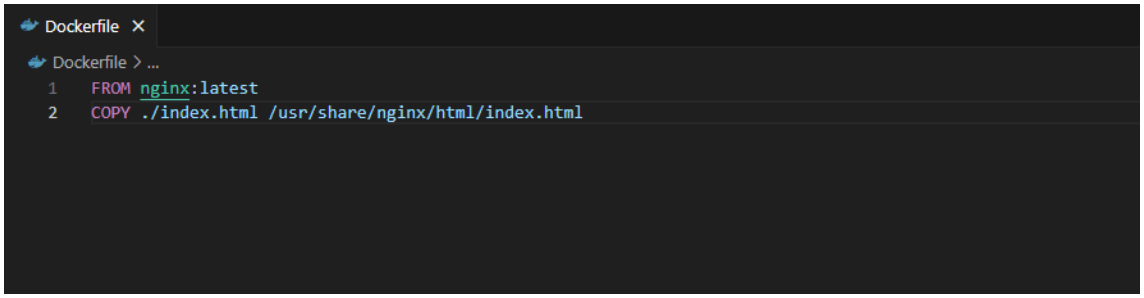
Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
basedatos2	2b1e63f85ff8	mysql	3320:3306	N/A	5 days ago	
servidorweb	d24b031d0557	guillermosr93/ubuntu_da	8000:80	N/A	7 days ago	
dac	-	-	-	N/A	8 minutes ago	

Ahora con -v, vinculamos nuestra carpeta del sistema con la que nos pide el ejercicio.

```
H:\>docker run -it -d --rm -p 80:80 -v c:\users\guillermosr26\web:/usr/share/nginx/html --name web nginx
```

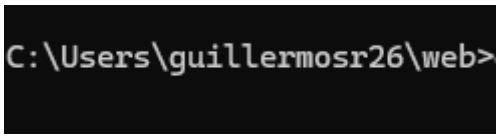
Si vamos a localhost nos saldrá ahora el index que teníamos en la carpeta web.

3.- Ahora vamos a abrir el visual para crear un fichero Dockerfile, yo abriré el visual en la carpeta web y creare el fichero Dockerfile, donde añadiré lo siguiente:

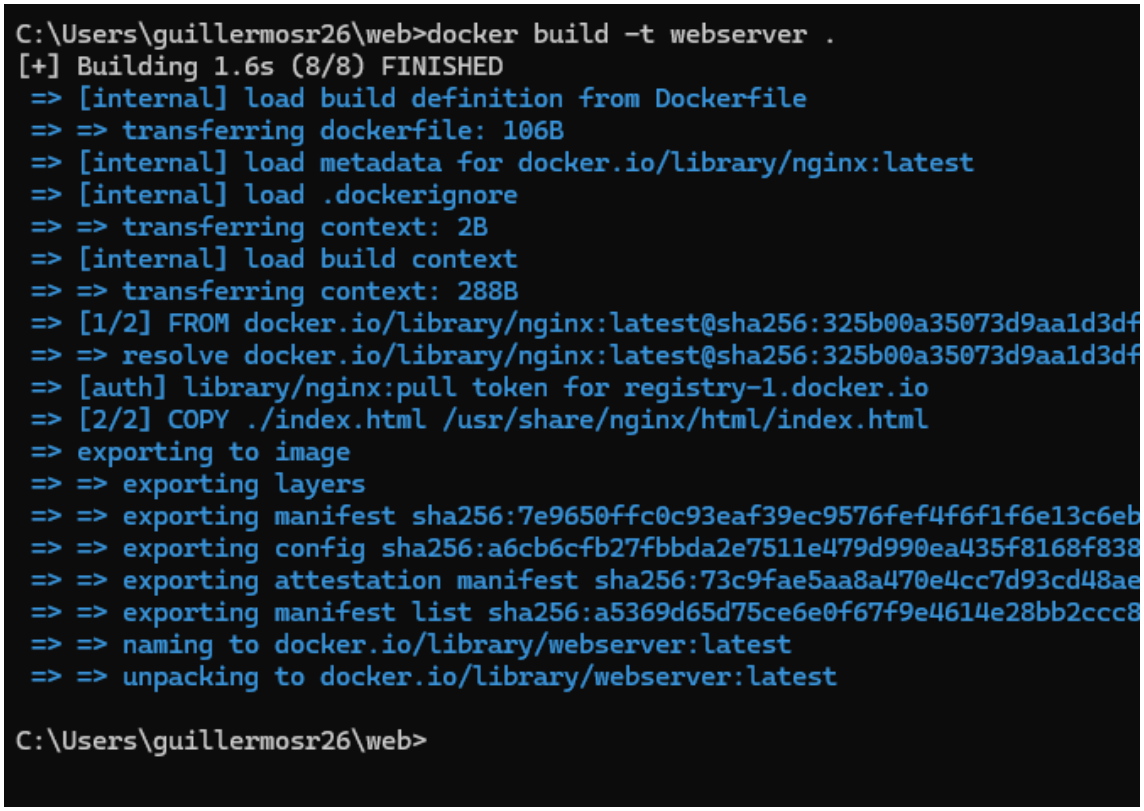


El . es para decir, desde este mismo directorio.

4.- Ahora iremos a nuestra carpeta de /web en la terminal.



Con el comando docker build construiremos nuestra imagen llamada webserver



Con docker images podemos comprobar que ya la hemos creado.

```
C:\Users\guillermosr26\web>docker images
REPOSITORY                  TAG       IMAGE ID       CREATED          SIZE
webserver                   latest    a5369d65d75c   24 seconds ago   225MB
nginx                       latest    325b00a35073   9 hours ago      225MB
mysql                       latest    fe036967257b   7 days ago       1.27GB
guillermosr93/ubuntu_daw2b  latest    6e1e1724daaa   11 days ago      365MB
ubuntu                      latest    c35e29c94501   7 weeks ago      117MB
hello-world                 latest    f7931603f70e   4 months ago     20.3kB
mariadb                     11.2      ff87d49107a1   13 months ago    551MB
```

Ahora vamos a volver a lanzar el contenedor, sin el comando -v para no vincularlo a ningún volumen.

```
C:\Users\guillermosr26\web>docker run -it -d --rm -p 80:80 --name web webserver
```

5.- Ahora vamos a cambiarle el nombre a nuestro contenedor con el comando docker tag.

```
C:\Users\guillermosr26\web>docker tag webserver guillermosr93/webserver
```

Con docker images vemos que nos ha creado una copia con nuestro nombre.

```
C:\Users\guillermosr26\web>docker images
REPOSITORY                  TAG       IMAGE ID       CREATED          SIZE
guillermosr93/webserver     latest    a5369d65d75c   7 minutes ago    225MB
webserver                   latest    a5369d65d75c   7 minutes ago    225MB
nginx                       latest    325b00a35073   9 hours ago      225MB
mysql                       latest    fe036967257b   7 days ago       1.27GB
guillermosr93/ubuntu_daw2b  latest    6e1e1724daaa   11 days ago      365MB
ubuntu                      latest    c35e29c94501   7 weeks ago      117MB
hello-world                 latest    f7931603f70e   4 months ago     20.3kB
mariadb                     11.2      ff87d49107a1   13 months ago    551MB
```

Y con el comando docker push subiremos a nuestro repositorio el archivo.

```
C:\Users\guillermosr26\web>docker push guillermosr93/webserver
```

Para comprobar que se ha subido ejecutaremos el comando docker search.

```
C:\Users\guillermosr26\web>docker search guillermosr93
NAME                          DESCRIPTION   STARS     OFFICIAL
guillermosr93/ubuntu_daw2b                  0
guillermosr93/webserver                     0

C:\Users\guillermosr26\web>
```