

Ejercicio 5 - Imagen con Dockerfile - Aplicación web

Realizado por Pablo Merino y Yolanda Frexes.

Arranca un contenedor que ejecute una instancia de la imagen `php:7.4-apache`, que se llame `web` y que sea accesible desde un navegador en el puerto `8000`.

```
$ docker run -d -p 8000:80 --name web php:7.4-apache
```

Coloca en el directorio raíz del servicio web (`/var/www/html`) un sitio web donde figure el nombre de los componentes del grupo - el sitio deberá tener al menos un archivo `index.html` y un archivo `.css`

```
root@78f4837a5aa6:/var/www/html# echo '<!DOCTYPE html><html lang=en><head><meta charset=UTF-8><meta name=viewport content=width=device-width, initial-scale=1.0><title>YoliYPablo</title><link rel=stylesheet href=styles.css></head><body><h1>GRupo formado por Yolanda Frexes y Pablo Merino</h1></body></html>' > index.html
root@78f4837a5aa6:/var/www/html# ls
index.html
```

```
root@78f4837a5aa6:/var/www/html# echo 'body { background-color: #fofofo; font-family: Arial, sans-serif; } h1 { color: blue; }' > styles.css
root@78f4837a5aa6:/var/www/html# ls
index.html  styles.css
root@78f4837a5aa6:/var/www/html#
```

```
$ echo '<!DOCTYPE html><html lang=en><head><meta charset=UTF-8><meta name=viewport content=width=device-width, initial-scale=1.0>
<title>YoliYPablo</title><link rel=stylesheet href=styles.css></head>
<body><h1>GRupo formado por Yolanda Frexes y Pablo Merino</h1></body>
</html>' > index.html

$ echo "body { background-color: #fofofo; font-family: Arial, sans-serif;
} h1 { color: blue; }" > styles.css
```



Coloca en ese mismo directorio raíz un archivo llamado `mes.php` que muestre el nombre del mes actual. Ver la salida del script en el navegador.

```
$ echo "<? php echo 'Estamos en el mes de ' . date('F'); ?>" > mes.php
```

```
root@78f4837a5aa6:/var/www/html# echo "<? php echo 'Estamos en el mes de ' . date('F'); ?>" > mes.php
root@78f4837a5aa6:/var/www/html# ls
index.html  mes.php  styles.css
root@78f4837a5aa6:/var/www/html#
```



Borrar el contenedor:

```
$ docker stop web
```

```
$ docker rm web
```

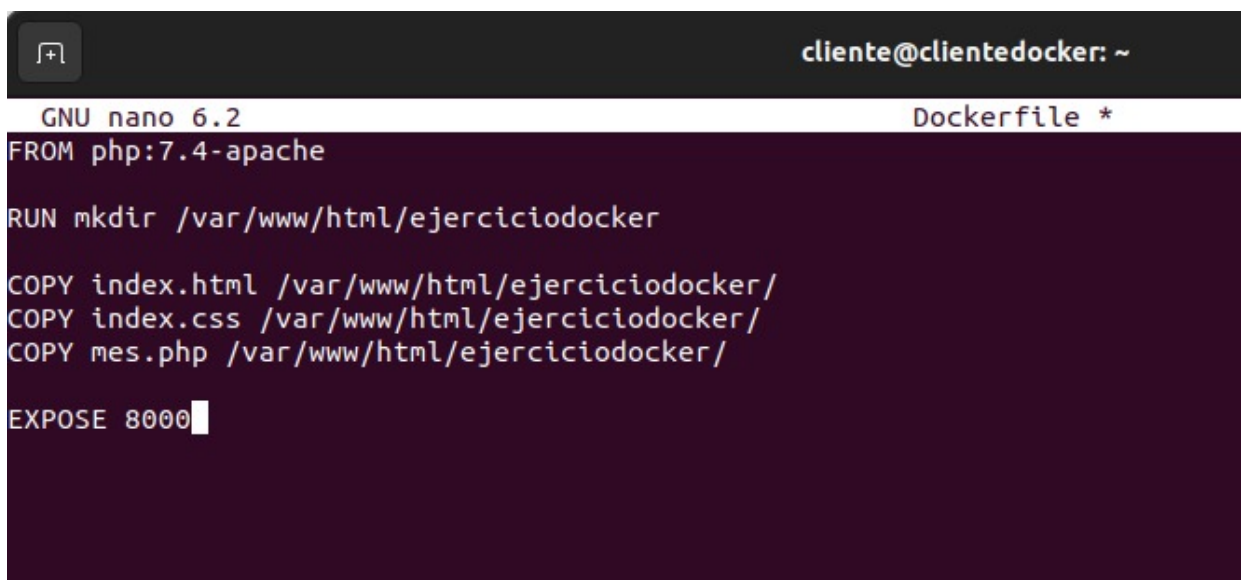
```
cliente@clientedocker:~$ docker stop web
web
cliente@clientedocker:~$ docker rm web
web
cliente@clientedocker:~$
```

Automatizar estas operaciones creando un fichero `Dockerfile`

Subir la imagen a la cuenta de Docker Hub

Deberás entregar los siguientes capturas de pantalla y los comandos empleados para resolver cada apartado:

1. Creación inicial del contenedor – documenta los pasos hasta el borrado del mismo
2. Bloque de código con el `Dockerfile`



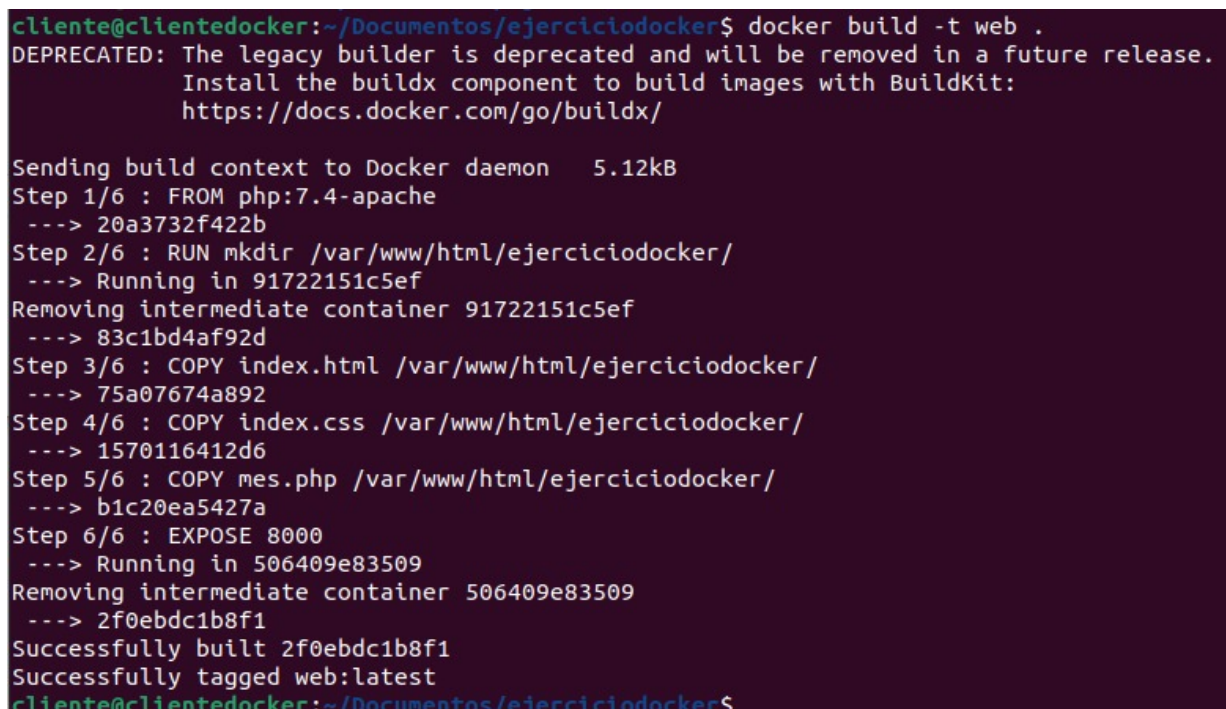
```
cliente@clientedocker: ~
GNU nano 6.2 Dockerfile *
FROM php:7.4-apache

RUN mkdir /var/www/html/ejerciciodocker

COPY index.html /var/www/html/ejerciciodocker/
COPY index.css /var/www/html/ejerciciodocker/
COPY mes.php /var/www/html/ejerciciodocker/

EXPOSE 8000
```

3. Captura de pantalla y documento donde se vea el comando que crea la nueva imagen.

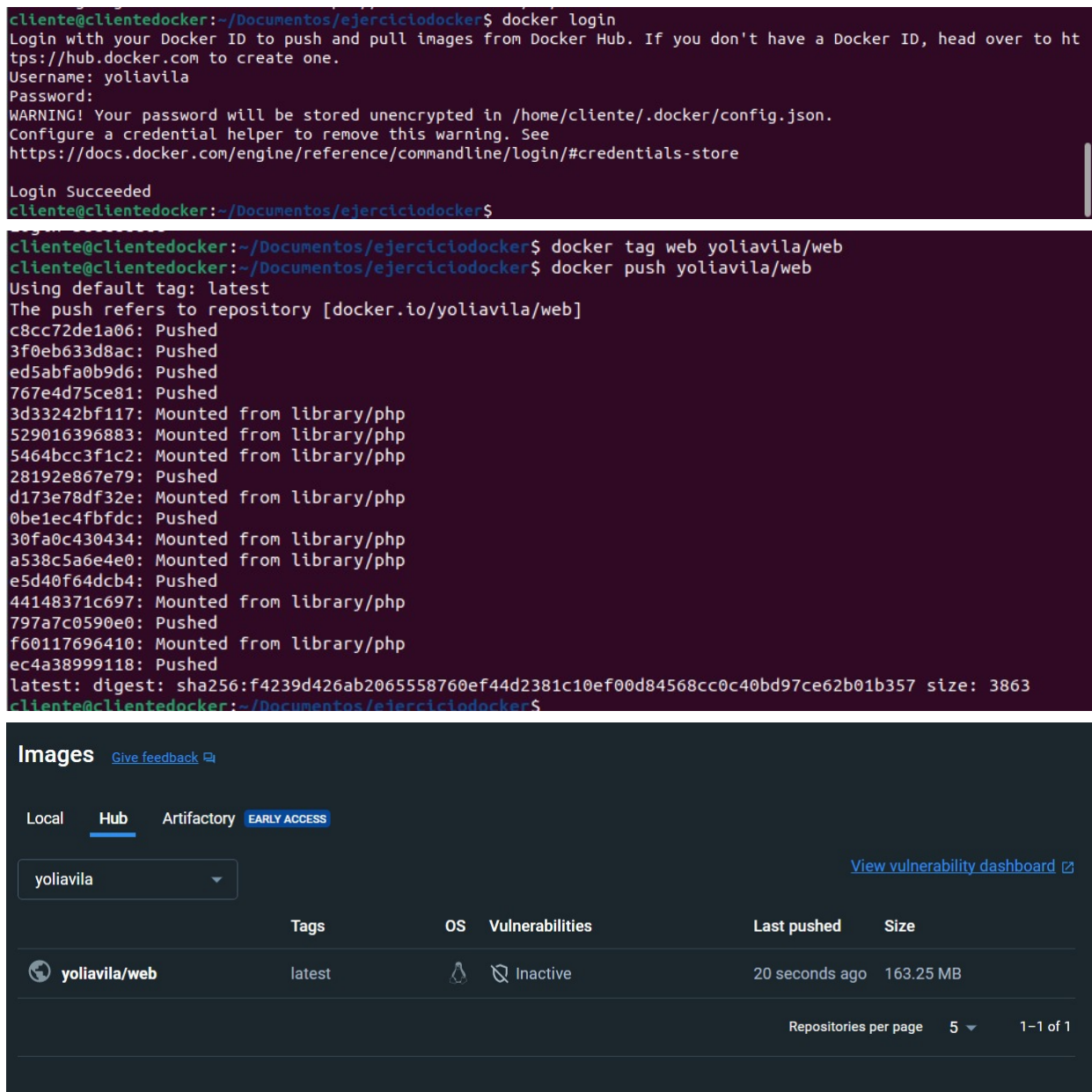


```
cliente@clientedocker:~/Documentos/ejerciciodocker$ docker build -t web .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 5.12kB
Step 1/6 : FROM php:7.4-apache
--> 20a3732f422b
Step 2/6 : RUN mkdir /var/www/html/ejerciciodocker/
--> Running in 91722151c5ef
Removing intermediate container 91722151c5ef
--> 83c1bd4af92d
Step 3/6 : COPY index.html /var/www/html/ejerciciodocker/
--> 75a07674a892
Step 4/6 : COPY index.css /var/www/html/ejerciciodocker/
--> 1570116412d6
Step 5/6 : COPY mes.php /var/www/html/ejerciciodocker/
--> b1c20ea5427a
Step 6/6 : EXPOSE 8000
--> Running in 506409e83509
Removing intermediate container 506409e83509
--> 2f0ebdc1b8f1
Successfully built 2f0ebdc1b8f1
Successfully tagged web:latest
cliente@clientedocker:~/Documentos/ejerciciodocker$
```



4. Captura de pantalla y documento donde se vea la imagen subida a tu cuenta de Docker Hub.



5. Captura de pantalla y documento donde se vea la bajada de la imagen – por parte de otra persona del grupo – y la creación de un contenedor.

```
serverdocker@serverdocker:~$ docker pull yoliavila/web
Using default tag: latest
latest: Pulling from yoliavila/web
a603fa5e3b41: Already exists
c428f1a49423: Already exists
156740b07ef8: Already exists
fb5a4c8af82f: Already exists
25f85b498fd5: Already exists
9b233e420ac7: Already exists
fe42347c4ecf: Already exists
d14eb2ed1e17: Already exists
66d98f73acb6: Already exists
d2c43c5efbc8: Already exists
ab590b48ea47: Already exists
80692ae2d067: Already exists
05e465aaa99a: Already exists
eb474b510b8d: Downloading      178B/178B
223c71c1faa4: Downloading      415B/415B
384982e6cf70: Downloading      278B/278B
63b1bec8de68: Downloading      265B/265B
write /var/lib/docker/tmp/GetImageBlob2913584021: no space left on device
serverdocker@serverdocker:~$
```

6. Captura de pantalla y documento donde se ve el acceso al navegador con el sitio servido

```
serverdocker@serverdocker:~$ docker run -d --name web -p 8000:80 yoliavila/web
Unable to find image 'yoliavila/web:latest' locally
latest: Pulling from yoliavila/web
a603fa5e3b41: Already exists
c428f1a49423: Already exists
156740b07ef8: Already exists
fb5a4c8af82f: Already exists
25f85b498fd5: Already exists
9b233e420ac7: Already exists
fe42347c4ecf: Already exists
d14eb2ed1e17: Already exists
66d98f73acb6: Already exists
d2c43c5efbc8: Already exists
ab590b48ea47: Already exists
80692ae2d067: Already exists
05e465aaa99a: Already exists
eb474b510b8d: Downloading      178B/178B
223c71c1faa4: Downloading      415B/415B
384982e6cf70: Downloading      278B/278B
63b1bec8de68: Downloading      265B/265B
docker: write /var/lib/docker/tmp/GetImageBlob2065105387: no space left on device.
See 'docker run --help'.
serverdocker@serverdocker:~$
```

Adjunto captura del comando que debería utilizarse para crear el contenedor pero debido a que no tengo espacio en la máquina virtual no he podido hacer la comprobación.