

Ejercicio 3 - imagen con Dockerfile - Aplicación web

Adrián García de la Cera

Ejercicio 3 - imagen con Dockerfile - Aplicación web

Enunciado

Desarrollo

Creación de archivos

Creación de la imagen

Comprobaciones

Subida de imágenes

Borrado y descarga de imagen

Enunciado

Necesitamos un fichero Dockerfile que automatice las siguientes operaciones para crear una imagen que contenga un servidor con un sitio web y un script php. Características de la imagen:

- Usa un contenedor que ejecute una instancia de la imagen `php:7.4-apache`, que se llame `ejercicio3` y que sea accesible desde un navegador en el puerto 8000.
- Coloca en el directorio raíz del servicio web (`/var/www/html`) un "sitio web" donde figure tu nombre - el sitio deberá tener al menos un archivo `index.html` sencillo y un archivo `.css`.
- Coloca en ese mismo directorio raíz el siguiente script php, llámalo `fecha.php`.

```
<?php
setlocale(LC_TIME, "es_ES.UTF-8");
$mes_actual = strftime("%B");
$fecha_actual = date("d/m/Y");
$hora_actual = date("H:i:s");
echo "<h1>Información</h1>";
echo "<p>Hoy es $fecha_actual</p>";
echo "<p>El mes es: <strong>$mes_actual</strong></p>";
echo "<p>Hora: $hora_actual</p>";
?>
```

- Ver la salida del script `fecha.php` y de la página `index.html` en el navegador.

Una vez creada la imagen, súbela a tu cuenta de Docker Hub

- Borra la imagen de tu Docker local
- Baja ('pull') de tu cuenta la imagen que acabas de subir
- Muestra las imágenes que tienes
- Ejecuta un contenedor usando esa imagen

Deberás entregar, al menos, las siguientes capturas de pantalla, los comandos empleados y/o operaciones con Docker Desktop para resolver cada apartado:

- creación inicial del contenedor - documenta los pasos hasta el borrado del mismo.
- bloque de código con el Dockerfile.
- creación de la nueva imagen.
- subida de la imagen a tu cuenta de Docker Hub.
- operación de 'pull' de la imagen de Docker Hub
- creación de un nuevo contenedor con esa imagen y su ejecución.

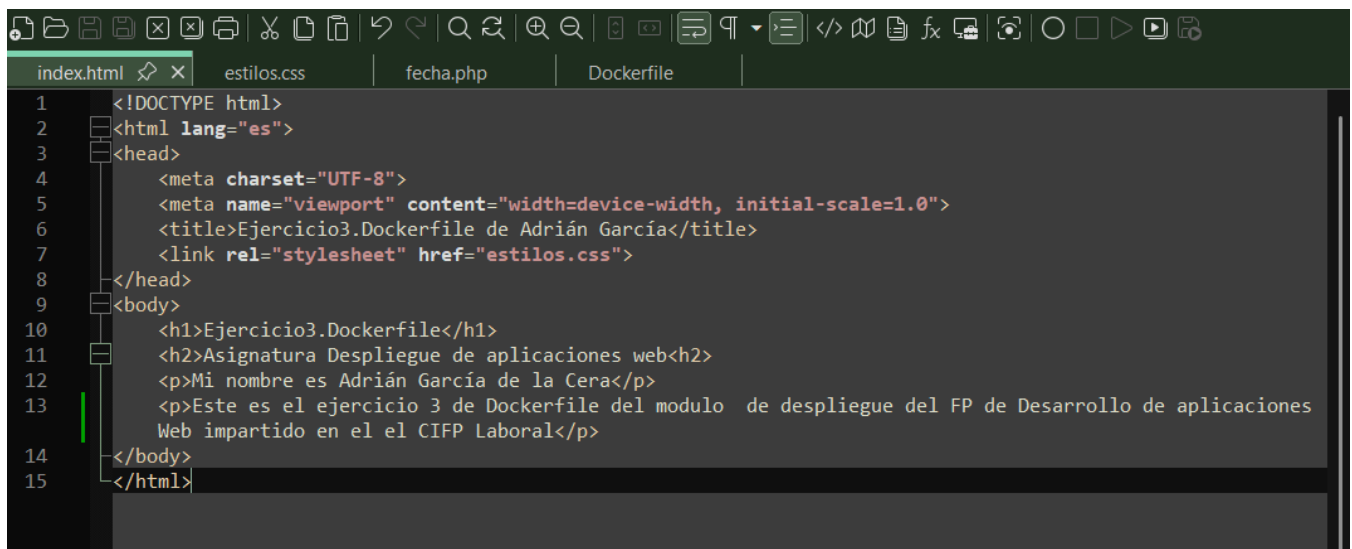
Cambia el puerto del contenedor, por ejemplo, `-p 1234:80`.

- el acceso al navegador con la página html y con el script `php`.

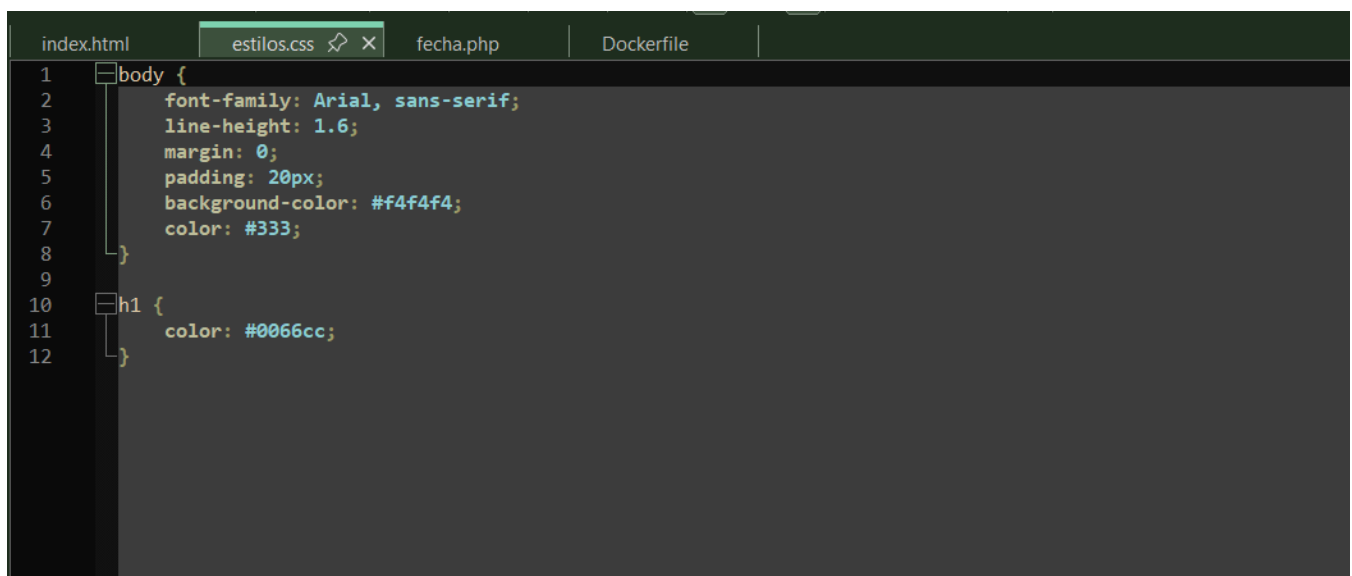
Desarrollo

Creación de archivos

Lo primero que vamos a realizar para la ejecución del ejercicio es crear los ficheros `index.html`, `estilos.css` y `fecha.php` en la carpeta web.



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Ejercicio3.Dockerfile de Adrián García</title>
7   <link rel="stylesheet" href="estilos.css">
8 </head>
9 <body>
10  <h1>Ejercicio3.Dockerfile</h1>
11  <h2>Asignatura Despliegue de aplicaciones web</h2>
12  <p>Mi nombre es Adrián García de la Cera</p>
13  <p>Este es el ejercicio 3 de Dockerfile del modulo de despliegue del FP de Desarrollo de aplicaciones
    Web impartido en el el CIEP Laboral</p>
14 </body>
15 </html>
```



```
1 body {
2   font-family: Arial, sans-serif;
3   line-height: 1.6;
4   margin: 0;
5   padding: 20px;
6   background-color: #f4f4f4;
7   color: #333;
8 }
9
10 h1 {
11   color: #0066cc;
12 }
```

```
index.html | estilos.css | fecha.php | Dockerfile
1 <?php
2 setlocale(LC_TIME, "es_ES.UTF-8");
3 $mes_actual = strftime("%B");
4 $fecha_actual = date("d/m/Y");
5 $hora_actual = date("H:i:s");
6 echo "<h1>Informacion</h1>";
7 echo "<p>Hoy es $fecha_actual</p>";
8 echo "<p>El mes es: <strong>$mes_actual</strong></p>";
9 echo "<p>Hora: $hora_actual</p>";
10 ?>
```

Una vez creados estos, creamos el archivo Dockerfile en el directorio raíz, el cual se compone de 3 elementos:

1. FROM: Indica la imagen base para construir la imagen Docker.
2. COPY: Copia el contenido de la carpeta web al directorio `/var/www/html/`.
3. RUN: Ejecuta los comandos `chown -R www-data:www-data /var/www/html` cambia el grupo y el propietario de los archivos al directorio que usa el servidor apache y `chmod -R 755 /var/www/html` modifica los permisos para que el propietario pueda leer, escribir y ejecutar y el grupo solo leer y ejecutar.

Usar la imagen base de PHP con Apache

FROM php:7.4-apache

Copia el contenido de la carpeta ./web/

COPY ./web/ /var/www/html/

Ejecuta dos comandos dentro del contenedor

RUN chown -R www-data:www-data /var/www/html && chmod -R 755 /var/www/html

```
index.html | estilos.css | fecha.php | Dockerfile
1 # Usar la imagen base de PHP con Apache
2 FROM php:7.4-apache
3
4 # Copia el contenido de la carpeta ./web/
5 COPY ./web/ /var/www/html/
6
7 # Ejecuta dos comandos dentro del contenedor
8 RUN chown -R www-data:www-data /var/www/html && chmod -R 755 /var/www/html
```

Creación de la imagen

Para crear la imagen usamos el siguiente comando:

```
docker build -t adriangarciaejerciciodockerfile/ejercicio3:v1 .
```

El comando se compone de las siguientes partes:

- `docker build`: Es el comando para crear la imagen.
- `-t`: Es la etiqueta que usamos para dar un nombre a la imagen.
- `adriangarciaejerciciodockerfile`: Es el nombre de la imagen, en este caso comienza con mi nombre y primer apellido (`adrian garcía`) al que se le ha agregado el nombre del ejercicio.
- `ejercicio3`: Se trata del nombre de la imagen.
- `v1`: Es la etiqueta de la imagen.
- `.`: Indica la ruta en la que están los archivos necesarios como es la actual se indica con ".".

```

/home/mobaxterm/Desktop/Ejercicio3-Dockerfile
2025-04-17 16:49:43 docker build -t adriangarciaejerciciodockerfile/ejercicio3:v1 .

[+] Building 20.7s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.1s
=> => transferring dockerfile: 296B                                0.0s
=> [internal] load metadata for docker.io/library/php:7.4-apache  2.5s
=> [auth] library/php:pull token for registry-1.docker.io         0.0s
=> [internal] load .dockerignore                                  0.1s
=> => transferring context: 2B                                       0.0s
=> [internal] load build context                                  0.2s
=> => transferring context: 3.84kB                                   0.0s
=> [1/3] FROM docker.io/library/php:7.4-apache@sha256:c9d7e608f73832673 12.9s
=> => resolve docker.io/library/php:7.4-apache@sha256:c9d7e608f738326734 0.1s
=> => sha256:ab590b48ea476386dd7b07c34de9eff7cf2103c4668 2.46kB / 2.46kB 0.4s
=> => sha256:05e465aaa99a358add4accedade8f39843089069f31fea0 892B / 892B 0.4s
=> => sha256:80692ae2d067c8358112c56490a2a97f69ef395fd8f7662 246B / 246B 0.3s
=> => sha256:d2c43c5efbc861f83ee6565c7102ca660d6f35e15 10.20MB / 10.20MB 1.1s
=> => sha256:66d98f73acb62e86c0c226f9eedcbc7eda305df0c1e171c 491B / 491B 0.3s
=> => sha256:d14eb2ed1e17ae00f5fcb44b0d562e2867c401c20 10.76MB / 10.76MB 1.3s
=> => sha256:fe42347c4ecfc90333acd9cad13912387eea39d13827a25 514B / 514B 0.3s
=> => sha256:9b233e420ac7bbca645bb82c213029762acf1742400c076 475B / 475B 0.2s
=> => sha256:25f85b498fd5bfc6cce951513219fe480850daba7 19.25MB / 19.25MB 2.6s
=> => sha256:fb5a4c8af82f00730b7427e47bda7f76cea2e2b9aea4217 270B / 270B 0.2s
=> => sha256:156740b07ef8a632f9f7bea4e57e4ee5541ade376 91.63MB / 91.63MB 6.4s
=> => sha256:c428f1a494230852524a2a5957cc5199c36c8b403305e0e 226B / 226B 0.2s
=> => sha256:a603fa5e3b4127f210503aaa6189abf6286ee5a73 31.41MB / 31.41MB 3.4s
=> => extracting sha256:a603fa5e3b4127f210503aaa6189abf6286ee5a73deeaab4 2.6s
=> => extracting sha256:c428f1a494230852524a2a5957cc5199c36c8b403305e0e8 0.1s
=> => extracting sha256:156740b07ef8a632f9f7bea4e57e4ee5541ade376adf9169 2.9s
=> => extracting sha256:fb5a4c8af82f00730b7427e47bda7f76cea2e2b9aea42175 0.0s
=> => extracting sha256:25f85b498fd5bfc6cce951513219fe480850daba71e6e997 0.4s
=> => extracting sha256:9b233e420ac7bbca645bb82c213029762acf1742400c0763 0.0s
=> => extracting sha256:fe42347c4ecfc90333acd9cad13912387eea39d13827a25c 0.0s
=> => extracting sha256:d14eb2ed1e17ae00f5fcb44b0d562e2867c401c20372829e 0.1s
=> => extracting sha256:66d98f73acb62e86c0c226f9eedcbc7eda305df0c1e171ca 0.0s
=> => extracting sha256:d2c43c5efbc861f83ee6565c7102ca660d6f35e158324fbb 0.4s
=> => extracting sha256:ab590b48ea476386dd7b07c34de9eff7cf2103c4668ade98 0.0s
=> => extracting sha256:80692ae2d067c8358112c56490a2a97f69ef395fd8f7662a 0.0s
=> => extracting sha256:05e465aaa99a358add4accedade8f39843089069f31fea02 0.0s
=> [2/3] COPY ./web/ /var/www/html/                                0.8s
=> [3/3] RUN chown -R www-data:www-data /var/www/html && chmod -R 755 /v 2.2s
=> => exporting to image                                             1.6s
=> => exporting layers                                              0.9s
=> => exporting manifest sha256:5d09b2fb002e9cec9a6831668938763ee417f0c4 0.0s
=> => exporting config sha256:8d2a61234f011e8191ef9175f03d8a354fea86cf84 0.0s
=> => exporting attestation manifest sha256:a2185b58fdf4258c343cb924134f 0.1s
=> => exporting manifest list sha256:969c42a5cef539c7c8250b662ee04ef079e 0.1s
=> => naming to docker.io/adriangarciaejerciciodockerfile/ejercicio3:v1 0.0s
=> => unpacking to docker.io/adriangarciaejerciciodockerfile/ejercicio3:v 0.3s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linu
/mhe7lo4xlv5zvoqow25kwikjj

/home/mobaxterm/Desktop/Ejercicio3-Dockerfile
2025-04-17 16:50:29

```

Comprobamos que ha creado la imagen:

```
docker images
```

```

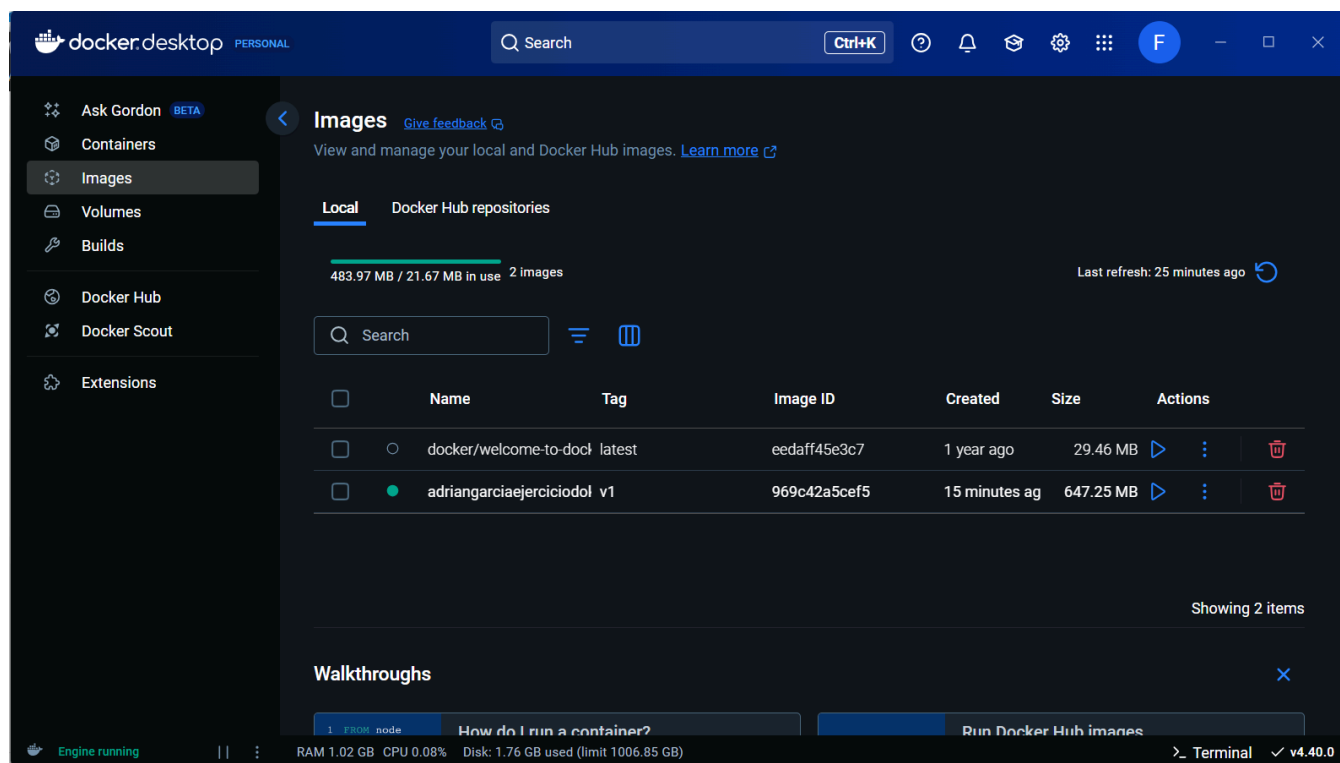
/home/mobaxterm/Desktop/Ejercicio3-Dockerfile
2025-04-17 16:50:29 docker images
REPOSITORY                                TAG      IMAGE ID      CREATED
SIZE
adriangarciaejerciciodockerfile/ejercicio3  v1       969c42a5cef5  7 minutes ago
647MB
docker/welcome-to-docker                  latest   eedaff45e3c7  17 months ago
29.5MB

```

Ejecutamos el contenedor en modo desatendido (-d) en el puerto 8080:

```
docker run -d -p 8080:80 adriangarciaejerciciodockerfile/ejercicio3:v1
```

Como podemos ver en Docker Desktop ya vemos la imagen creada:



Comprobamos si se accede a través del navegador:



Comprobaciones

Subida de imagenes

Para subir una imagen a Docker Hub lo primero es logarse con el comando `docker login`

```
2025-04-17 17:16:33 /home/mobaxterm/Desktop/Ejercicio3-Dockerfile docker login
Authenticating with existing credentials... [Username: fuejomusta]

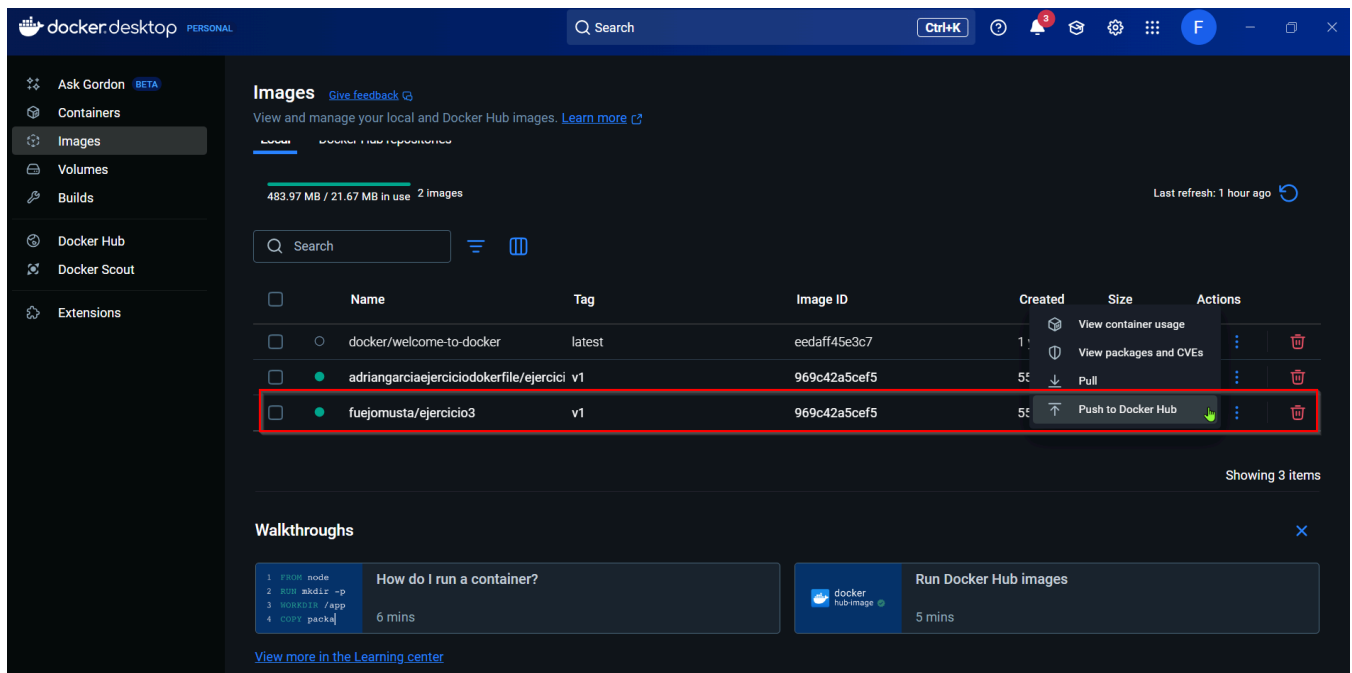
Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
```

En este caso al intentar subir la imagen a Docker hub da error por que el nombre indicado en la imagen no coincide con el usuario de Docker hub, por lo que procedemos a modificar la etiqueta de la imagen de `adriangarciaejerciciodockerfile/ejercicio3:v1` a `fuejomusta/ejercicio3:v1`, creando una nueva etiqueta para esa imagen y permitiendo subirla:

```
docker tag adriangarciaejerciciodockerfile/ejercicio3:v1 fuejomusta/ejercicio3:v1
```

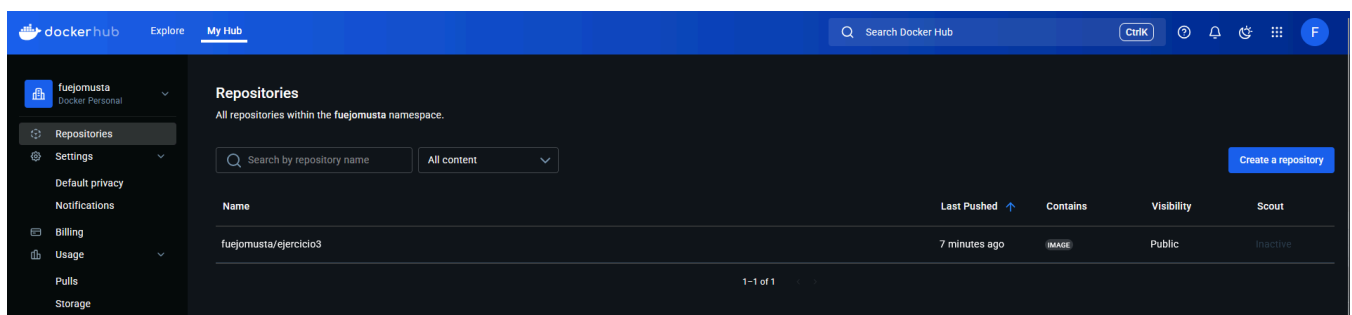
Una vez solucionado el problema de la etiqueta tenemos 2 opciones para subir la imagen a Docker Hub, a través de Docker Desktop, de manera muy sencilla, junto a la etiqueta aparecen 3 simbolos, un botón de play, un cubo de basura y 3 puntos verticales que al pulsar sobre ellos nos dejara indicar `Push to Docker Hub`, lo que iniciará la subida:



Otra manera es a través de la terminal con el código:

```
docker push fuejomusta/ejercicio3:v1
```

Una vez confirmado que ha subido la imagen, accedemos a nuestro repositorio en la web de Docker Hub y confirmamos que está la imagen creada.



Borrado y descarga de imagen

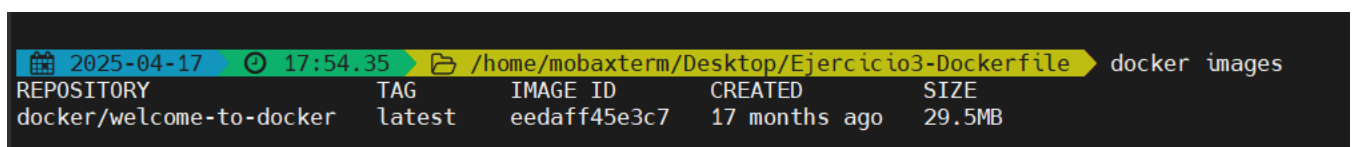
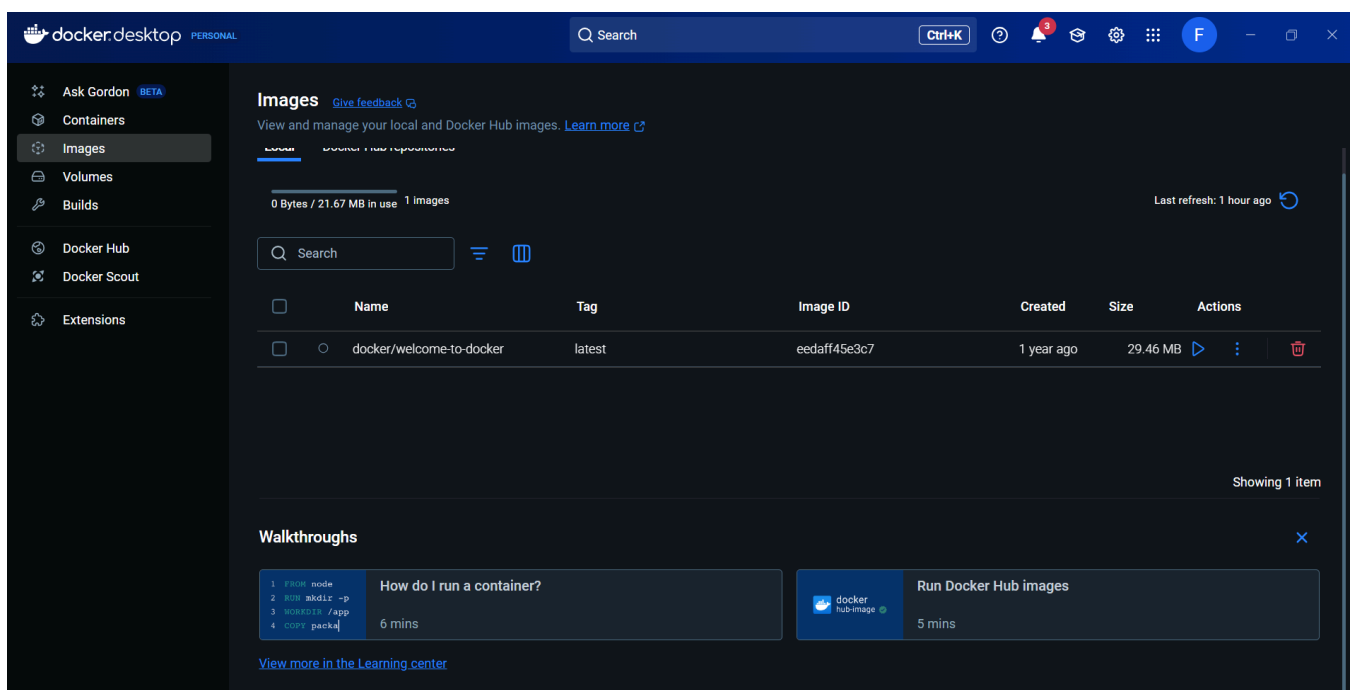
Una vez subida la imagen procedemos a borrar la imagen local bien con el botón con símbolo de basura de Docker Desktop o con el comando:

```
docker rmi fuejomusta/ejercicio3:v1
```

Es posible que nos dé error por estar trabajando el contenedor, en tal caso será necesario comprobar el contenedor (`docker ps -a`), parar la ejecución del contenedor (`docker stop nombre contenedor`) y luego borrarlo (`docker rm -f nombre contenedor`).

Una vez borrado comprobamos que ya no existe la imagen en Docker Desktop o con comando:

```
docker images
```



Para descargar la imagen de nuevo, al igual que para eliminarla existen 2 posibilidades, buscarla desde Docker Desktop y descargarla con el botón "Pull" o directamente con el comando:

```
docker pull fuejomusta/ejercicio3:v1
```

Y comprobamos de nuevo tenemos la imagen

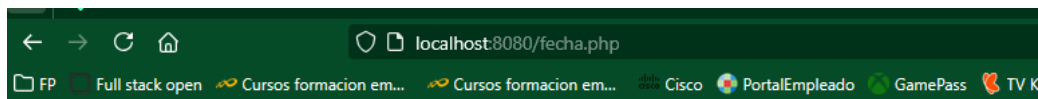
| <input type="checkbox"/> | Name | Tag | Image ID | Created | Size | Actions |
|--------------------------|--------------------------|--------|--------------|-------------|-----------|--|
| <input type="checkbox"/> | docker/welcome-to-docker | latest | eedaff45e3c7 | 1 year ago | 29.46 MB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | fuejomusta/ejercicio3 | v1 | 969c42a5cef5 | 2 hours ago | 647.25 MB | ▶ ⋮ 🗑️ |


```
2025-04-17 17:59:57 /home/mobaxterm/Desktop/Ejercicio3-Dockerfile docker pull fuejomusta/ejercicio3:v1
v1: Pulling from fuejomusta/ejercicio3
2968821aa7c5: Pull complete
845fbc9d5029: Pull complete
Digest: sha256:969c42a5cef539c7c8250b662ee04ef079e177758b1227ba0493ab2f0381db75
Status: Downloaded newer image for fuejomusta/ejercicio3:v1
docker.io/fuejomusta/ejercicio3:v1

2025-04-17 18:22:27 /home/mobaxterm/Desktop/Ejercicio3-Dockerfile docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
fuejomusta/ejercicio3  v1          969c42a5cef5   2 hours ago    647MB
docker/welcome-to-docker latest       eedaff45e3c7   17 months ago 29.5MB
```

Una vez descargada, es necesario iniciarla, para ello usamos el comando:

```
docker run -d -p 8080:80 fuejomusta/ejercicio3:v1
```



Informacion

Hoy es 17/04/2025

El mes es: **April**

Hora: 16:29:42