

Ejercicio 2 - Docker Compose

Documento que resuelve el ejercicio 2 de la Tarea 3- Despliegue de Aplicaciones Web, realizado por Epifanía Peralta López para Despliegue de Aplicaciones Web.

Ejercicio 2 - Docker Compose

Este ejercicio se puede resolver utilizando comandos, o Docker Desktop, o combinando ambos.

Enunciado: Explorar la imagen de la aplicación **FileBrowser** en este repositorio en GitHub: <https://hub.docker.com/r/hurlenko/filebrowser>.

*El usuario por defecto es **admin** con contraseña **admin**.*

Escribir un fichero **compose.yaml** para desplegarla. Los datos se pueden guardar utilizando volúmenes o utilizando bind-mount.

Entregar, al menos, las siguientes capturas de pantalla y los comandos y/o operaciones con Docker Desktop empleados para resolver el ejercicio:

- Captura de pantalla y documento donde se vea el fichero **docker-compose.yaml** que has creado.

Al explorar la imagen indicada, me informa de lo siguiente:

```
docker run -d --name filebrowser -p 80:8080 hurlenko/filebrowser
```

Para ejecutar como usuario actual y asignar ubicaciones de volúmenes personalizados, utilice:

```
docker run -d \  
  --name filebrowser \  
  --user $(id -u):$(id -g) \  
  -p 8080:8080 \  
  -v /DATA_DIR:/data \  
  -v /CONFIG_DIR:/config \  
  -e FB_BASEURL=/filebrowser \  
  hurlenko/filebrowser
```

docker-compose

El minimalismo `docker-compose.yml` podría verse así:

```
version: "3"  
  
services:  
  filebrowser:  
    image: hurlenko/filebrowser  
    user: "${UID}:${GID}"  
    ports:  
      - 443:8080  
    volumes:  
      - /DATA_DIR:/data  
      - /CONFIG_DIR:/config  
    environment:  
      - FB_BASEURL=/filebrowser  
    restart: always
```

Copiar

Parto de esa información para escribir el **compose.yaml**.

Descripción de los puertos: el `8080` como puerto predeterminado del explorador de archivos.

Variables de entorno admitidas:

En la imagen explica que FileBrowser usa una base de datos interna llamada **Bolt DB** que es un archivo único administrado automáticamente por la aplicación, de forma que no se necesita configurar un servidor de base de datos externo (como MySQL) para que funcione.

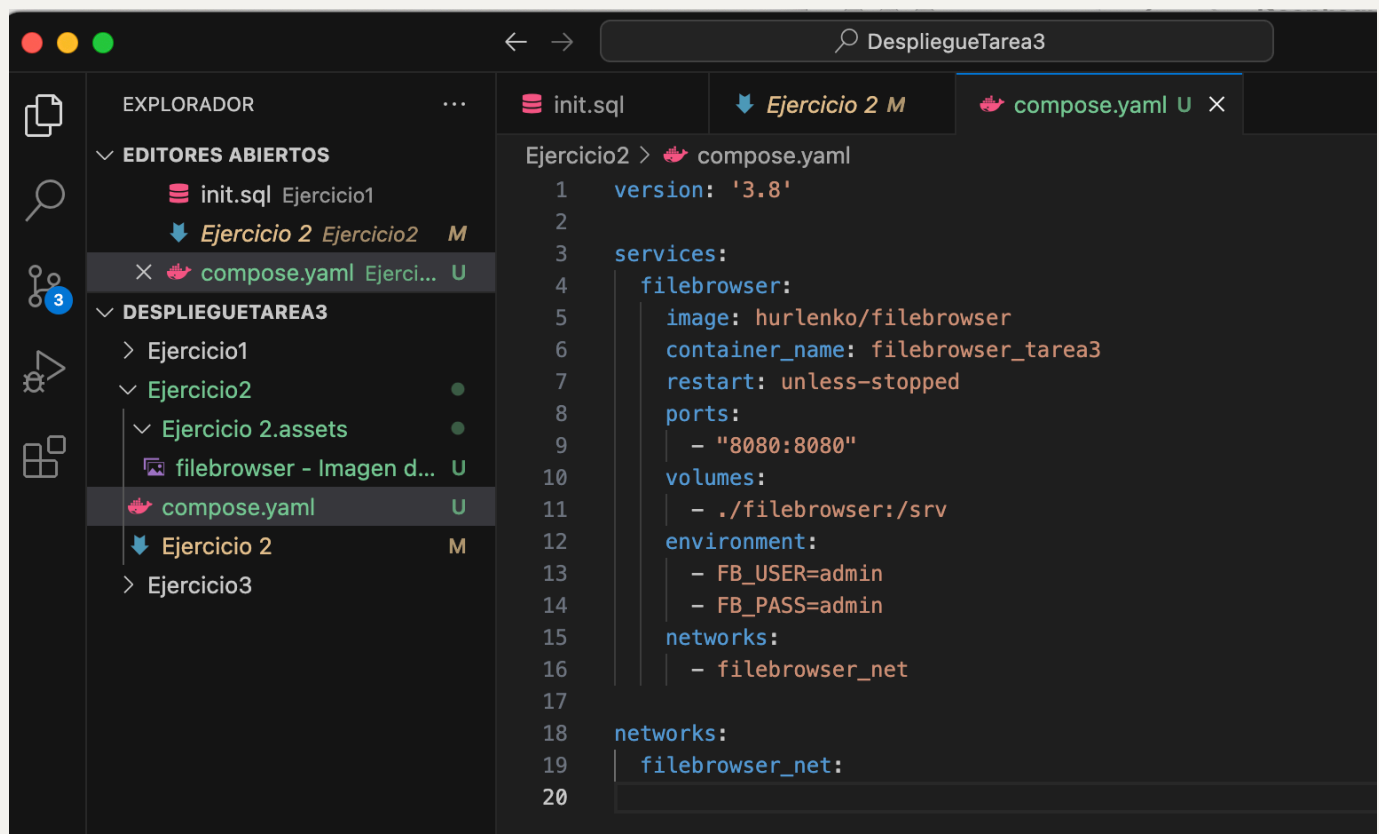
FileBrowser tiene una interfaz de línea de comandos (CLI) con la que se puede:

1. Crear la base de datos
2. Gestionar usuarios.
3. Configurar opciones sin necesidad de entrar en la interfaz web.

Las variables de entorno llevan el prefijo "FB_" seguido por el nombre de la opción en mayúsculas. Así, para la "Database", la variable de entorno sería: FB_DATABASE.

Volúmenes admitidos:

Voy a montar un bind-mount, ya que me permite ver en el sistema de archivos de mi macOS dónde se guardan los datos de FileBrowser. Para eso, creo un archivo en `Ejercicio 2/FileBrowser`.



The screenshot shows a code editor window titled "DespliegueTarea3" with three tabs: "init.sql", "Ejercicio 2 M", and "compose.yaml U". The "compose.yaml" tab is active, displaying the following content:

```
1 version: '3.8'
2
3 services:
4   filebrowser:
5     image: hurlenko/filebrowser
6     container_name: filebrowser_tarea3
7     restart: unless-stopped
8     ports:
9       - "8080:8080"
10    volumes:
11      - ./filebrowser:/srv
12    environment:
13      - FB_USER=admin
14      - FB_PASS=admin
15    networks:
16      - filebrowser_net
17
18 networks:
19   filebrowser_net:
20
```

The left sidebar shows a file explorer with the following structure:

- EXPLORADOR
 - EDITORES ABIERTOS
 - init.sql Ejercicio1
 - Ejercicio 2 Ejercicio2 M
 - compose.yaml Ejerci... U
 - DESPLIEGUETAREA3
 - Ejercicio1
 - Ejercicio2
 - Ejercicio 2.assets
 - filebrowser - Imagen d... U
 - compose.yaml U
 - Ejercicio 2 M
 - Ejercicio3