

## Sistemas de gestión empresarial

# Dockers. Introducción a redes y volúmenes

## Wordpress y MariaDB en la misma red

En esta práctica utilizamos imágenes ya existentes de Wordpress y de MariaDB para ponerlas en comunicación y en servicio.

### 1. Crear la red

Los contenedores que vamos a crear van a estar comunicados a través de la red **redwp** que creamos

```
docker network create redwp
```

### 2. Crear el contenedor de MariaDB

El siguiente paso es crear la base de datos conectándolo con la red. Se usa la imagen **mariadb:10.6**

```
docker run --name nuestromariadb --network redwp
-v /home/alumno/dockers/pr3_wp_mariadb_vols/mariadbdata:/var/lib/mysql
-e MARIADB_ROOT_PASSWORD=cefireroot -e MARIADB_USER=cefireuser -e MARIADB_PASSWORD=cefirepass
-e MARIADB_DATABASE=cefiredb
-d mariadb:10.6
```

Los parámetros que usa al crear el contenedor son:

- **nombre:** *nuestromariadb*
- **red:** *redwp*
- **volúmenes:** enlaza el directorio local */home/alumno/dockers/pr3\_wp\_mariadb\_vols/* al del contenedor, *mariadbdata:/var/lib/mysql*
- **variables de entorno:** las necesarias para establecer la conexión con la base de datos, *MARIADB\_ROOT\_PASSWORD*, *MARIADB\_USER*, *MARIADB\_PASSWORD* y *MARIADB\_DATABASE*. El usuario creado tiene privilegios de root.
- *-d* para ejecutar en segundo plano

### 3. Crear el contenedor de Wordpress

Ya existen imágenes con Apache, php y wordpress instalado por lo que simplemente hay que crear el contenedor a partir de esa imagen:

```
docker run --name nuestrowp --network redwp -p 8888:80 -d wordpress
```

Lo conectamos a la red **redwp** y publicamos en el puerto 8888 el predeterminado 80.

### 4. Configurar wordpress

Al acceder a **http://localhost:8888** vemos la interfaz inicial de wordpress en la que hay que configurar el acceso a la base de datos.

Rellenamos el formulario de acceso con los valores establecidos al crear el contenedor de mariaDB.

- Nombre de la base de datos: *cefiredb*
- Nombre de usuario: *cefireuser*
- Contraseña: *cefirepass*
- Servidor de la BD: *nuestromariadb*
- Prefijo de tabla: *wp\_*

## 5. Migrar la base de datos a otra versión

En este último paso vamos a cambiar el contenedor de la base de datos para actualizarlo a otra versión, de la 10.6 a la 10.7

El problema sería que al cambiar de contenedor, los datos se perdiesen. Sin embargo, al haber asociado el directorio que contiene los datos de la base de datos a un directorio local, al cambiar el contenedor los datos permanecen en nuestro directorio local.

Lo probamos. Previamente, paramos el contenedor de la BD para eliminarlo después.

```
docker stop nuestromariadb
```

```
docker rm nuestromariadb
```

Por último, volvemos a generar un nuevo contenedor que contiene mariaDB, con la versión 10.7, enlazando el volumen de los datos ya creados con la carpeta que contiene los datos en mariaDB. Las variables de entorno deben ser las mismas para que wordpress pueda seguir conectando.

```
docker run --name nuestromariadb --network redwp
-v /home/alumno/dockers/pr3_wp_mariadb_vols:/var/lib/mysql
-e MARIADB_ROOT_PASSWORD=cefireroot -e MARIADB_USER=cefireuser -e MARIADB_PASSWORD=cefirepass
-e MARIADB_DATABASE=cefiredb
-d mariadb:10.7
```