

Introducción a Docker

UD 05. Caso práctico 01

- Wordpress + MariaDB



Fons Social Europeu

L'FSE inverteix en el teu futur

Autor: Sergi García Barea

Actualizado Marzo 2023

Licencia



Reconocimiento – NoComercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

1. Introducción	3
2. Paso 1: Creando la red	3
3. Paso 2: creando contenedor MariaDB	3
4. Paso 3: Creando contenedor con Wordpress	4
5. Paso 4: Instalando Wordpress	4
6. Migrando contenedor MariaDB de 10.6 a 10.7	5
7. Bibliografía	5

UD05. CASO PRÁCTICO 01

1. INTRODUCCIÓN

En este caso práctico vamos a poner en marcha el popular CMS Wordpress. Para ello usaremos una red donde se conectarán dos contenedores: el primero utilizando Apache + PHP, junto con una versión instalada de Wordpress, mientras que el segundo contendrá un servidor de bases de datos MariaDB. Además, realizaremos un ejemplo de una migración de versión del contenedor MariaDB.

2. PASO 1: CREANDO LA RED

El primer paso a dar en este caso práctico, es crear la red que compartirán los contenedores que creamos, a la que llamaremos **“redwp”**. La podemos crear con el comando:

```
docker network create redwp
```

3. PASO 2: CREANDO CONTENEDOR MARIADB

El primer paso será crear un contenedor con el servidor de bases de datos MariaDB dentro de la red. Lo haremos utilizando la imagen https://hub.docker.com/_/mariadb con el siguiente comando:

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

Con esta orden estamos creando un contenedor de nombre **“nuestromariadb”**, dentro de la **“redwp”** y mapeando mediante “binding mount” el directorio local **“/home/sergi/mariadbddata”** con **“/var/lib/mysql”** (valor por defecto, pese a ser MariaDB) del contenedor, donde se guarda la información de las bases de datos de MariaDB.

Siguiendo la información de la imagen, establecemos diferentes variables de entorno que soporta este contenedor. En ellas indicamos:

- Contraseña de “root” de MariaDB.
- Usuario **“cefireuser”** y contraseña **“cefirepass”** de un usuario de MariaDB que creará con privilegios de **“root”**.
 - Creamos este usuario debido a que, por seguridad, MariaDB no soporta conexiones remotas como root. En la práctica, los permisos de MariaDB deben ser ajustados y es desaconsejable usar usuarios con permisos totales.
- Nombre de base de datos a crear.

Además, con propósito de realizar posteriormente una sustitución de contenedor, hemos usado la versión de la imagen **“mariadb:10.6”**.

Una vez listo el contenedor con MariaDB, procedemos a preparar el contenedor Wordpress.

4. PASO 3: CREANDO CONTENEDOR CON WORDPRESS

Crearemos el contenedor que contendrá Apache + PHP, así como Wordpress preinstalado. Nos basaremos en esta imagen: https://hub.docker.com/_/wordpress

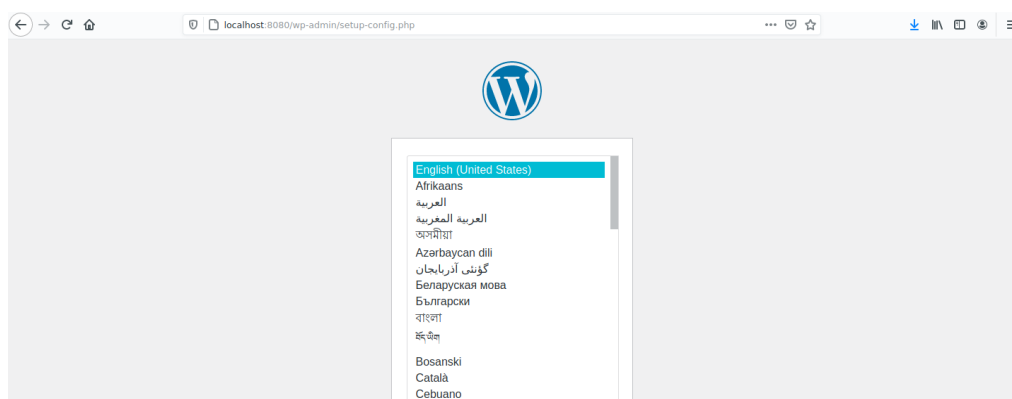
Podemos crearlo con:

```
docker run --name nuestrowp --network redwp -p 8080:80 -d wordpress
```

Con esta orden estamos creando un contenedor de nombre “**nuestrowp**”, dentro de la “**redwp**” y cargando el puerto 80 del contenedor en el puerto 8080 de la máquina anfitrión.

5. PASO 4: INSTALANDO WORDPRESS

Si en este momento accedemos a <http://localhost:8080> veremos algo similar a:



Ahora deberemos proseguir con la instalación e indicar dónde está el sistema de bases de datos. Usaremos como nombre de host, el nombre que le dimos al servidor MariaDB. Como usuario y nombre de base de datos utilizaremos las variables proporcionadas al crear el contenedor MariaDB. Quedará de forma similar a la siguiente imagen.

Nombre de la base de datos	<input type="text" value="cefiredb"/>	El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario	<input type="text" value="cefireuser"/>	El nombre de usuario de tu base de datos.
Contraseña	<input type="text" value="cefirepass"/>	La contraseña de tu base de datos.
Servidor de la base de datos	<input type="text" value="nuestromysql"/>	Deberías recibir esta información de tu proveedor de alojamiento web, si localhost no funciona.
Prefijo de tabla	<input type="text" value="wp_"/>	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

Una vez introducidos los datos, si todo es correcto, Wordpress nos informará de que ya podemos proceder a instalar y configurar nuestro sitio Wordpress.



Con esto ya hemos configurado Docker correctamente para poner en marcha este servicio. Ahora simplemente deberemos seguir los pasos de la instalación de Wordpress y ya tendremos listo nuestro sitio.

6. MIGRANDO CONTENEDOR MARIADB DE 10.6 A 10.7

Vamos a realizar un pequeño ejemplo donde migraremos el contenedor original, que tiene MariaDB versión 10.6 a un contenedor con MariaDB 10.7 (Elegimos estas versiones por simplicidad).

En primer lugar, deberemos parar y eliminar el contenedor “nuestromariadb”.

```
docker stop nuestromariadb
```

```
docker rm nuestromariadb
```

Una vez parado y eliminado, recordamos que no hemos perdido la información importante de MariaDB (configuración y bases de datos), ya que esta información ha quedado almacenada en la carpeta del anfitrión “**/home/sergi/mariadbdata**”. Al crear el nuevo contenedor, simplemente deberemos mapear esos datos. Podemos crearlo con:

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

Si todo ha ido bien, al acceder a <http://localhost:8080> veremos nuestro sitio Wordpress funcionando con normalidad.

7. BIBLIOGRAFÍA

[1] Docker Docs <https://docs.docker.com/>