

Introducción a Docker

UD 05. Caso práctico 01

- Wordpress + MySQL



Fons Social Europeu

L'FSE inverteix en el teu futur

Autor: Sergi García Barea

Actualizado Marzo 2021

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 MySQL	3
4. Paso 3: Creando contenedor con Wordpress	4
5. Paso 4: Instalando Wordpress	4
6. Migrando contenedor MySQL de 5.6 a 5.7	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 MySQL. Además, realizaremos un ejemplo de una migración de versión del contenedor MySQL.

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 MySQL

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

```
docker run --name nuestromysql --network redwp -v  
/home/sergi/mysqldata:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=cefireroot  
-e MYSQL_USER=cefireuser -e MYSQL_PASSWORD=cefirepass -e  
MYSQL_DATABASE=cefiredb -d mysql:5.6
```

Con esta orden estamos creando un contenedor de nombre **“nuestromysql”**, dentro de la **“redwp”** y mapeando mediante “binding mount” el directorio local **“/home/sergi/mysqldata”** con **“/var/lib/mysql”** del contenedor, donde se guarda la información de las bases de datos de MySQL.

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

- Contraseña de “root” de MySQL.
- Usuario **“cefireuser”** y contraseña **“cefirepass”** de un usuario de MySQL que creará con privilegios de **“root”**.
 - Creamos este usuario debido a que por seguridad, MySQL no soporta conexiones remotas como root. En la práctica, los permisos de MySQL 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 **“mysql:5.6”**.

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

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 MySQL DE 5.6 A 5.7

Vamos a realizar un pequeño ejemplo donde migraremos el contenedor original, que tiene MySQL versión 5.6 a un contenedor con MySQL 5.7 (Elegimos estas versiones por simplicidad, no migramos a 8 porque el proceso de migración es más complicado).

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

```
docker stop nuestromysql
```

```
docker rm nuestromysql
```

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

```
docker run --name nuestromysql --network redwp -v  
/home/sergi/mysqldata:/var/lib/mysql -d mysql:5.7
```

Destacar que en esta orden no hemos definido de nuevo las variables de entorno. El motivo de no haberlo hecho, es que se usaban para configurar MySQL, pero al tener ya los datos anteriores mapeados, este paso no es necesario.

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/>