

Enlazar contenedores (Wordpress y MariaDB)

En esta parte de la práctica crearemos un contenedor de una imagen de Wordpress (Gestor de contenido) y una imagen de MariaDB (Base de datos) y los enlazaremos para tener nuestro Wordpress bien configurado con una base de datos. Para enlazarlos los uniremos a la misma red.

Contenido de ayuda:

- Ver documentación en la página Web de Docker Hub para ver los parámetros necesarios para crear contenedor de base de datos.

1. Creamos una red o usamos una ya creada de las actividades anteriores.
2. Descargamos las imágenes de wordpress y mariadb.

```
$docker pull wordpress
$docker pull mariadb
```

3. Empezamos creando un contenedor de base de datos (MariaDB). Crearemos nuestra base de datos llamada (wordpress), el usuario de la base de datos (usuario_wordpress), contraseña para el usuario (password1). También le tenemos que dar una contraseña al usuario root.

```
$docker run -d --name Nombre_Contenedor --network Nombre_Red -e MARIADB_DATABASE=wordpress
-e MARIADB_USER=Usuario_Wordpress -e MARIADB_PASSWORD=Contraseña -e
MARIADB_ROOT_PASSWORD=Contraseña_Root mariadb
```

4. Con el contenedor de base de datos ya creado procedemos acceder al mismo para comprobar que se ha creado todo correctamente.

```
$docker exec -it mariadb1 bash
```

5. Nos conectamos a la base de datos de MariaDB para ello se utiliza el siguiente comando dentro del contenedor:

```
$mariadb -u Usuario_Wordpress -p
```

```
[iesroqueamagro@Chromecast ~]$ docker exec -it mariadb1 bash
root@7bc8d7113a57:/# mariadb -u usuario_wordpress -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.11.2-MariaDB-1:10.11.2+maria-ubu2204 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Usamos el siguiente comando para ver que efectivamente se ha creado nuestra base de datos para Wordpress:

```
$show databases;
```

Con lo anterior comprobamos que el usuario accede correctamente a la base de datos y que la base de Wordpress se ha creado correctamente.

Para salir de la base de datos usamos el siguiente comando:

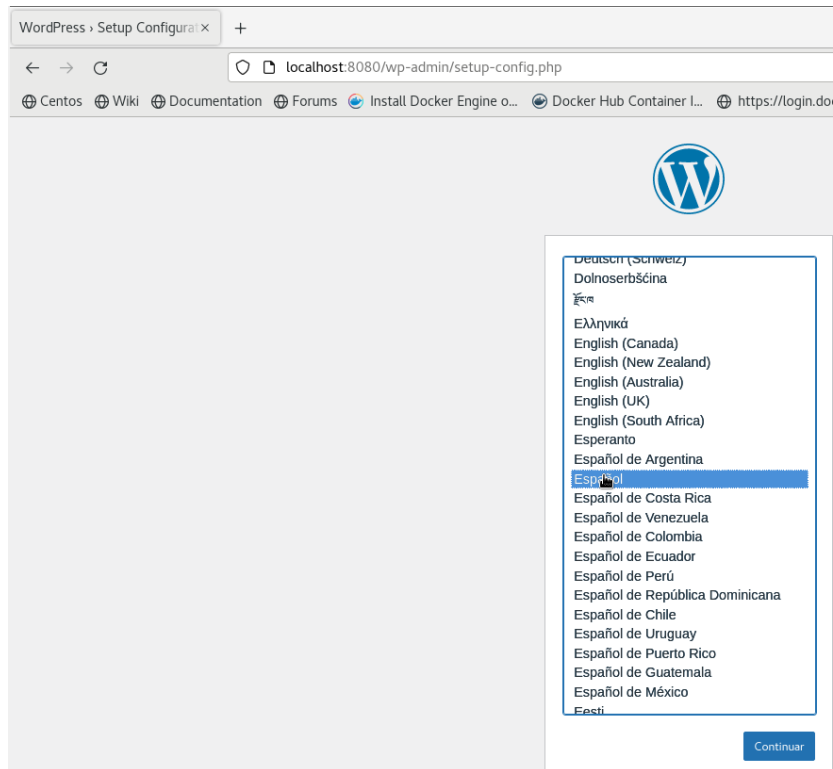
```
$quit
```

Llegados a este punto ya tenemos la base de datos preparada para tener almacenada toda la información que necesitará Wordpress para funcionar. Pasamos ahora a crear nuestro contenedor de wordpress.

6. Creamos el contenedor de wordpress, asociado a la misma red y elegimos el puerto 8080 de la máquina host para escuchar. Se podría pasar todos los datos necesarios por variables para que wordpress se conecte directamente a la base de datos, pero en este caso optamos por hacerlo de manera gráfica al conectarnos por el puerto 8080.

```
$ docker run --name Nombre_Wordpress --network Nombre_Red -d -p 8080:80  
wordpress
```

7. Comprobar e inspeccionar que los contenedores están en marcha y en la misma red.
8. Nos conectamos a un navegador por el puerto 8080 y nos debe aparecer el proceso de instalación de Wordpress:



The screenshot shows the WordPress database configuration page. At the top is the WordPress logo. Below it, a message states: 'A continuación tendrás que introducir los detalles de tu conexión con la base de datos. Si no estás seguro de ellos, contacta con tu proveedor de alojamiento.' The page contains five input fields with labels and descriptions:

Field Label	Value	Description
Nombre de la base de datos	wordpress	El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario	usuario_wordpress	El nombre de usuario de tu base de datos.
Contraseña	password1	La contraseña de tu base de datos.
Servidor de la base de datos	localhost	Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.
Prefijo de tabla	wp_	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

At the bottom left, there is an 'Enviar' button.

Hola

¡Este es el famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

Información necesaria

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio

Sitio Pruebas

Nombre de usuario

Mi Sitio

Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Contraseña

••••••••

Strong

Show

Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

Tu correo electrónico

prueba@prueba.com

Comprueba bien tu dirección de correo electrónico antes de continuar.

Visibilidad en los motores de búsqueda

☐ Pedir a los motores de búsqueda que no indexen este sitio

Depende de los motores de búsqueda atender esta petición o no.

Instalar WordPress



Ya tenemos nuestros dos contenedores enlazados y funcionando entre sí.

Volumenes en Docker

En Docker, un volumen es un mecanismo que permite a los contenedores de Docker almacenar y acceder a datos de forma persistente. Los volúmenes en Docker son utilizados para compartir y persistir datos entre contenedores o entre el sistema operativo anfitrión y los contenedores.

Estos volúmenes son directorios montados en el sistema de archivos del host o en otro contenedor, y están diseñados para ser independientes del ciclo de vida de los contenedores. Esto significa que los datos almacenados en un volumen persistirán incluso si se elimina el contenedor que lo creó. Los volúmenes también son utilizados para compartir datos entre contenedores, lo que permite la comunicación y colaboración entre distintos contenedores en un entorno de Docker.

Se utilizan para manejar datos que deben ser persistentes, como bases de datos, archivos de configuración, registros, y otros datos que necesitan sobrevivir al ciclo de vida de los contenedores individuales. Los volúmenes son una herramienta poderosa en Docker para separar la lógica de la aplicación de los datos, permitiendo una mayor flexibilidad y portabilidad en el despliegue de aplicaciones en contenedores Docker.

Comando de ayuda:

```
$ docker volume --help
```

Antes de crear nuestros propios volúmenes pasaremos a usar una imagen que al crear un contenedor de la misma crea un volumen por defecto. La imagen utilizada es Owncloud, servicio que nos permite disponer de nuestro almacenamiento privado en la nube. Este servicio escucha por el puerto 8080.

Recuerda inspeccionar las imágenes para conocer información importante de dicha imagen.

1. Descargamos Owncloud/Server y comprobamos que la imagen se ha descargado correctamente.
2. Vamos ahora a comprobar los volúmenes que tenemos en nuestro Host con el siguiente comando:

```
$ docker volume ls
```

- a. Para eliminar los volúmenes que no se están utilizando por ningún contenedor se usa el siguiente comando:

```
$ docker volume prune
```

- b. Para eliminarlos manualmente, se puede usar el siguiente comando:

```
$ docker volume rm ID_VOLUME
```

3. Listamos los volúmenes que tenemos actualmente.
4. Creamos un contenedor de owncloud (puerto por el que escucha el servicio, 8080).
5. Listamos los volúmenes y debería aparecer un volumen nuevo.
 - a. Los volúmenes Docker por defecto se almacenan en /var/lib/docker/volumes. Accedemos a ese directorio con usuario administrador y vemos que tiene que aparecer una carpeta con el mismo nombre que el volumen creado por defecto por el contenedor de owncloud creado.
6. Ahora accedemos a nuestro contenedor desde un navegador y nos logueamos con el usuario admin con contraseña admin y creamos un fichero.
 - a. Si ahora accedemos a la siguiente ruta en nuestro equipo Host podemos ver el archivo creado.

```
$ cd /var/lib/docker/volumes/Nombre_Volumen_Owncloud/_data/files/admin/files
```

Vemos que los archivos se localizan en el servidor Host y son accedidos desde el contenedor creado.

Podemos cerrar el contenedor creado para esta actividad.

En los siguientes pasos crearemos nuestros propios volúmenes.

Con la opción “-v Directorio” al crear un contenedor podemos crear un volumen y asignarlo al contenedor.

7. Creamos un contenedor de Ubuntu y le asignamos un volumen:

```
$ docker run --name ubuntu1 -d -it -v /datos ubuntu
```

8. Accedemos al contenedor y creamos un fichero en “/datos”:

9. Ahora si accedemos al volumen que se nos creó al iniciar el contenedor podemos ver que está el fichero recién creado dentro del contenedor:

```
$ cd /var/lib/docker/volumes/Nombre_Volumen /_data/
```

10. Si ahora creamos un fichero en la dirección del punto anterior podemos ver dicho fichero en el contenedor: