Introducción a Docker

UD 06. Caso práctico 01 - Wordpress + MySQL

Autor: Sergi García Barea

Actualizado Abril 2021

Licencia

Nomenclatura

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

📖 Importante

❕ Atención

💬 Interesante

UD06. CASO PRÁCTICO 01

1. INTRODUCCIÓN

En este caso práctico vamos a poner en marcha el popular CMS Wordpress. Para ello usaremos un fichero "***docker-compose.yml***" comentado que nos pondrá en marcha 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. Este ejemplo es similar al propuesto como ejemplo durante el contenido de la unidad.

2. FICHERO "DOCKER-COMPOSE.YML" DEL CASO PRÁCTICO

El contenido del fichero "***docker-compose.yml***" que incluimos comentado, es el siguiente:

```
#Versión del fichero docker-compose 3.9. No obligatorio desde la versión de docker-compose 1.27.0
version: "3.9"

#Indicamos los servicios a lanzar
services:
  #Plantill del servicio "db"
  db:
        #Se basa en la imagen "mysql", version 5.7
        image: mysql:5.7
        #Mapea en el volumen "db_data" el directorio "/var/lib/mysql", lo que da persistencia al contenido
de
        #Wordpress almacenado en la base de datos
        volumes:
        - db_data:/var/lib/mysql
        #Indica que siempre que el servicio finalice, se reiniciará
        restart: always
        #Define un conjunto de variables de entorno para estos contenedores,
        #indicando password de root de mysql, nombre de base de datos,
        # usuario con permisos root (necesario para conexiones remotas) y password de ese usuario
        environment:
        MYSQL_ROOT_PASSWORD: somewordpress
        MYSQL_DATABASE: wordpress
        MYSQL_USER: wordpress
        MYSQL_PASSWORD: wordpress
  #Plantilla del servicio "wordpress"
  wordpress:
        #Indicamos que para lanzar este servicio, debe estar en marcha "db"
        depends_on:
        - db
        #Indicamos que basa en la imagen "wordpress", version "latest"
        image: wordpress:latest
        #Indicamos que el puerto 80 del contenedor se mapea con el puerto 8000 del anfitrion
        ports:
        - "8000:80"
        #Indica que siempre que el servicio finalice, se reiniciará
        restart: always
        #Definimos "variables de entorno". Definimos donde conectarnos a la base de datos,
        #usuario de la base de datos, password de la base de datos y nombre de la base de datos
        environment:
        WORDPRESS_DB_HOST: db:3306
        WORDPRESS_DB_USER: wordpress
        WORDPRESS_DB_PASSWORD: wordpress
        WORDPRESS_DB_NAME: wordpress
        #Indicamos los volúmenes creados y compartidos a lo largo del fichero docker-compose.yml
volumes:
  db_data:
```

Este fichero se ha explicado en detalle de forma didáctica durante la unidad, pero aquí repasamos las claves más importantes:

- Indicamos que los servidores de bases de datos ("***db***")
  - Enlacen su información a un volumen, dotándolo de persistencia.
  - Definan una variables de entorno definiendo usuarios, contraseñas y bases de datos a usar.
- Indicamos que los servidores con Apache + PHP + Wordpress ("***wordpress***"):
  - Para iniciarse, debe iniciarse antes un servicio "***db***".
  - Establece variables de entorno definiendo valores para la conexión de base de datos de Wordpress.
  - Enlaza puerto 80 del contenedor a puerto 8000 del anfitrión.

3. PASO 1: PONIENDO EN MARCHA EL SISTEMA

Para poner en marcha este sistema, simplemente nos situamos en el directorio donde tengamos el fichero "***docker-compose.yml***" de este caso práctico y escribimos:

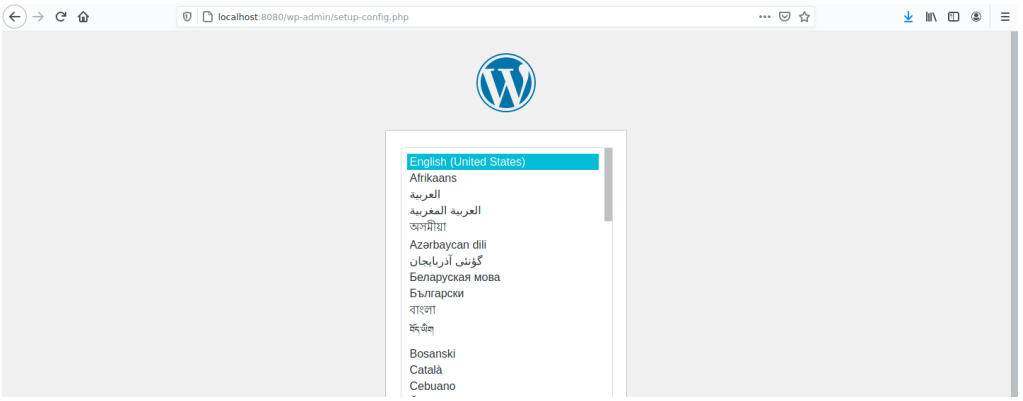
```
docker-compose up -d
```

La opción "***-d***" indica que "***Docker Compose***" se ejecute en segundo plano.

La opción "***up***", descarga y construye imágenes (si no estaban ya). Tras ello lanza los contenedores asociados, siguiendo orden de dependencia.

```
sergi@ubuntu:~/Desktop/docker-composeUD06/CasoPractico1-Wordpress$ docker-compose up -d
Creating network "casopractico1-wordpress_default" with the default driver
Creating volume "casopractico1-wordpress_db_data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
f7ec5a41d630: Pull complete
9444bb562699: Pull complete
6a4207b96940: Pull complete
181cefd361ce: Pull complete
8a2090759d8a: Pull complete
15f235e0d7ee: Pull complete
d870539cd9db: Pull complete
7310c448ab4f: Pull complete
4a72aac2e800: Pull complete
b1ab932f17c4: Pull complete
1a985de740ee: Pull complete
Digest: sha256:e42a18d0bd0aa746a734a49cbbcc079ccdf6681c474a238d38e79dc0884e0ecc
Status: Downloaded newer image for mysql:5.7
Pulling wordpress (wordpress:latest)...
latest: Pulling from library/wordpress
75646c2fb410: Already exists
854fb08fe050: Pull complete
d099f6707d86: Pull complete
038e5b090752: Pull complete
56671971dcc6: Pull complete
6da3e75ee2ca: Pull complete
88fd46807e1d: Pull complete
f33a657f956e: Pull complete
249520ff71af: Pull complete
4213c3e42364: Pull complete
4915809df15f: Pull complete
2faa4b167ab4: Pull complete
78435232ad8f: Pull complete
662883b7bb15: Pull complete
bf62eea5448f: Pull complete
92a1afd88c46: Pull complete
e0f9cda83bc3: Pull complete
a01ecf9f410a: Pull complete
608ccbf945cb: Pull complete
fd4a2a57c3c7: Pull complete
0ca288048117: Pull complete
Digest: sha256:e0cb92a0ff71ca1421d6cb1a2823a6e4fef5ad813fe7a6bd7f50700d21a290ed
Status: Downloaded newer image for wordpress:latest
Creating casopractico1-wordpress_db_1 ... done
Creating casopractico1-wordpress_wordpress_1 ... done
sergi@ubuntu:~/Desktop/docker-composeUD06/CasoPractico1-Wordpress$
```

Tras ello, podemos probar que todo es correcto accediendo a http://localhost:8000 donde veremos algo similar a:



Y simplemente podremos proseguir a poner en marcha nuestro sitio Wordpress.

4. PASO 2: PARANDO EL SISTEMA

Parar el sistema es tan sencillo como utilizar el comando

```
docker-compose down
```

```
sergi@ubuntu:~/Desktop/docker-composeUD06/CasoPractico1-Wordpress$ docker-compose down
Stopping casopractico1-wordpress_wordpress_1 ... done
Stopping casopractico1-wordpress_db_1        ... done
Removing casopractico1-wordpress_wordpress_1 ... done
Removing casopractico1-wordpress_db_1        ... done
Removing network casopractico1-wordpress_default
sergi@ubuntu:~/Desktop/docker-composeUD06/CasoPractico1-Wordpress$
```

Con ello se pararán y eliminarán los contenedores. No se eliminarán ni las imágenes ni los volúmenes (el sistema Wordpress mantiene la persistencia, al tener mapeados la información de la base de datos a un volumen).

5. PASO 3: RE-LANZANDO EL SISTEMA

Relanzar el sistema es tan sencillo como volver a lanzar el comando

```
docker-compose up -d
```

```
sergi@ubuntu:~/Desktop/docker-composeUD06/CasoPractico1-Wordpress$ docker-compose up -d
Creating network "casopractico1-wordpress_default" with the default driver
Creating casopractico1-wordpress_db_1 ... done
Creating casopractico1-wordpress_wordpress_1 ... done
sergi@ubuntu:~/Desktop/docker-composeUD06/CasoPractico1-Wordpress$
```

Observamos que el sistema aprovecha las imágenes ya creadas para acelerar el proceso de puesta en marcha, simplemente creando y lanzando los contenedores.

6. BIBLIOGRAFÍA

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

[2] Docker Compose Docs https://docs.docker.com/compose/