

Introducció a Docker

Part 06. Cas pràctic 01 - Wordpress amb Docker Compose



Autor: Sergi García Barea

Actualitzat setembre 2021

Llicència



Reconeixement - No comercial - CompartirIgual (BY-NC-SA): No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original.

Nomenclatura

Al llarg d'aquest tema s'utilitzaran diferents símbols per distingir elements importants dins del contingut. Aquests símbols són:

Important

Atenció

Interessant

Introducció	3
Fitxer "docker-compose.yml" del cas pràctic	3
Pas 1: Posant en marxa el sistema	4
Pas 2: Parant el sistema	5
Pas 3: Re-llançant el sistema	5
Bibliografia	5

PART 06. CAS PRÀCTIC 01

1. INTRODUCCIÓ

En aquest cas pràctic posarem en marxa el popular CMS Wordpress. Per a això farem servir un fitxer **"docker-compose.yml"** comentat que ens posarà en marxa dos contenidors: el primer utilitzant **"Apache+PHP"**, juntament amb una versió instal·lada de **Wordpress**, mentre que el segon contindrà un servidor de base de dades MySQL. Aquest exemple és similar al proposat com a exemple durant el contingut de la unitat anterior.

2. FITXER "DOCKER-COMPOSE.YML" DEL CAS PRÀCTIC

El contingut d'el fitxer **"docker-compose.yml"** que incloem comentat, és el següent:

```
#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:
  #Plantilla 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 anfitrión
    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:
```

Aquest fitxer s'ha explicat en detall de manera didàctica durant la unitat, però ací repassem les claus més importants:

- Indiquem que els servidors de bases de dades ("db")
 - Enllacen la seua informació a un volum, dotant-lo de persistència.
 - Defineixen unes variables d'entorn definint usuaris, contrasenyes i bases de dades a utilitzar.
- Indiquem que els servidors amb Apache + PHP + Wordpress ("wordpress"):
 - Per iniciar-se, s'ha d'iniciar abans un servei "db".
 - Estableix variables d'entorn definint valors per a la connexió de base de dades de Wordpress.
 - Enllaça port 80 del contenidor a port 8000 de l'amfitrió.

3. PAS 1: POSANT EN MARXA EL SISTEMA

Per posar en marxa aquest sistema, simplement ens situem en el directori on tinguem el fitxer "**docker-compose.yml**" d'aquest cas pràctic i escrivim:

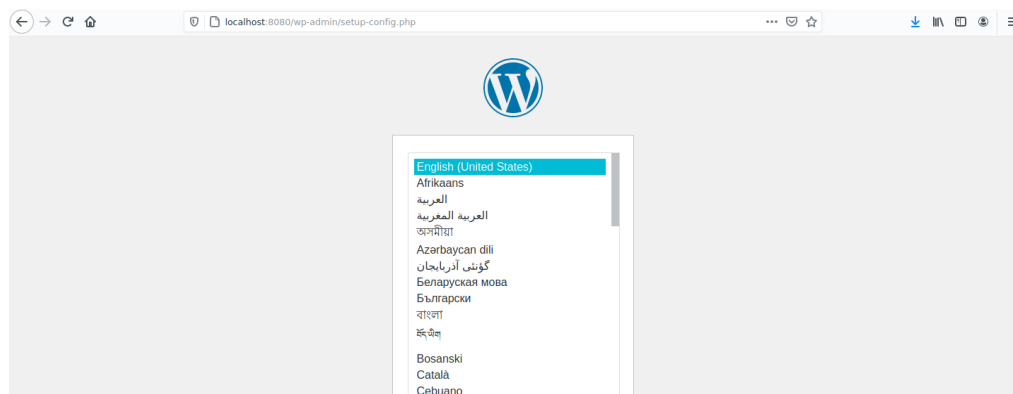
```
docker-compose up -d
```

L'opció "**-d**" indica que "**Docker Compose**" s'executa en segon pla.

L'opció "**up**", descàrrega i construeix imatges (si no estaven ja). Després d'això llança els contenidors associats, seguint ordre de dependència.

```
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
608ccb9f945cb: 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$
```

Després d'això, podem provar que tot és correcte accedint a <http://localhost:8000> on veurem una cosa semblant a:



I simplement podem prosseguir a posar en marxa el nostre lloc Wordpress.

4. PAS 2: PARANT EL SISTEMA

Aturar el sistema és tan senzill com utilitzar la comanda

```
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$
```

Amb això es pararan i s'eliminaran els contenidors. No s'eliminaran ni les imatges ni els volums (el sistema Wordpress manté la persistència, en tindre mapejats la informació de la base de dades a un volum).

5. PAS 3: RE-LLANÇANT EL SISTEMA

Relançar el sistema és tan senzill com tornar a llançar la comanda

```
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$
```

Observem que el sistema aprofita les imatges ja creades per accelerar el procés de posada en marxa, simplement creant i llançant els contenidors.

6. BIBLIOGRAFIA

- [1] Docker Docs <https://docs.docker.com/>
- [2] Docker Compose Docs <https://docs.docker.com/compose/>