

Docker compose

Realizado por Pablo Sánchez Hidalgo



Usar Docker-Compose.....	3
Preguntas.....	5
Respuestas.....	5

Lo primero que deberemos hacer es escribir un archivo **yml**, llamado **docker-compose**, el cual tendrá la siguiente información:

```
version: '2.2'

services:
  bbdd:
    image: mysql:5.7
    environment:
      - MYSQL_ROOT_PASSWORD=root
      - MYSQL_DATABASE=wordpress
      - MYSQL_USER=wordpress
      - MYSQL_PASSWORD=wordpress
    volumes:
      - db:/var/lib/mysql

  wordpress:
    image: wordpress:latest
    depends_on:
      - bbdd
    volumes:
      - wordpress:/var/www/html
    environment:
      - WORDPRESS_DB_USER=wordpress
      - WORDPRESS_DB_PASSWORD=wordpress
      - WORDPRESS_DB_HOST=bbdd
      - WORDPRESS_DB_NAME=wordpress
    ports:
      - 8080:80

volumes:
  wordpress:
  db:
```

Como podemos ver, tenemos el archivo.

```
root@usuario-VirtualBox:~/docker# ls
docker-compose.yaml
```

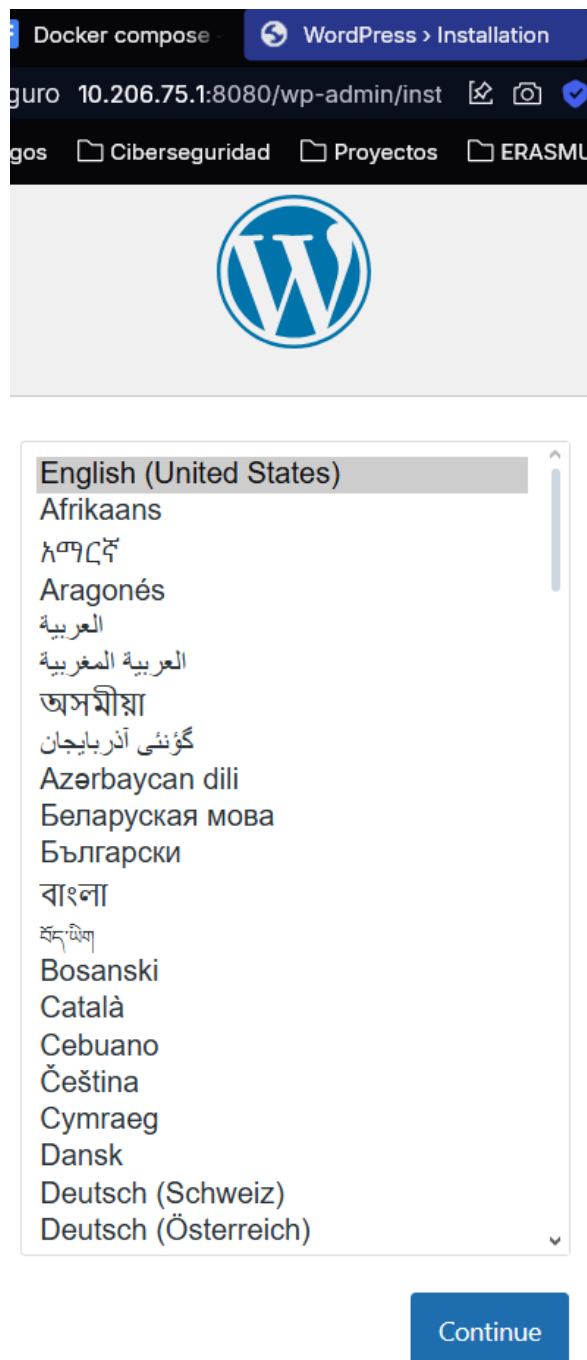
Al ejecutar el comando **docker-compose up -d**, vamos a levantar los servicios que tenga dentro del archivo, no sin antes que **docker** haga una comprobación lingüística de su correcto funcionamiento.

```
root@usuario-VirtualBox:~/docker# docker-compose up -d
Creating network "docker_default" with the default driver
Creating volume "docker_wordpress" with default driver
Creating volume "docker_db" with default driver
Pulling bbbd (mysql:5.7)...
5.7: Pulling from library/mysql
20e4dcac4c69: Downloading [=====]          22.4MB/50.5MB
1c56c3d4ce74: Pulling fs layer
1c56c3d4ce74: Downloading [=====]          1c56c3d4ce74: Downloading [=====]1c56c3d4ce74: Download complete
c3d4ce74: Download complete
e9f03alc24ce: Downloading [=====]          20e4dcac4c69: Downloading [=====]          ]
18.85MB/50.5MBDownloading [=====]          e9f03alc24ce: Download complete        68c3898c
ae71319cb779: Downloading [=====]          ] 9.439MB/25.53MB
90986bb8de6e: Download complete
43d05e938198: Downloading [====]          ae71319cb779: Downloading [=====]          ]
62MB/56.29MB          ] 3.783MB/56.29MB
```

Tras esto, podemos ver cómo se han creado dos contenedores.

```
Status: Downloaded newer image for wordpress:latest
Creating docker_bbdd_1 ... done
Creating docker_wordpress_1 ... done
```

Accediendo a la **ip** del equipo **con el puerto 8080**, podemos ver como tenemos abierto el archivo



Como podemos observar también dentro del equipo, con un **docker ps**, podemos verificar que se ha levantado correctamente el servicio.

```
root@usuario-VirtualBox:~/docker# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
08fcf8e68061   wordpress:latest "docker-entrypoint.s-"   3 minutes ago Up 3 minutes  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  docker_wordpress_1
9fc43e50f83f   mysql:5.7     "docker-entrypoint.s-"   3 minutes ago Up 3 minutes  3306/tcp, 33060/tcp                docker_bbdd_1
```

Preguntas

Indica cuántos servicios se levantan.

¿Qué imágenes se descargan?

¿Qué servicio se inicia antes?. Indica la razón.

¿En qué puerto local se encuentra escuchando el servidor web? Para acceder a la web, ¿qué dirección y puerto se han de poner en el navegador?

¿Cuántos volúmenes se crean? ¿En qué directorio se encuentran? ¿Qué sucede con los volúmenes si se borran los contenedores?

Respuestas

Se han levantado **dos servidores, wordpress y mysql**, con las correspondientes imágenes necesarias para que inicien, en este caso las últimas.

El primer servicio en iniciarse es **mysql**, ya que depende de wordpress depende de este servicio para iniciarse.

El servidor usa el puerto **8080**, para **salida al exterior del equipo**, sin embargo de forma interna tenemos el puerto **80**, dentro del contenedor **wordpress** abierto para la comunicación de información.

Los **volúmenes** que hemos creado han sido uno por cada contenedor activado, estos contenedores se usan para almacenar información en caso de pérdida o reinicio de los contenedores. Estás ubicado en el directorio raíz.