

# Docker Compose

## ¿Qué es?

Hasta ahora hemos trabajado con contenedores compuestos por una sola imagen. Pero ¿qué ocurre si queremos tener dos contenedores que compartan sus recursos?. Por ejemplo, en uno instalamos una base de datos y en otro corremos una aplicación web que use dicha base de datos.

En este caso, usaremos `docker compose` que es una herramienta para crear y ejecutar aplicaciones multi-contenedores.

Usar Compose es básicamente un proceso de tres pasos:

1. Definir el entorno de su aplicación con un Dockerfile para que pueda reproducirse en cualquier lugar.
2. Definir los servicios que componen la aplicación en `docker-compose.yml` para que puedan ejecutarse juntos en un entorno aislado.
3. Ejecutar `docker-compose up` y Compose inicia y ejecuta toda la aplicación.

Un `docker-compose.yml` se ve así:

```
version: "3.9"  # optional since v1.27.0
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./code
      - logvolume01:/var/log
    links:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

---

En este archivo vemos que se tenemos un servicio denominado **web** que se va a construir (**build: .**) a partir de un fichero Dockerfile en el mismo directorio y que este servicio depende de otro llamado **redis** (link [redis](#)). De esta forma se le está indicando que un contenedor puede acceder a otro (recordad que los contenedores son estancos).

## Práctica Wordpress

No hay nada mejor para entenderlo que realizar una práctica.

En este caso vamos a instalar Wordpress en su propio contenedor. Sigue estos pasos:

1. Crea un directorio de proyecto vacío.  
Este directorio es el contexto de la imagen de la aplicación. El directorio solo debe contener recursos para construir esa imagen. Este directorio de proyecto contiene un archivo **docker-compose.yml** que es un buen proyecto de inicio de wordpress.
2. Cambia al directorio del proyecto. Por ejemplo, si nombraste el directorio **my\_wordpress**:

```
cd my_wordpress
```

3. Crea un archivo **docker-compose.yml** que inicie tu blog de WordPress y una instancia de MySQL separada con un montaje de volumen para la persistencia de datos:

```
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
```

---

```
image: wordpress:latest
ports:
  - "8000:80"
restart: always
environment:
  WORDPRESS_DB_HOST: db:3306
  WORDPRESS_DB_USER: wordpress
  WORDPRESS_DB_PASSWORD: wordpress
  WORDPRESS_DB_NAME: wordpress
volumes:
  db_data: {}
```

## Construir el proyecto

Ahora, ejecuta `docker-compose up -d` en dicho directorio. Esto ejecuta `docker-compose up` en modo separado, extrae las imágenes de Docker necesarias e inicia los contenedores de wordpress y base de datos, como se muestra en el siguiente ejemplo.

```
$ docker-compose up -d
Creating network "my_wordpress_default" with the default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
efd26ecc9548: Pull complete
a3ed95caeb02: Pull complete
...
Digest: sha256:34a0aca88e85f2efa5edff1cea77cf5d3147ad93545dbec99cfe705b03c520de
Status: Downloaded newer image for mysql:5.7
Pulling wordpress (wordpress:latest)...
latest: Pulling from library/wordpress
efd26ecc9548: Already exists
a3ed95caeb02: Pull complete
589a9d9a7c64: Pull complete
...
Digest: sha256:ed28506ae44d5def89075fd5c01456610cd6c64006addfe5210b8c675881aff6
Status: Downloaded newer image for wordpress:latest
Creating my_wordpress_db_1
Creating my_wordpress_wordpress_1
```

En este punto, WordPress debería estar ejecutándose en el puerto 8000 de su Docker host `http://localhost:8000`, y puedes completar la “famosa instalación de cinco minutos” como administrador de WordPress.

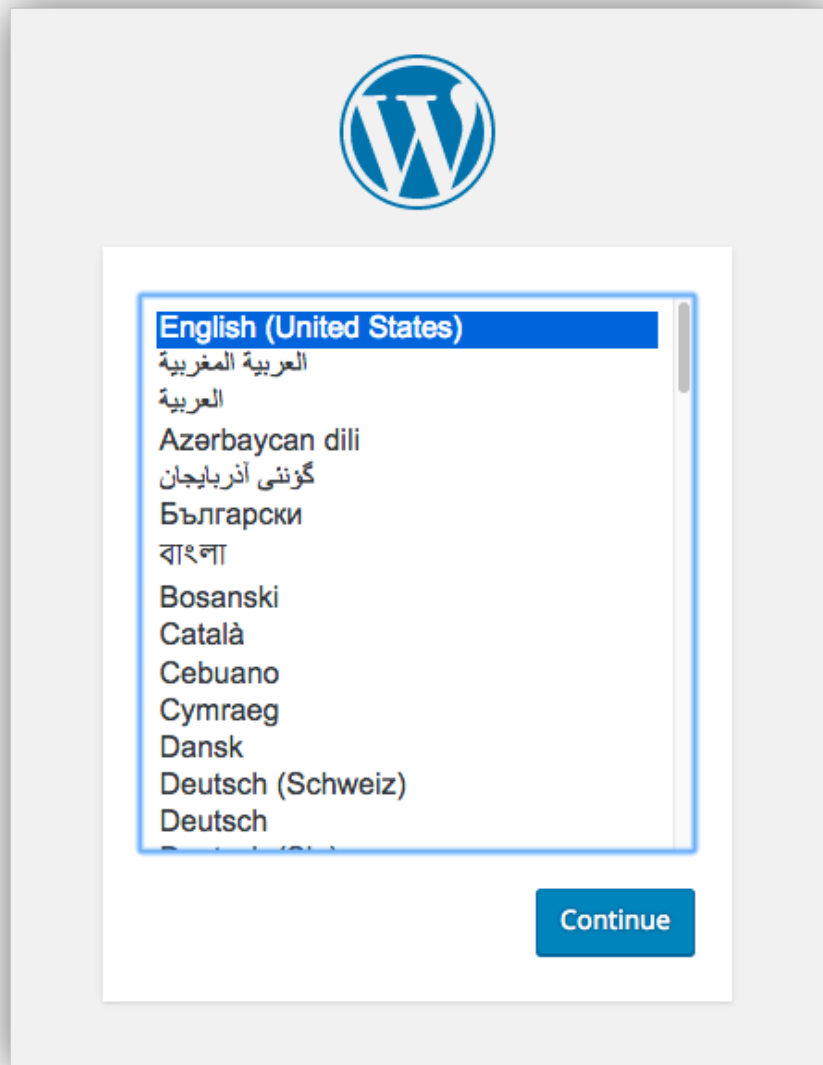


Figure 1: img

---

## Práctica Django

Siguiendo las instrucciones, instala Django

## Práctica Rails

Siguiendo las instrucciones, instala Rails

**Más info en**

- <https://docs.docker.com/compose/>
- <https://docs.docker.com/compose/wordpress/>