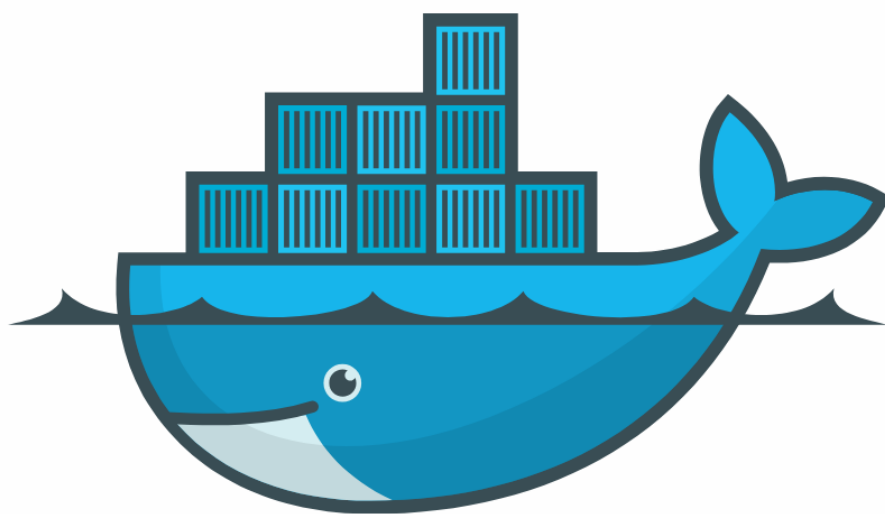


COMPOSE: INSTALACIÓN DE WORDPRESS



Ivana Sánchez Pérez

Introducción

Docker Compose es una herramienta que facilita la ejecución de aplicaciones multicontenedor en Docker. Permite definir y gestionar configuraciones de contenedores con un archivo YAML llamado `docker-compose.yml`.

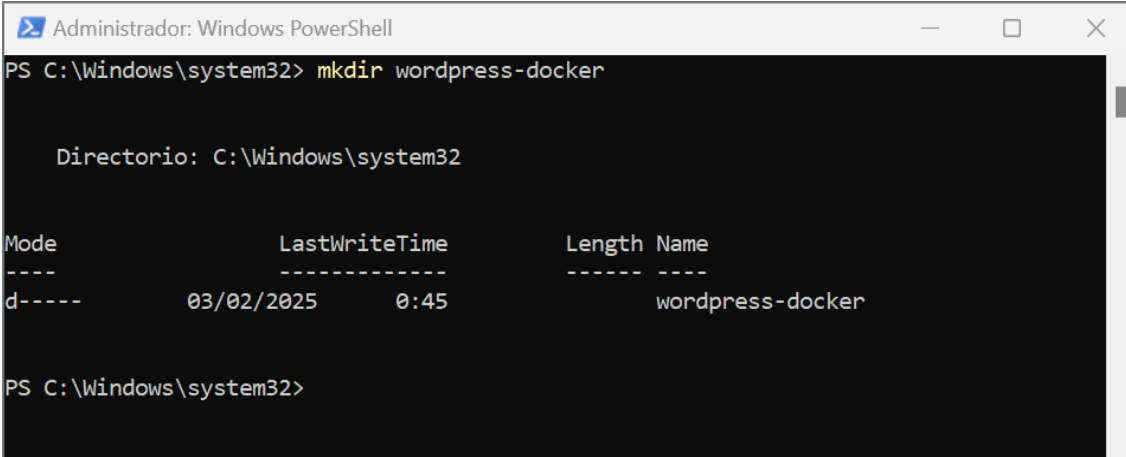
Con Docker Compose, puedes definir servicios como bases de datos, aplicaciones web y otros contenedores de manera sencilla, sin necesidad de escribir comandos largos y complejos para crear los contenedores individualmente.

En este caso, vamos a usar Docker Compose para configurar y ejecutar una aplicación de WordPress junto con una base de datos MySQL.

Crear un proyecto WordPress

Docker Compose viene incluido con la instalación de Docker Desktop en Windows, por lo que procedemos a crear un archivo **docker-compose.yml** que definirá los servicios de WordPress y MySQL.

Primero crearemos el directorio para nuestro proyecto de WordPress



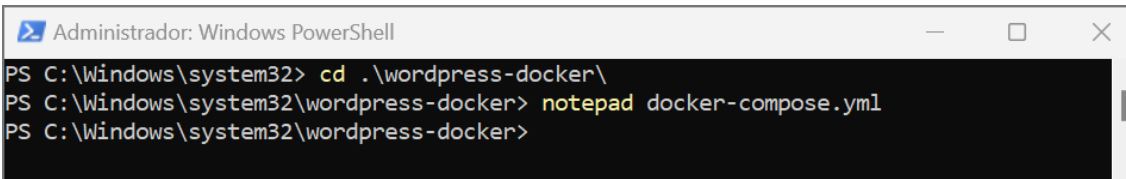
```
Administrador: Windows PowerShell
PS C:\Windows\system32> mkdir wordpress-docker

Directorio: C:\Windows\system32

Mode                LastWriteTime         Length Name
----                -
d-----          03/02/2025   0:45                wordpress-docker

PS C:\Windows\system32>
```

Nos ubicaremos en él, y crearemos el archivo **docker-compose.yml** con Notepad.



```
Administrador: Windows PowerShell
PS C:\Windows\system32> cd .\wordpress-docker\
PS C:\Windows\system32\wordpress-docker> notepad docker-compose.yml
PS C:\Windows\system32\wordpress-docker>
```

```
docker-compose.yml
+

Archivo  Editar  Ver

version: '3.1'

services:
  wordpress:
    image: wordpress:latest
    restart: always
    ports:
      - "8080:80" # Mapea el puerto 80 del contenedor al 8080 del host
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_NAME: exampledb
      WORDPRESS_DB_USER: root
      WORDPRESS_DB_PASSWORD: examplepassword
    volumes:
      - wordpress_data:/var/www/html # Persistencia de datos de WordPress
    depends_on:
      - db # Espera que el contenedor db esté listo

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: examplepassword
      MYSQL_DATABASE: exampledb
    volumes:
      - db_data:/var/lib/mysql # Persistencia de datos de la base de datos

volumes:
  wordpress_data:
  db_data:
```

Ln 31, Col 1 | 755 caracteres.

Explicación del archivo:

- **wordpress:** Es el contenedor de WordPress.
 - **image: wordpress:latest:** Utiliza la última versión de la imagen oficial de WordPress.
 - **ports:** Mapea el puerto 8080 en el host al puerto 80 del contenedor.
 - **environment:** Define las variables de entorno necesarias para conectar WordPress con MySQL.
 - **volumes:** Asegúrese de que los datos de WordPress se mantengan incluso si el contenedor se detiene.
 - **depends_on:** Asegúrese de que el contenedor de WordPress espere a que el contenedor de la base de datos se inicie.

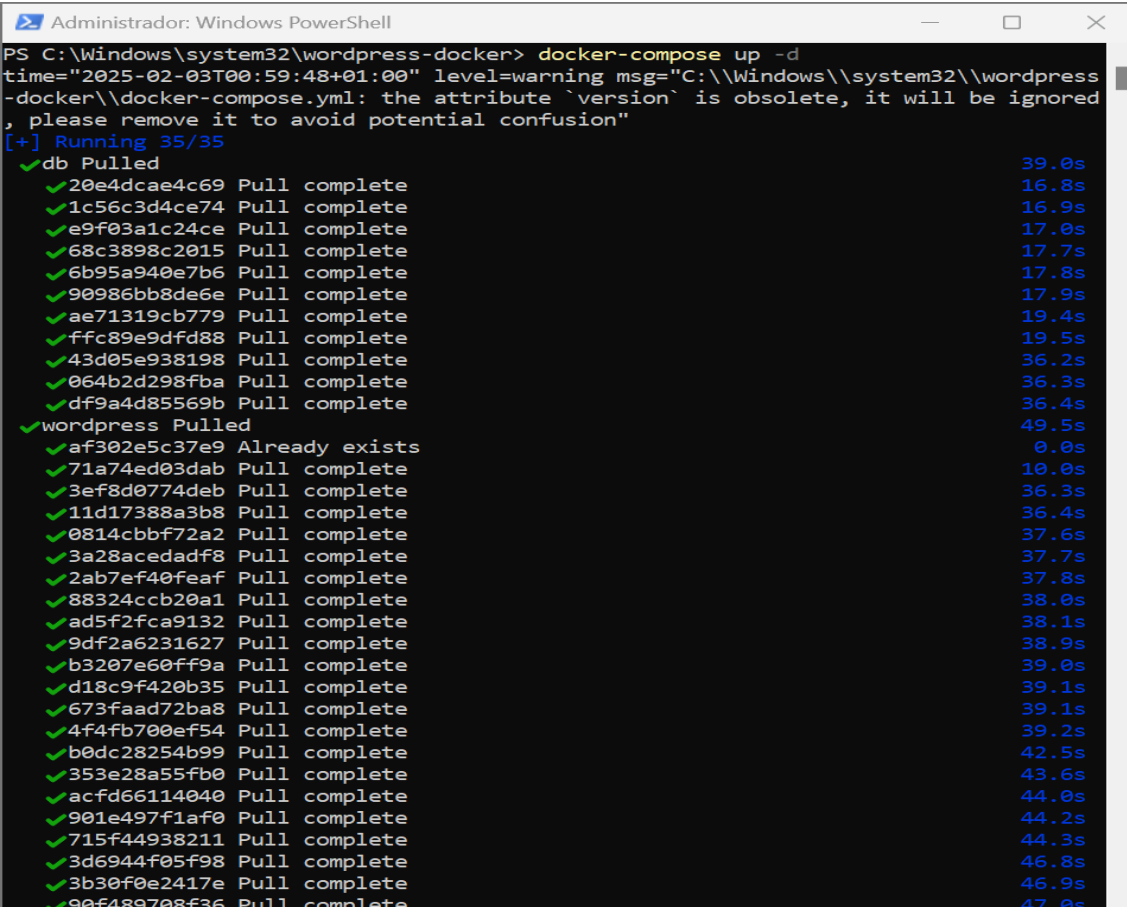
- **db:** Es el contenedor de la base de datos MySQL.
 - image: mysql:5.7: Utiliza la versión 5.7 de la imagen oficial de MySQL.
 - environment: Establece la contraseña de root y el nombre de la base de datos.
 - volumes: Asegúrese de que los datos de la base de datos persistan.
- **volumes:** Define dos volúmenes persistentes para almacenar los datos de WordPress y la base de datos MySQL.

Levantar los contenedores

Levantaremos los contenedores con **docker-compose up -d**

Explicación :

- up: Levante los contenedores definidos en el archivo **docker-compose.yml**.
- -d: Ejecuta los contenedores en segundo plano (modo separado).

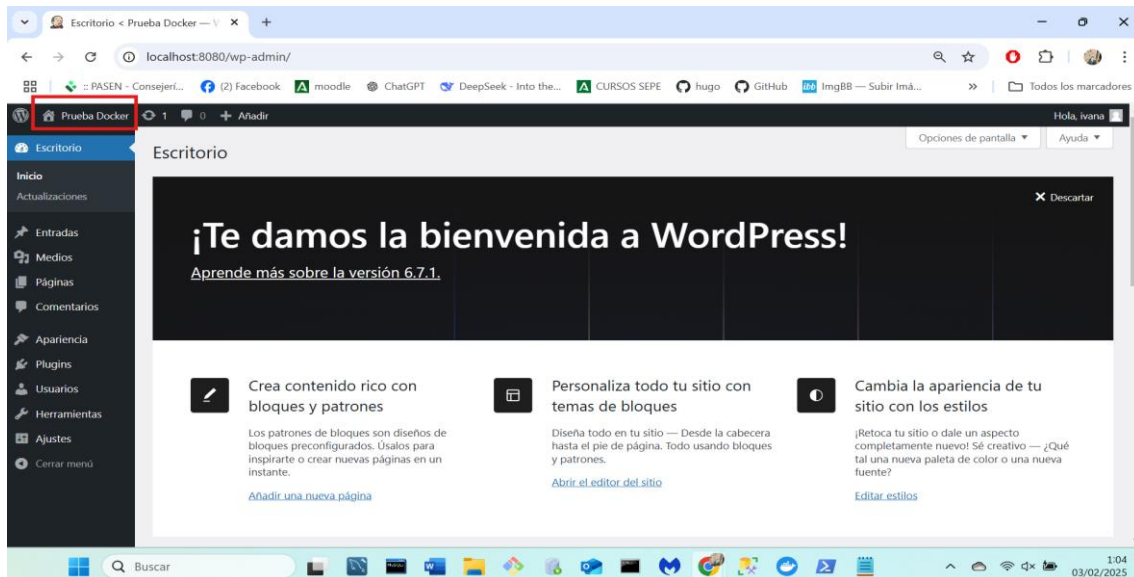
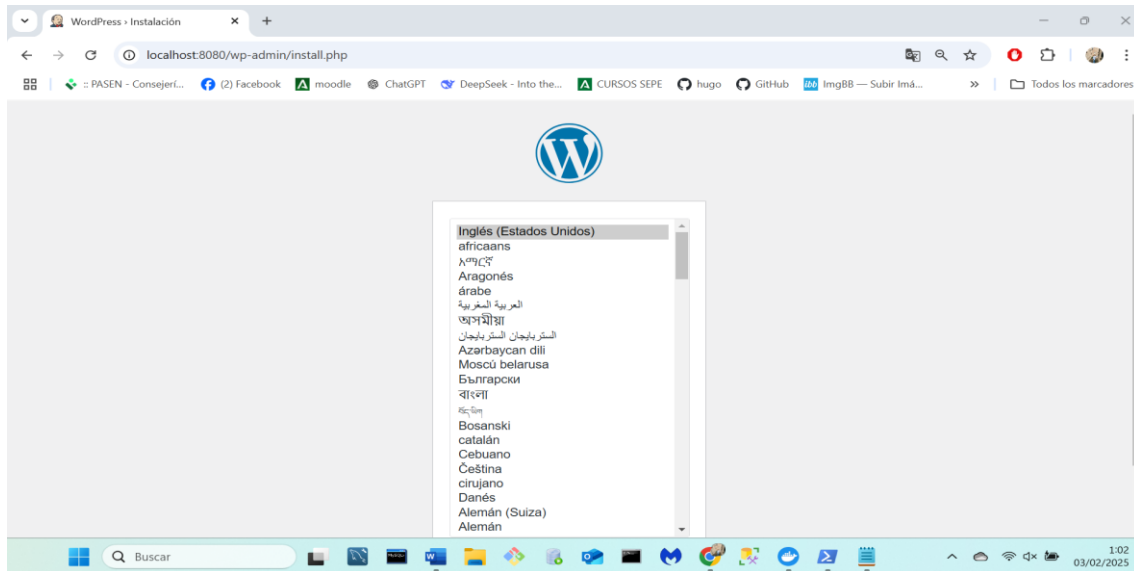


```

PS C:\Windows\system32\wordpress-docker> docker-compose up -d
time="2025-02-03T00:59:48+01:00" level=warning msg="C:\\Windows\\system32\\wordpress-docker\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 35/35
✔ db Pulled 39.0s
✔ 20e4dcae4c69 Pull complete 16.8s
✔ 1c56c3d4ce74 Pull complete 16.9s
✔ e9f03a1c24ce Pull complete 17.0s
✔ 68c3898c2015 Pull complete 17.7s
✔ 6b95a940e7b6 Pull complete 17.8s
✔ 90986bb8de6e Pull complete 17.9s
✔ ae71319cb779 Pull complete 19.4s
✔ ffc89e9dfd88 Pull complete 19.5s
✔ 43d05e938198 Pull complete 36.2s
✔ 064b2d298fba Pull complete 36.3s
✔ df9a4d85569b Pull complete 36.4s
✔ wordpress Pulled 49.5s
✔ af302e5c37e9 Already exists 0.0s
✔ 71a74ed03dab Pull complete 10.0s
✔ 3ef8d0774deb Pull complete 36.3s
✔ 11d17388a3b8 Pull complete 36.4s
✔ 0814cbbf72a2 Pull complete 37.6s
✔ 3a28acedadf8 Pull complete 37.7s
✔ 2ab7ef40feaf Pull complete 37.8s
✔ 88324ccb20a1 Pull complete 38.0s
✔ ad5f2fca9132 Pull complete 38.1s
✔ 9df2a6231627 Pull complete 38.9s
✔ b3207e60ff9a Pull complete 39.0s
✔ d18c9f420b35 Pull complete 39.1s
✔ 673faad72ba8 Pull complete 39.1s
✔ 4f4fb700ef54 Pull complete 39.2s
✔ b0dc28254b99 Pull complete 42.5s
✔ 353e28a55fb0 Pull complete 43.6s
✔ acfd66114040 Pull complete 44.0s
✔ 901e497f1af0 Pull complete 44.2s
✔ 715f44938211 Pull complete 44.3s
✔ 3d6944f05f98 Pull complete 46.8s
✔ 3b30f0e2417e Pull complete 46.9s
✔ 90f489708f36 Pull complete 47.0s
  
```

```
[+] Running 5/5
✓Network wordpress-docker_default          Created           0.1s
✓Volume "wordpress-docker_wordpress_data"  Created           0.0s
✓Volume "wordpress-docker_db_data"         Created           0.0s
✓Container wordpress-docker-db-1           Started           2.1s
✓Container wordpress-docker-wordpress-1    Started           1.8s
PS C:\Windows\system32\wordpress-docker>
```

Verificar funcionamiento de WordPress



Verificar persistencia de los datos

Podemos verificar los volúmenes que Docker ha creado para persistir con el comando `docker volume ls`

```
Administrador: Windows PowerShell
PS C:\Windows\system32\wordpress-docker> docker volume ls
DRIVER      VOLUME NAME
local       3b69f8840245f2eead59b5717896fc366fe9b6213ba8392df5320b5edb45e6d1
local       35117ca942260842487cca5066b2599587fa487b2b7dec2cc54f95a7ba020595
local       fbf9c651c67f67287d522bb0ef642d3b7361c92ec2f22774b429dc50a06f51e
local       jenkins_home
local       wordpress-docker_db_data
local       wordpress-docker_wordpress_data
PS C:\Windows\system32\wordpress-docker>
```

Para probar la persistencia, detendremos y eliminaremos los contenedores con **`docker-compose down`**.

```
Administrador: Windows PowerShell
PS C:\Windows\system32\wordpress-docker> docker-compose down
time="2025-02-03T01:10:00+01:00" level=warning msg="C:\\Windows\\system32\\wordpress-docker\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 3/3
✔ Container wordpress-docker-wordpress-1   Removed      1.7s
✔ Container wordpress-docker-db-1          Removed      2.0s
✔ Network wordpress-docker_default         Removed      0.3s
PS C:\Windows\system32\wordpress-docker>
```

Volveremos a levantarlos y accederemos de nuevo a localhost para verificar que nuestra instalación de WordPress siga intacta con los mismos datos (temas, plugins, entradas).

```
Administrador: Windows PowerShell
PS C:\Windows\system32\wordpress-docker> docker-compose up -d
time="2025-02-03T01:11:58+01:00" level=warning msg="C:\\Windows\\system32\\wordpress-docker\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 3/3
✔ Network wordpress-docker_default         Created      0.1s
✔ Container wordpress-docker-db-1          Started      1.1s
✔ Container wordpress-docker-wordpress-1   Started      1.5s
PS C:\Windows\system32\wordpress-docker>
```

