

# Docker Taller 4

## 1. Creación docker-compose

Primero creamos el docker-compose.yaml. Para hacerlo persistente creo un volumen cuya ubicación se va a encontrar en la misma carpeta donde guardo este taller.

```
1 version: "3.7"
2
3 services:
4   mariadb:
5     image: mariadb
6     container_name: mariadb_cont
7     volumes:
8       - C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatcenter-docker-t4-jmmd\data:/var/lib/mysql
9     environment:
10      - MYSQL_ROOT_PASSWORD=root
11      - MYSQL_DATABASE=wordpress
12      - MYSQL_USER=root
13      - MYSQL_PASSWORD=root
14   wordpress:
15     image: wordpress
16     container_name: wordpress_cont
17     depends_on:
18       - mariadb
19     environment:
20      - WORDPRESS_DB_USER=root
21      - WORDPRESS_DB_PASSWORD=root
22      - WORDPRESS_DB_HOST=mariadb
23     ports:
24      - 8080:80
25
26
```

## 2. Lanzar el Docker-compose.yaml

Una vez creado el docker-compose.yaml lo lanzamos usando el comando up. Esto automáticamente nos iniciará ambos contenedores, uno para mariadb y otro para wordpress. Además, nos creará automáticamente la network.

```
C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatcenter-docker-t4-jmmd>docker-compose up -d
[*] Running 2/2
- Container mariadb_cont Started 0.5s
- Container wordpress_cont Running 0.0s

C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatcenter-docker-t4-jmmd>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
1baca0080845   wordpress  "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:8080->80/tcp    wordpress_cont
ce5e9ba2dfaf   mariadb    "docker-entrypoint.s..." About a minute ago Up 7 seconds   3306/tcp            mariadb_cont
```

## 3. Comprobación de localhost

Si ponemos localhost:8080, que es puerto que le hemos indicado, en nuestro navegador, podemos ver que funciona correctamente

