

# PrestaShop

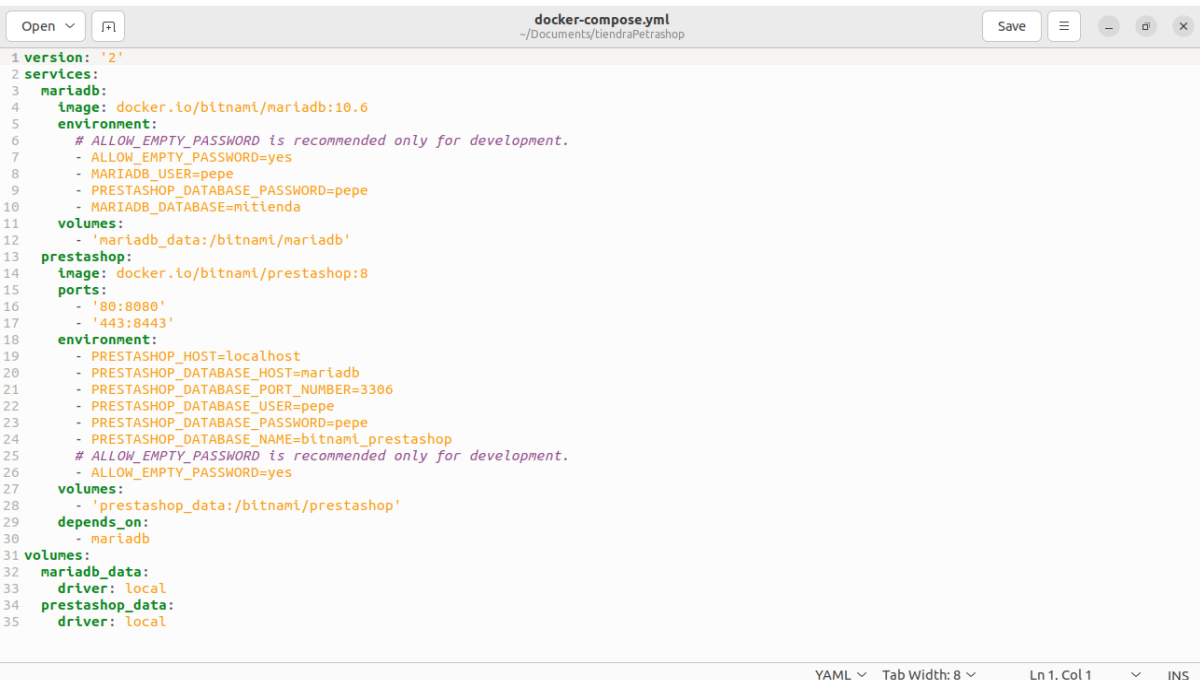


# PrestaShop

# Índice

1. [Explicación docker-compose.yml](#)
2. [Ejecución comando](#)
3. [Aplicación en ejecución](#)

# Explicación docker-compose.yml



```
1 version: '2'
2 services:
3   mariadb:
4     image: docker.io/bitnami/mariadb:10.6
5     environment:
6       # ALLOW_EMPTY_PASSWORD is recommended only for development.
7       - ALLOW_EMPTY_PASSWORD=yes
8       - MARIADB_USER=pepe
9       - PRESTASHOP_DATABASE_PASSWORD=pepe
10      - MARIADB_DATABASE=mittenda
11     volumes:
12       - 'mariadb_data:/bitnami/mariadb'
13   prestashop:
14     image: docker.io/bitnami/prestashop:8
15     ports:
16       - '80:8080'
17       - '443:8443'
18     environment:
19       - PRESTASHOP_HOST=localhost
20       - PRESTASHOP_DATABASE_HOST=mariadb
21       - PRESTASHOP_DATABASE_PORT_NUMBER=3306
22       - PRESTASHOP_DATABASE_USER=pepe
23       - PRESTASHOP_DATABASE_PASSWORD=pepe
24       - PRESTASHOP_DATABASE_NAME=bitnami_prestashop
25       # ALLOW_EMPTY_PASSWORD is recommended only for development.
26       - ALLOW_EMPTY_PASSWORD=yes
27     volumes:
28       - 'prestashop_data:/bitnami/prestashop'
29     depends_on:
30       - mariadb
31 volumes:
32   mariadb_data:
33     driver: local
34   prestashop_data:
35     driver: local
```

En la línea 2 abrimos la etiqueta **services**, donde incluiremos todos los contenedores que necesitamos.

En la línea 3 indicamos el servicio que utilizaremos, en nuestro será mariadb y dentro especificamos sus valores. Indicamos la imagen que usará mediante la url del repositorio de docker. Seguido de esto declaramos todas las variables de entorno, en la línea 7 indicamos que la contraseña estará vacía que se usa en estado de pruebas, en la línea 8 indicamos el usuario de la base de datos seguido de la contraseña. Por último indicamos el nombre de la base de datos.

Luego creamos el otro servicio que en este caso es prestashop, seguimos los pasos

# Ejecución comando

```
estudiante@DAW1: ~/Documents/tiendaPetrashop
estudiante@DAW1:~/Documents/tiendaPetrashop$ docker compose up
WARN[0000] volume "tiendrapetrashop_mariadb_data" already exists but was not cre
ated by Docker Compose. Use `external: true` to use an existing volume
WARN[0000] volume "tiendrapetrashop_prestashop_data" already exists but was not
created by Docker Compose. Use `external: true` to use an existing volume
[+] Running 2/0
  :: Container tiendrapetrashop-mariadb-1   Recreated          0.0s
  :: Container tiendrapetrashop-prestashop-1 Recreated          0.0s
Attaching to tiendrapetrashop-mariadb-1, tiendrapetrashop-prestashop-1
tiendrapetrashop-mariadb-1 | mariadb 08:30:53.42
tiendrapetrashop-mariadb-1 | mariadb 08:30:53.42 Welcome to the Bitnami mari
adb container
tiendrapetrashop-mariadb-1 | mariadb 08:30:53.42 Subscribe to project update
s by watching https://github.com/bitnami/containers
tiendrapetrashop-mariadb-1 | mariadb 08:30:53.42 Submit issues and feature r
equests at https://github.com/bitnami/containers/issues
tiendrapetrashop-mariadb-1 | mariadb 08:30:53.42
tiendrapetrashop-mariadb-1 | mariadb 08:30:53.43 INFO ==> ** Starting Maria
DB setup **
tiendrapetrashop-mariadb-1 | mariadb 08:30:53.45 INFO ==> Validating settin
gs in MYSQL_*/MARIADB_* env vars
tiendrapetrashop-mariadb-1 | mariadb 08:30:53.45 WARN ==> You set the enviro
nment variable ALLOW_EMPTY_PASSWORD=yes. For safety reasons, do not use this fl
ag in a production environment.
```

# Aplicación en ejecución

