

# Como levantar MOODLE con docker



<b>1. Obtener la imagen.</b>	<b>2</b>
a. Realizar un docker pull.	2
b. Crear la build nosotros mismos.	2
<b>2. Cómo utilizar la imagen.</b>	<b>2</b>
a. Utilizando Docker Compose:	3
b. Utilizando la línea de comando de Docker:	3
<b>3. Persiste tu aplicación.</b>	<b>3</b>
Montar un host como data volume con Docker compose.	4

# 1. Obtener la imagen.

Lo primero que tendremos que hacer para levantar nuestra moodle es obtener la imagen, para ello tenemos dos opciones.

## a. Realizar un docker pull.

El caso de querer la última versión utilizaremos:

```
docker pull bitnami/moodle:latest
```

En el caso de querer utilizar cualquier otra utilizaremos el siguiente comando:

```
docker pull bitnami/moodle:[TAG]
```

## b. Crear la build nosotros mismos.

Para ello puedes crear la imagen tu mismo clonando el repositorio, cambiando al directorio que contiene el Dockerfile y ejecutando el comando de compilación de docker. Recuerde reemplazar los marcadores de posición de la ruta APP, VERSION y OPERATING-SYSTEM en el siguiente comando de ejemplo con los valores correctos.

```
$ git clone https://github.com/bitnami/containers.git
$ cd bitnami/APP/VERSION/OPERATING-SYSTEM
$ docker build -t bitnami/APP:latest .
```

# 2. Cómo utilizar la imagen.

Moodle necesita acceso a MySQL o MariaDB. Para ello utilizaremos Bitnami Docker image for MariaDB en los requerimientos de la base de datos.

Para ello volvemos a tener dos opciones:

## a. Utilizando Docker Compose:

Para ello utilizaremos el comando:

```
$ curl -sL
https://raw.githubusercontent.com/bitnami/containers/main/bitnami/moodle/docker-compose.
yml > docker-compose.yml
$ docker-compose up -d
```

## **b. Utilizando la línea de comando de Docker:**

### **Paso 1: Crear la red:**

```
$ docker network create moodle-network
```

### **Paso 2: Crear un volume para la persistencia y el contenedor de MariaDB:**

```
$ docker volume create --name mariadb_data
```

```
$ docker run -d --name mariadb \
--env ALLOW_EMPTY_PASSWORD=yes \
--env MARIADB_USER=bn_moodle \
--env MARIADB_PASSWORD=bitnami \
--env MARIADB_DATABASE=bitnami_moodle \
--network moodle-network \
--volume mariadb_data:/bitnami/mariadb \
bitnami/mariadb:latest
```

### **Paso 3: Crear un volumen para la persistencia y la ejecución del contenedor de moodle:**

```
$ docker volume create --name moodle_data
$ docker run -d --name moodle \
-p 8080:8080 -p 8443:8443 \
--env ALLOW_EMPTY_PASSWORD=yes \
--env MOODLE_DATABASE_USER=bn_moodle \
--env MOODLE_DATABASE_PASSWORD=bitnami \
--env MOODLE_DATABASE_NAME=bitnami_moodle \
--network moodle-network \
--volume moodle_data:/bitnami/moodle \
bitnami/moodle:latest
```

## **3. Persiste tu aplicación.**

Si eliminamos el contenedor perderemos toda la información y la próxima vez que abramos la imagen esta será reiniciada.

Para persistencia, debe montar un directorio en la ruta /bitnami/moodle. Si el directorio montado está vacío, se inicializará en la primera ejecución. Además, debe montar un volumen para la persistencia de los datos de MariaDB.

Para ello tendremos que:

### Montar un host como data volume con Docker compose.

Para ello tendremos que cambiar el archivo “**docker-compose.yml**” de tal forma que quede algo así.

```
    mariadb:
        ...
        volumes:
-         - 'mariadb_data:/bitnami/mariadb'
+         - /path/to/mariadb-persistence:/bitnami/mariadb
        ...
    moodle:
        ...
        volumes:
-         - 'moodle_data:/bitnami/moodle'
+         - /path/to/moodle-persistence:/bitnami/moodle
        ...
-volumes:
-  mariadb_data:
-    driver: local
-  moodle_data:
-    driver: local
```