# MEMORIA DE PRÁCTICAS PARA DESPLEGAR UN CONTENEDOR CON MOODLE

SANDRA RUIZ JIMÉNEZ
DESPLIEGUE DE APLICACIONES WEB | 2º DAW

# Tabla de contenido

Moodle™ es un sistema de gestión de aprendizaje, gratuito y de código abierto escrito en PHP y distribuido bajo la Licencia Pública General GNU.

En la siguiente memoria se detallan los pasos para instalar Moodle usando contenedores de Docker.

## Paso 1: Instalación de Moodle

Usamos la imagen de Bitnami para la instalación de Docker con el siguiente comando:

**$ docker pull bitnami/moodle**

```
sandra@sandra-UX410UAK:~$ docker pull bitnami/moodle
Using default tag: latest
latest: Pulling from bitnami/moodle
1d8866550bdd: Pull complete
dc87c8d14191: Pull complete
Digest: sha256:6633fa0f5ca3f005f94cd1b17d284ad0e9e960d1e8638a9ce8b771731ac1e97d
Status: Downloaded newer image for bitnami/moodle:latest
docker.io/bitnami/moodle:latest
```

## Paso 2: Instalación de la imagen de la base de datos con Docker Compose

Ejecutamos la aplicación con los siguientes comandos:

**$ curl -sSL https://raw.githubusercontent.com/bitnami/containers/main/bitnami/moodle/docker-compose.yml > docker-compose.yml**

```
sandra@sandra-UX410UAK:~$ curl -sSL https://raw.githubusercontent.com/bitnami/container
s/main/bitnami/moodle/docker-compose.yml > docker-compose.yml
```

**$ docker-compose up -d**

```
sandra@sandra-UX410UAK:~$ docker-compose up -d
Creating network "sandra_default" with the default driver
Creating volume "sandra_mariadb_data" with local driver
Creating volume "sandra_moodle_data" with local driver
Creating volume "sandra_moodledata_data" with local driver
Pulling mariadb (docker.io/bitnami/mariadb:10.6)...
10.6: Pulling from bitnami/mariadb
1d8866550bdd: Already exists
1cdd9548fd70: Downloading [>                                                  ] 1cdd954
8fd70: Downloading [>                                                      ] 1cdd9548fd70:
Downloading [=>                                                  ] 1cdd9548fd70: Downloa
ding [===>                                                   ] 1cdd9548fd70: Pull complete

Digest: sha256:3672fbb111fbfeef7c24a001f877d106a4e26e0148ccead07266b1a9bc312222
Status: Downloaded newer image for bitnami/mariadb:10.6
Pulling moodle (docker.io/bitnami/moodle:4)...
4: Pulling from bitnami/moodle
Digest: sha256:6633fa0f5ca3f005f94cd1b17d284ad0e9e960d1e8638a9ce8b771731ac1e97d
Status: Downloaded newer image for bitnami/moodle:4
Creating sandra_mariadb_1 ... done
Creating sandra_moodle_1  ... done
```

Ahora podremos acceder a nuestra aplicación en http://nuestra-ip/

## Paso 3: Persistir la aplicación

Si eliminásemos el contenedor, se perderían todos los datos y la próxima vez que ejecutásemos la imagen, la base de datos se reiniciaría.
Para evitar esta pérdida de datos, debemos montar un volumen que persistirá incluso después de eliminar el contenedor.
Para ello, debemos montar un directorio en la /bitnami/ruta_de_moodle. Si el directorio montado está vacío, se inicializará en la primera ejecución.
Además, debemos montar un volumen para la persistencia de los datos de MariaDB. Podemos hacerlo, modificando el fichero docker-compose.yml con lo siguiente:

```
services:
  mariadb:
  ...
    volumes:
      - /path/to/mariadb-persistence:/bitnami/mariadb
  ...
```

## Paso 4: Montar directorios de host como volúmenes de datos con Docker Compose

Para evitar la eliminación accidental de volúmenes, podemos montar directorios de host como volúmenes de datos.

Podemos hacerlo, modificando el fichero docker-compose.yml con lo siguiente:

```
   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
```

## Paso 5: Configuración de variables de entorno

Cuando iniciamos la imagen de Moodle™, podemos ajustar la configuración de la instancia con una o más variables de entorno.
Para agregar una nueva variable de entorno, agregamos el nombre y el valor de la variable en la sección de la aplicación en el fichero docker-compose.yml.

```
moodle:
  ...
  environment:
    - MOODLE_PASSWORD=my_password
  ...
```

### Variables de entorno de configuración y sitio

- MOODLE_USERNAME: nombre de usuario de la aplicación Moodle. Predeterminado: user
- MOODLE_PASSWORD: Contraseña de la aplicación Moodle. Predeterminado: bitnami
- MOODLE_EMAIL: Correo electrónico de la aplicación Moodle. Predeterminado: user@example.com
- MOODLE_SITE_NAME: nombre del sitio de Moodle. Predeterminado: New Site
- MOODLE_SKIP_BOOTSTRAP: No inicializa la base de datos de Moodle para un nuevo despliegue. Esto es necesario en caso de que se use una base de datos que ya tenga datos de Moodle. Predeterminado: no
- MOODLE_HOST: Permite configurar la función wwwroot de Moodle. Ej: example.com. Por defecto es una variable superglobal de PHP. Predeterminado: $_SERVER['HTTP_HOST']
- MOODLE_REVERSEPROXY: Permite activar la función de proxy inverso de Moodle. Predeterminado: no
- MOODLE_SSLPROXY: Permite activar la función sslproxy de Moodle. Predeterminado: no
- MOODLE_LANG: Permite establecer el idioma predeterminado del sitio. Predeterminado: en

### Variables de entorno para usar una base de datos existente

- MOODLE_DATABASE_TYPE: Tipo de base de datos. Valores válidos: *mariadb*, *mysqli*, *pgsql*, *auroramysql*. Predeterminado: mariadb
- MOODLE_DATABASE_HOST: Nombre de host para el servidor de la base de datos. Predeterminado: mariadb
- MOODLE_DATABASE_PORT_NUMBER: Puerto utilizado por el servidor de la base de datos. Predeterminado: 3306
- MOODLE_DATABASE_NAME: Nombre de la base de datos que usará Moodle para conectarse con la base de datos. Predeterminado: bitnami_moodle
- MOODLE_DATABASE_USER: Usuario de la base de datos que Moodle usará para conectarse con la base de datos. Predeterminado: bn_moodle

- MOODLE_DATABASE_PASSWORD: Contraseña de la base de datos que utilizará Moodle para conectarse con la base de datos. Sin valores predeterminados.
- ALLOW_EMPTY_PASSWORD: Se puede utilizar para permitir contraseñas en blanco. Predeterminado: no

**Variables de entorno para configuración SMTP**

- MOODLE_SMTP_HOST: Servidor SMTP.
- MOODLE_SMTP_PORT: Puerto SMTP.
- MOODLE_SMTP_USER: Usuario de la cuenta SMTP.
- MOODLE_SMTP_PASSWORD: Contraseña de la cuenta SMTP.
- MOODLE_SMTP_PROTOCOL: Protocolo SMTP.

**Variables de entorno para configuración PHP**

- PHP_ENABLE_OPCACHE: Habilita OPcache para scripts PHP. Ningún valor predeterminado.
- PHP_EXPOSE_PHP: Habilita el encabezado HTTP con la versión de PHP. Ningún valor predeterminado.
- PHP_MAX_EXECUTION_TIME: Tiempo máximo de ejecución para scripts PHP. Ningún valor predeterminado.
- PHP_MAX_INPUT_TIME: Tiempo máximo de entrada para scripts PHP. Ningún valor predeterminado.
- PHP_MAX_INPUT_VARS: Cantidad máxima de variables de entrada para scripts PHP. Ningún valor predeterminado.
- PHP_MEMORY_LIMIT: Límite de memoria para scripts PHP. Predeterminado: 256M
- PHP_POST_MAX_SIZE: Tamaño máximo para solicitudes PHP POST. Ningún valor predeterminado.
- PHP_UPLOAD_MAX_FILESIZE: Tamaño máximo de archivo para cargas de PHP. Ningún valor predeterminado.

## Paso 6: Inicio de sesión

La imagen de Bitnami Docker para Moodle™ envía los registros del contenedor a stdout.
Para ver dichos registros, ejecutamos:
**$ docker-compose logs moodle**

## Paso 7: Mantenimiento

### Copia de seguridad del contenedor

1. Detener el contenedor que se está ejecutando:
   **$ docker-compose stop moodle**
2. Ejecutar el comando de copia de seguridad:
   **$ docker run --rm -v /path/to/moodle-backups:/backups --volumes-from moodle busybox \ cp -a /bitnami/moodle /backups/latest**

### Restaurar una copia de seguridad

Restaurar una copia de seguridad es tan simple como montar la copia de seguridad como volúmenes en los contenedores.

Para el contenedor de la base de datos MariaDB:

```
$ docker run -d --name mariadb \
  ...
- --volume /path/to/mariadb-persistence:/bitnami/mariadb \
+ --volume /path/to/mariadb-backups/latest:/bitnami/mariadb \
  bitnami/mariadb:latest
```

Para el contenedor Moodle™:

```
$ docker run -d --name moodle \
  ...
- --volume /path/to/moodle-persistence:/bitnami/moodle \
+ --volume /path/to/moodle-backups/latest:/bitnami/moodle \
  bitnami/moodle:latest
```

## Paso 8: Actualizar la imagen de Moodle

1. Obtener la imagen actualizada:
   **$ docker pull bitnami/moodle:latest**

2. Detener el contenedor en ejecución:
   **$ docker-compose stop moodle**

3. Hacer una copia de seguridad del contenedor para tomar una instantánea del estado actual de la aplicación.

4. Eliminar el contenedor que se está ejecutando actualmente:
   **$ docker-compose rm -v moodle**

5. Actualizar la etiqueta de la imagen docker-compose.yml.

6. Volver a crear el contenedor con la nueva imagen:
   **$ docker-compose up -d**