

# **Instalando LAMP + Wordpress en un contenedor**

INTRODUCCIÓN .....	3
PREPARANDO EL CONTENEDOR .....	3
INSTALANDO LAMP Y WORDPRESS .....	4
Preparando Apache para usar Wordpress .....	4
Preparando MySQL Server .....	6
Configurando Wordpress .....	8

## INTRODUCCIÓN

Vamos a utilizar como base de contenedor la imagen oficial de *"ubuntu"*. En ese contenedor instalaremos Apache, PHP y MySQL (LAMP), junto al CMS WordPress. Adelanto que esta instalación contendrá errores que se solucionarán en temas posteriores.

## PREPARANDO EL CONTENEDOR

Crearemos un contenedor con la imagen base *"ubuntu"*, al que llamaremos LAMP y en el que expondremos su puerto 80 dentro del puerto 8080 de nuestro sistema. Además al crearlo, dejaremos lista una *"shell"* para instalar los programas pertinentes.

```
docker run -it -p 8080:80 --name LAMP ubuntu /bin/bash
```

Cuando ejecutemos NO saldremos del contenedor.

```
jorge@jorgepc:~$ docker run -it -p 8080:80 --name LAMP ubuntu /bin/bash
root@372f7fd6ca22:/# exit
exit
jorge@jorgepc:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
jorge@jorgepc:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
372f7fd6ca22   ubuntu   "/bin/bash"   12 seconds ago   Exited (0) 6 seconds ago           LAMP
```

Aunque pudiera darse que hubiéramos salido entonces para volver a entrar deberemos arrancar el contenedor y entrar, recordemos que si ejecutamos otra vez un run, nos crea una nueva instancia.

```
jorge@jorgepc:~$ docker start LAMP
LAMP
jorge@jorgepc:~$ docker exec -it LAMP bash
root@372f7fd6ca22:/#
```

## INSTALANDO LAMP Y WORDPRESS

```
apt install nano;
```

Actualizando repositorio e instalando LAMP + Wordpress En primer lugar, actualizamos la lista de paquetes del repositorio:

```
apt update
```

### Preparando Apache para usar Wordpress

Antes que nada, podemos instalar un editor de texto en modo consola que sea de nuestro agrado. Por ejemplo con este comando podemos instalar "nano":

Ahora ya podemos instalar nuestro servidor apache:

```
apt install apache2;
```

Tras ello, instalamos los paquetes necesarios para instalar LAMP .

```
apt install php libapache2-mod-php php-mysql
```


Tras ello podemos lanzar el servicio Apache con el comando:

```
service apache2 start
```

Con esto ya tenemos todo el software instalado. Podemos hacer ya una pequeña prueba conectando desde nuestra máquina <http://localhost:8080>

Apache2 Ubuntu Default Page

localhost:8080



# Apache2 Default Page

# Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled
```

## Preparando MySQL Server

Una vez todo instalado, debemos preparar el servicio MySQL. En primer lugar, deberemos poner

```
service mysql start  
apt install mysql-server
```

```
service mysql start
```

en marcha el servicio con el comando

Tras ello, deberemos ejecutar el comando para generar un password de root de MySQL Server de forma segura (deberemos recordarlo) y otras opciones.

```
mysql_secure_installation
```

Si por alguna de aquellas me surgiera algún error ejecutaré el siguiente comando:

```
mysql> alter user 'root'@'localhost' identified with mysql_native_password by '12345678';  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> exit  
Bye  
root@372f7fd6ca22:/# mysql_secure_installation
```

Tras ello, accederemos a la base de datos con el cliente MySQL de la siguiente forma

```
mysql -u root -p
```

```
root@372f7fd6ca22:/# mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 20  
Server version: 8.0.30-0ubuntu0.22.04.1 (Ubuntu)  
  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement  
mysql>
```

Tras indicar la contraseña de "root", podremos escribir comandos para MySQL. Escribiremos los siguientes comandos. En primer lugar creamos la base de datos *"wordpress"*:

```
CREATE DATABASE wordpress;
```

Tras ellos, creamos el usuario "wordpress" (con contraseña "12345678") y le damos permisos totales en la base de datos "wordpress".

```
CREATE USER 'wordpress'@'%' IDENTIFIED BY '12345678';
```

```
GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'%' WITH GRANT  
OPTION;
```

Finalmente, propagamos los privilegios establecidos para que ya estén operativos en el servidor.

```
FLUSH PRIVILEGES;
```

## Configurando Wordpress

Podremos descargar la última versión de wordpress via consola mediante:

```
wget https://wordpress.org/latest.tar.gz
```

Y extraeremos la carpeta con el comando:

```
tar xzf latest.tar.gz
```

Más tarde moveremos la carpeta wordpress a /var/www/html

```
tar xzf latest.tar.gz
```

Configurando nuestro sitio Wordpress desde el navegador

Si todo ha ido bien, accediendo a la url <http://localhost:8080/wordpress> podremos configurar nuestro sitio Wordpress. Esta configuración no funciona totalmente correcta ya que docker no está pensado para que en un mismo contenedor haya múltiples servicios, aún así SI que se podría llegar a hacer funcionar pero el tiempo que hay que dedicar es enorme y en temas posteriores veremos como realizar configuraciones multiservicio.