

LAMP

Instalación de la pila LAMP

Víctor Ponz



Contenidos

0.1	LAMP	3
0.2	Paso Uno. Instalar apache	3
0.3	Paso Dos. Instalar MySQL (opcional)	4
0.4	Paso Tres. Instalar PHP	5
0.5	Paso Cuatro. Prueba del Procesador PHP en el Servidor Web	6
0.6	Paso adicional. Instalación de phpMyAdmin	8

0.1 LAMP

Se denomina «LAMP» a un grupo de software de código libre que se instala normalmente en conjunto para habilitar un servidor para alojar sitios y aplicaciones web dinámicas. Este término en realidad es un acrónimo que representa un sistema operativo **L**inux con un servidor **A**pache, los datos se almacenan en base de datos **M**ySQL y el contenido dinámico es procesado con **P**HP.

En esta guía, vamos a instalar LAMP en un desktop con Kali Linux. Los pasos son parecidos para instalarlo en Windows y Mac

0.2 Paso Uno. Instalar apache

El servidor Web **apache** es actualmente el más popular del mundo, lo que hace que sea una buena opción para montar nuestros sitios.

Podemos instalar **apache** fácilmente desde el gestor de paquetes de Kali, apt. Un gestor de paquetes nos permite instalar con mayor facilidad un software desde un repositorio conservado por Kali. Puedes aprender más sobre [como utilizar apt](#) aquí.

Para nuestros propósitos, podemos iniciar escribiendo los siguientes comandos:

```
sudo apt-get update
sudo apt-get install apache2
```

Ya que estamos utilizando el comando sudo, estas operaciones son ejecutadas con privilegios de administrador, por lo que te pedirá la contraseña para verificarlo.

Después de esto, ya tendremos instalado nuestro servidor web.

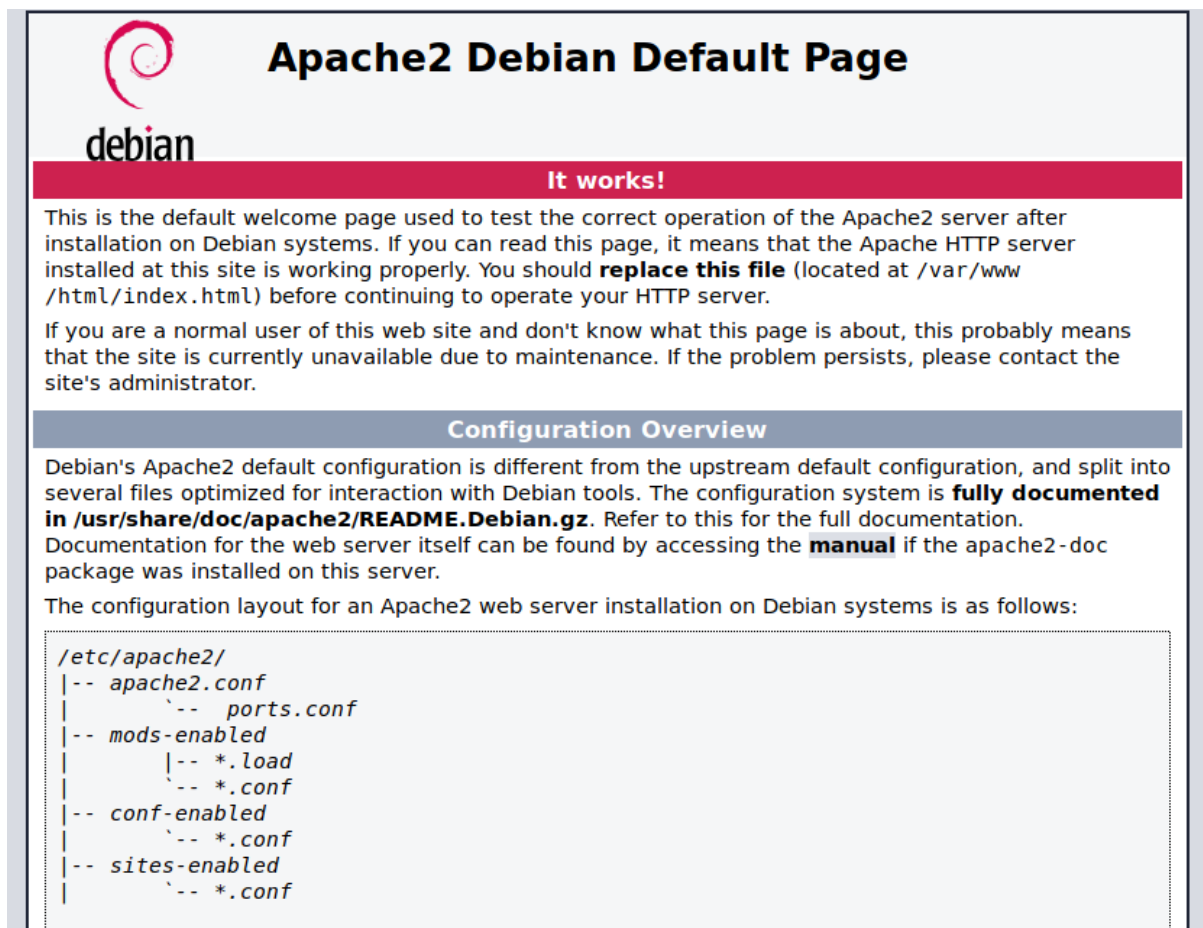
Para que se ejecute en esta sesión, lanza el comando (al reiniciar el sistema ya funcionará sin este paso)

```
systemctl start apache2
```

Puedes hacer una prueba después de esto para verificar que todo haya ido según lo previsto, visitando la dirección IP pública de tu servidor en el navegador web (ver la nota en el siguiente apartado para averiguar cuál es tu dirección IP pública, si es que no tienes esta información ya).

```
http://localhost
o
http://127.0.0.1
```

Podrás ver la imagen por defecto de la página web apache Ubuntu/Kali, que esta ahí para fines informativos. Deberá ser algo como esto:



Si puedes ver esta página, entonces tu servidor web ya se ha instalado correctamente.

0.3 Paso Dos. Instalar MySQL (opcional)

Ahora que ya tenemos nuestro servidor web configurado y corriendo, es el momento de instalar **MySQL** (o **MariaDB**). MySQL es un sistema de gestión de base de datos. Básicamente, se encarga de organizar y facilitar el acceso a las bases de datos donde nuestro sitio puede almacenar información.

En el caso de Kali, hemos de descargar la versión desde <https://dev.mysql.com>

```
sudo apt install -y wget
wget https://dev.mysql.com/get/mysql-apt-config_0.8.15-1_all.deb
sudo dpkg -i mysql-apt-config_0.8.15-1_all.deb
```

En el proceso de instalación seleccionaremos `deb ian buster`

Confirma los pasos por defecto pulsando <OK>

Para ejecutarlo en esta sesión usa:

```
sudo systemctl enable --now mysql
```

Y para crear la contraseña usa:

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

Nota.

Como contraseña usaremos **sa**

0.4 Paso Tres. Instalar PHP

PHP es el componente de nuestra configuración que procesará código para mostrar contenido dinámico. Puede ejecutar secuencias de comandos, conectarse a nuestras bases de datos MySQL para obtener información, y entregar el contenido procesado a nuestro servidor web para mostrarlo.

Una vez más podemos aprovechar el sistema `apt` para instalar nuestros componentes. Vamos a incluir algunos paquetes de ayuda, así:

```
sudo apt-get install php libapache2-mod-php
```

Instalaremos también el paquete de conexión `mysql` con `php`

```
sudo apt-get install php-mysql
```

Esto deberá instalar PHP sin ningún problema. Vamos a probar esto en un momento.

En la mayoría de los casos, vamos a querer modificar la forma en que Apache sirve archivos cuando se solicita un directorio. Actualmente, si un usuario solicita un directorio del servidor, Apache buscará primero un archivo llamado `index.html`. Nosotros queremos decirle a nuestro servidor web que elija los archivos PHP de preferencia, por lo que vamos a hacer Apache busque un archivo `index.php` primero.

Para ello, escribe este comando para abrir el archivo `dir.conf` en un editor de texto con privilegios de root:

```
sudo nano /etc/apache2/mods-enabled/dir.conf
```

Se verá de forma similar a esto:

```
<IfModule mod_dir.c>

    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
    ↪    index.htm

</IfModule>
```

Queremos mover el índice del archivo PHP destacándolo a la primera posición después de la especificación del `DirectoryIndex`, así:

```
<IfModule mod_dir.c>

    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml
    ↪    index.htm

</IfModule>
```

Cuando hayas terminado, guarda y cierra el archivo presionando «CTRL-X». Vas a tener que confirmar el guardado escribiendo «S» y luego pulsando «ENTER» para confirmar la ubicación de almacenamiento de archivos.

Después de esto, tenemos que reiniciar el servidor web Apache para que nuestros cambios sean reconocidos. Puedes hacerlo ejecutando esto:

```
sudo systemctl restart apache
```

En este punto, el LAMP está instalado y configurado. Sin embargo, todavía debemos probar nuestro PHP.

0.5 Paso Cuatro. Prueba del Procesador PHP en el Servidor Web

Con el fin de probar que nuestro sistema se ha configurado correctamente para PHP, podemos crear un script PHP muy básico.

Vamos a llamar a este script `index.php`. Para que Apache pueda buscar el archivo y lo trabaje correctamente, se debe guardar en un directorio muy específico, al cual se le conoce como «raíz».

En Linux, este directorio se encuentra en `/var/www/html/`. Podemos crear el archivo en esa ubicación ejecutando:

```
sudo nano /var/www/html/index.php
```

Esto abrirá un archivo en blanco. Queremos poner el texto siguiente, que es el código PHP válido, dentro del archivo:

```
<?php
phpinfo();
?>
```

Cuando hayas terminado, guarda y cierra el archivo.

Ahora podemos probar si nuestro servidor web puede visualizar correctamente el contenido generado por un script PHP. Para probar esto, sólo tenemos que visitar esta página en nuestro navegador web. De nuevo necesitarás la dirección IP pública del servidor.

La dirección que deseas visitar será:

```
http://127.0.0.1/
```

La página que verás debe ser algo como esto:


PHP Version 7.4.5 	
System	Linux kali 5.7.0-kali1-amd64 #1 SMP Debian 5.7.6-1kali2 (2020-07-01) x86_64
Build Date	Jun 3 2020 20:55:00
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled

Figura 1: PHP

0.6 Paso adicional. Instalación de phpMyAdmin

phpMyAdmin es un software web gratuito para trabajar con MySQL en la web, proporciona un interfaz visual conveniente a las capacidades de MySQL.

```
sudo apt-get install phpmyadmin apache2-utils
```

Durante la instalación, phpMyAdmin nos guiará a través de una configuración básica.

Una vez que el proceso se inicia, seguimos estos pasos:

1. Seleccionamos Apache2 como servidor
2. Seleccionamos Sí cuando se le pregunte si desea configurar la base de datos para phpmyadmin con dbconfig-common
3. Introducimos nuestra contraseña de **MySQL** cuando se le solicite
4. Introducimos nuestra contraseña que deseamos utilizar para iniciar sesión en phpmyadmin

Una vez finalizada la instalación, añadimos phpmyadmin a la configuración de **apache**.


```
sudo nano /etc/apache2/apache2.conf
```

Agregamos la configuración de phpmyadmin al archivo.

```
Include /etc/phpmyadmin/apache.conf
```

Creamos un usuario para conectarnos a MySQL

```
mysql -u root -p // Pedirá la contraseña del usuario root
CREATE USER 'phpmyadmin'@'localhost' IDENTIFIED BY 'sa';
GRANT ALL PRIVILEGES ON *.* TO 'phpmyadmin'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Reiniciar **apache**:

```
sudo systemctl restart apache2
```

A continuación, podemos acceder a phpmyadmin accediendo a <http://localhost/phpmyadmin> e introduciremos el usuario (phpmyadmin) y contraseña (sa).

Nota.

Nunca instalaremos phpMyAdmin en producción o si lo hacemos debemos hacerlo de [forma segura](#)

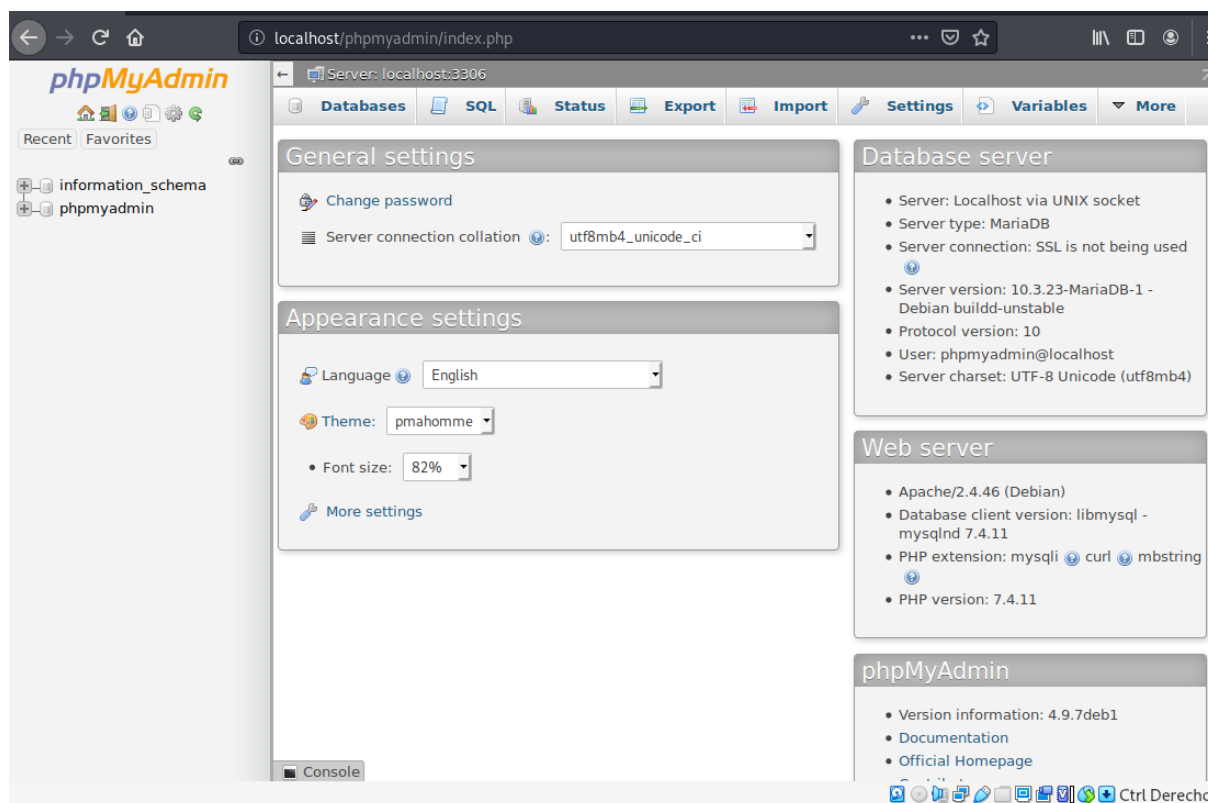


Figura 2: phpMyAdmin

Basado en los siguientes documentos

<https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-on-ubuntu-12-04>

<https://computingforgeeks.com/how-to-install-mysql-on-kali-linux/>