

Cómo instalar la pila Linux, Apache, MySQL y PHP (LAMP) en Ubuntu

Introducción

Una pila “LAMP” es un conjunto de aplicaciones de software de código abierto que se suelen instalar juntas para que un servidor pueda alojar aplicaciones y sitios web dinámicos escritos en PHP. Este término es en realidad un acrónimo que representa al sistema operativo Linux, con el servidor web Apache. Los datos del sitio se almacenan en una base de datos MySQL y el contenido dinámico se procesa mediante PHP.

En esta guía, instalaremos una pila LAMP en un servidor Ubuntu 20.04 o superior.

Paso 1: Instalar Apache y actualizar el firewall

El servidor web Apache está entre los más populares del mundo. Está bien documentado, tiene una comunidad de usuarios activa y ha sido muy utilizado durante gran parte de la historia de la web, por lo que es una excelente opción predeterminada para alojar sitios web.

Instale Apache usando el administrador de paquetes de Ubuntu, apt:

```
sudo apt update
sudo apt upgrade
sudo apt install apache2
```

Si es la primera vez que utiliza sudo en esta sesión, se le pedirá que proporcione su contraseña de usuario para confirmar que tenga los privilegios adecuados para administrar los paquetes del sistema con apt. También se le solicitará que confirme la instalación de Apache al pulsar **Y** y **ENTER**.

Una vez que la instalación se complete, deberá ajustar la configuración de su firewall para permitir tráfico HTTP y HTTPS. UFW tiene diferentes perfiles de aplicaciones que puede aprovechar para hacerlo. Para enumerar todos los perfiles de aplicaciones de UFW disponibles, puede ejecutar lo siguiente:

```
sudo ufw app list
```

Verá un resultado como este:

```
Available applications:
Apache
Apache Full
Apache Secure
OpenSSH
```

A continuación, explicamos cada uno de estos perfiles:

- Apache: este perfil abre solo el puerto 80 (tráfico web normal no cifrado).
- Apache Full: este perfil abre los puertos 80 (tráfico web normal no cifrado) y 443 (tráfico TLS/SSL cifrado).
- Apache Secure: este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado).

Por ahora, es mejor permitir conexiones únicamente en el puerto 80, ya que se trata de una instalación nueva de Apache y todavía no tiene un certificado TLS/SSL configurado para permitir tráfico HTTPS en su servidor.

Para permitir tráfico únicamente en el puerto 80 utilice el perfil **Apache**, pero si queremos usar tanto el puerto 80 como el puerto 443 es necesario usar el perfil **Apache Full**, ten en cuenta que también se puede sustituir por **http** (80) o por **https** (443), en nuestro caso vamos a utilizar **Apache Full**:

```
sudo ufw allow "Apache Full"
```

Puede verificar el cambio con el siguiente comando:

```
sudo ufw status
```

Si nos sale que el error: **Status: inactive**

Eso significa que hay que activar **ufw**.

Para activar **ufw**, vamos a ejecutar el siguiente comando:

```
sudo ufw enable
```

Nos indicará que va a eliminar (disrupt) el puerto `ssh`.

Vamos a confirmar con `y`.

Al ejecutar de nuevo el comando:

```
sudo ufw status
```

Ahora si vemos cual es el estado.

```
Status: active
To Action From
--
Apache Full ALLOW Anywhere
Apache Full (v6) ALLOW Anywhere (v6)
```

Para que en el futuro no tengamos problemas con el acceso remoto al equipo vamos a activar el puerto 22 (Open SSH), para ello vamos a ejecutar los siguientes comandos y luego veremos el estado.

```
sudo ufw allow openssh
```

La salida ahora sería:

```
Status: active
To Action From
--
Apache Full ALLOW Anywhere
OpenSSH ALLOW Anywhere
Apache Full (v6) ALLOW Anywhere (v6)
OpenSSH (v6) ALLOW Anywhere (v6)
```

Ahora, se permite tráfico en el puerto 80 a través del firewall.

Puede realizar una verificación rápida para comprobar que todo se haya realizado según lo previsto dirigiéndose a la dirección IP pública de su servidor en su navegador web (consulte la nota de la siguiente sección para saber cuál es su dirección IP pública si no dispone de esta información):

`http://your_server_ip`

Para ello vamos a ir a AWS y comprobar por una parte la configuración de seguridad de las reglas de entrada y por otra parte ver nuestra **IP pública**.

Revisar configuración en AWS

Las reglas de configuración de seguridad de entrada, deben de ser la siguientes: ##### Reglas de seguridad de entrada

Detalles	Estado y alarmas	Novedad	Monitoreo	Seguridad	Redes	Almacenamiento	Etiquetas
▼ Detalles de seguridad							
Rol de IAM		ID del propietario		Hora de lanzamiento			
-		905519122762		Fri May 17 2024 15:19:22 GMT+0200 (hora de verano de Europa central)			
Grupos de seguridad							
sg-00069c9b4840f09cf (LAMP)							
▼ Reglas de entrada							
Filtrar reglas							
Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen			
-	sgr-0a4c7ceef796c02ef	80	TCP	0.0.0.0/0			
-	sgr-068d9fc77a86b47fa	22	TCP	0.0.0.0/0			
-	sgr-0666de9e23feb4639	443	TCP	0.0.0.0/0			

Ip pública

EC2 > Instancias > i-05030d798ab94eb49

Resumen de instancia de i-05030d798ab94eb49 (LAMP Formación) Información

Conectar Estado de la instancia Acciones

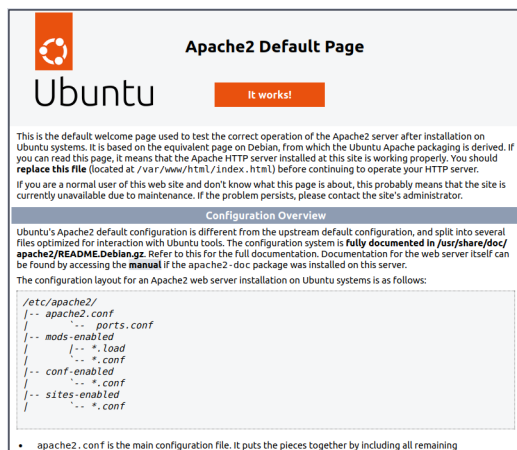
Se ha actualizado hace 14 minutos

ID de la instancia i-05030d798ab94eb49 (LAMP Formación)	Dirección IPv4 pública 3.228.8.192 dirección abierta	Direcciones IPv4 privadas 172.31.5.35
Dirección IPv6 -	Estado de la instancia En ejecución	DNS de IPv4 pública ec2-3-228-8-192.compute-1.amazonaws.com dirección abierta

En este caso la ip pública es: **3.228.8.192**, con lo que si ejecutamos en el navegador de nuestro PC lo siguiente:

`http://3.228.8.192`

Nos debe aparecer la página inicial de nuestro servidor.



Paso 2: Instalar MySQL

Ahora que dispone de un servidor web funcional, deberá instalar un sistema de base de datos para poder almacenar y gestionar los datos de su sitio. MySQL es un sistema de administración de bases de datos popular que se utiliza en entornos PHP.

Una vez más, utilice `apt` para adquirir e instalar este software:

```
sudo apt install mysql-server
```

Cuando se le solicite, confirme la instalación al escribir `Y` y, luego, `ENTER`.

Cuando la instalación se complete, se recomienda ejecutar una secuencia de comandos de seguridad que viene preinstalada en MySQL. Con esta secuencia de comandos se eliminarán algunos ajustes predeterminados poco seguros y se bloqueará el acceso a su sistema de base de datos. Inicie la secuencia de comandos interactiva ejecutando lo siguiente:

```
sudo mysql_secure_installation
```

Se le preguntará si desea configurar el `VALIDATE PASSWORD PLUGIN`.

Elija `Y` para indicar que sí, o cualquier otra cosa para continuar sin la habilitación.

Si responde “sí”, se le solicitará que seleccione un nivel de validación de contraseña. Tenga en cuenta que, si ingresa 2 para indicar el nivel más seguro, recibirá mensajes de error al intentar establecer cualquier contraseña que no contenga números, letras en mayúscula y minúscula, y caracteres especiales, o que se base en palabras comunes del diccionario. Por ello es recomendable en este caso responder `no`.

Para el resto de las preguntas, presione `Y` y `ENTER` en cada mensaje. Con esto, se eliminarán algunos usuarios anónimos y la base de datos de prueba, se deshabilitarán las credenciales de inicio de sesión remoto de root y se cargarán estas nuevas reglas para que MySQL aplique de inmediato los cambios que realizó.

Cuando termine, compruebe si puede iniciar sesión en la consola de MySQL al escribir lo siguiente:

```
sudo mysql
```

Esto permitirá establecer conexión con el servidor de MySQL como root user de la base de datos administrativa, lo que se infiere del uso de sudo cuando se ejecuta este comando. Debería ver el siguiente resultado:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.36-2ubuntu3 (Ubuntu)

Copyright (c) 2000, 2024, 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> 
```

Para salir escribir:

```
mysql> exit;
```

Paso 3: Instalar PHP

Instaló Apache para presentar su contenido y MySQL para almacenar y gestionar sus datos. PHP es el componente de nuestra configuración que procesará el código para mostrar contenido dinámico al usuario final. Además del paquete php, necesitará php-mysql, un módulo PHP que permite que este se comuniquen con bases de datos basadas en MySQL. También necesitará libapache2-mod-php para habilitar Apache para gestionar archivos PHP. Los paquetes PHP básicos se instalarán automáticamente como dependencias.

Para instalar estos paquetes, ejecute lo siguiente:

```
sudo apt install php libapache2-mod-php php-mysql php-mcrypt php-cgi php-curl php-json
```

Una vez que la instalación se complete, podrá ejecutar el siguiente comando para confirmar su versión de PHP:

```
php -v
```

En este punto, su pila LAMP está plenamente operativa, pero, para poder probar su configuración con una secuencia de comandos PHP, lo mejor es instalar un host virtual de Apache adecuado para almacenar los archivos y las carpetas de su sitio web. Lo haremos en el siguiente paso.

3.1 – Comprobación de la instalación de PHP

Ahora que has instalado PHP, vamos a comprobar si funciona correctamente creando un archivo de prueba y abriéndolo en el navegador. Ahora, ingrese el siguiente comando.

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

Este comando abrirá nano editor con un archivo en blanco test.php para su edición. El directorio /var/www/html.

Aquí es donde Apache busca el archivo solicitado en la URL del sitio web de forma predeterminada si no se ha configurado para buscar otro sitio. Consulta la página de documentación de Apache Ubuntu para obtener información sobre su configuración.

Además, necesita privilegios de root para poder escribir en este directorio. Hemos utilizado **sudo** antes de nuestro comando. Ahora ingrese el siguiente texto en el editor abierto:

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


Después de ingresar este texto presione Ctrl + X (o CMD + X si está en Mac), y luego Y, y luego presione ENTER. Esto guardará el archivo y saldrá del editor. Ahora abra la siguiente dirección web en su navegador

```
http://<your_vps_ip_address>/test.php
```

En nuestro caso:

http://3.228.8.192/test.php

El resultado debe ser algo parecido a esta imagen y eso significa que Apache puede procesar ficheros PHP.

PHP Version 8.3.6	
	
System	Linux ip-172-31-5-35 6.8.0-1008-aws #8-Ubuntu SMP Sat Apr 20 00:46:25 UTC 2024 x86_64
Build Date	Apr 15 2024 19:21:47
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-curl.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-gd.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-intl.ini, /etc/php/8.3/apache2/conf.d/20-mbstring.ini, /etc/php/8.3/apache2/conf.d/20-mcrypt.ini, /etc/php/8.3/apache2/conf.d/20-mysql.ini, /etc/php/8.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-posix.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.3/apache2/conf.d/20-sysvsem.ini, /etc/php/8.3/apache2/conf.d/20-sysvshm.ini, /etc/php/8.3/apache2/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831.NTS
PHP Extension Build	API20230831.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*, mcrpyt.*, mdecrypt.*

Paso 4: Cómo instalar PhpMyAdmin

Procederemos a la instalación de PhpMyAdmin. Se trata de una aplicación web que nos permitirá gestionar nuestras bases de datos directamente desde el navegador, algo que resulta práctico y cómodo.

La instalación de PhpMyAdmin es, posiblemente, el paso más complejo de todo el manual, ya que requiere de la intervención del usuario. Por favor, lee atentamente los pasos que detallaremos a continuación: ##### 1. Inicia la instalación

```
sudo apt install phpmyadmin php-mbstring
```

2. Crear contraseña para el usuario root y poder acceder a la BD des phpmyadmin Primero de todo debemos de entrar en la base de datos con el usuario root

```
sudo mysql -u root
```

A continuación cambiamos la contraseña a P@ssw0rd, con el siguiente comando:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd';
exit;
sudo systemctl reload apache2
```

A partir de ahora nuestro usuario / contraseña para entrar en mysql será: root / P@ssw0rd

Paso 5: Comprobar el acceso a PhpMyAdmin

En el navegador abre la siguiente dirección web en su navegador.

http://<your_vps_ip_adress>/phpmyadmin

En nuestro caso:

http://3.228.8.192/phpmyadmin.php

El resultado debe ser algo parecido a esta imagen y eso significa que Apache puede procesar ficheros PHP.




Bienvenido a phpMyAdmin

Idioma (Language)

Español - Spanish



Iniciar sesión 

Usuario:

root

Contraseña:

Iniciar sesión