

## Introducción

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 del sitio son almacenados en base de datos como **M**ariaDB o **M**ySQL y el contenido dinámico es procesado con **P**HP.

En esta guía, vamos a instalar LAMP en Ubuntu 16.04. Ubuntu cumplirá con nuestro primer requisito: un sistema operativo Linux.

## Requisitos Previos

### Usuario con derechos administrativos.

---

Antes de comenzar con esta guía, debes tener una cuenta de usuario independiente que no sea root, con privilegios de sudo configurados en su servidor. La idea es crear un usuario con el que evitemos realizar acciones indeseadas en el sistema. Vamos a ello:

```
#adduser nombredeusuario
```

Donde *nombredeusuario* es el nombre del usuario que vas a crear. (Fuera de un entorno de aprendizaje es siempre recomendable el uso de una buena contraseña).

Dado que la idea es que este usuario adquiera en cierto momento privilegio administrativo. Por defecto, en Ubuntu todo usuario que pertenece al grupo "sudo" le es permitido hacer uso de la palabra sudo antes de cada comando.

```
#usermod -aG sudo nombredeusuario
```

### Ssh.

---

Para comprobar si está instalado:

```
$service ssh status  
$service ssh start  
$apt-cache policy openssh-server
```

Si no está instalado:

```
$sudo apt-get install openssh-server
```

Así, ya podremos conectarnos desde una máquina cliente:

```
$ssh nombredeusuario@ipdelservidor
```

### Ufw (Uncomplicated Firewall)

---

Ufw es una interfaz de línea de comando que consiste en un pequeño número de órdenes simples que nos permite configurar las iptables de nuestro firewall.



**Nota:** Siguiendo el siguiente enlace: [UFW, configura fácilmente tu firewall con iptables](#) activa el firewall con ufw habilitando las siguientes aplicaciones:

```
"Apache Full"  
"Bind9"  
"OpenSSH"
```

# Instalación de Apache

Si aún no tienes instalado Apache, ahora es el momento:

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

Se te indicará qué paquetes se instalarán y cuanto espacio ocuparán en disco.

## Establecer ServerName para Suprimir los Errores de Sintaxis

---

A continuación, agregaremos una línea al archivo `/etc/apache2/apache2.conf` para suprimir un mensaje de advertencia. Si no se define `ServerName` globalmente, recibirá la siguiente advertencia cuando compruebe la configuración de Apache para los errores de sintaxis:

```
$sudo apache2ctl configtest
```

Output

```
AH00558: apache2: Could not reliably determine the server's fully qualified
domain name, using 127.0.1.1. Set the 'ServerName' directive globally to
suppress this message
```

Syntax OK

Abre el archivo de configuración principal con su editor preferido:

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

Dentro, en la parte inferior del archivo, agrega una directiva `ServerName`, apuntando a tu nombre de dominio principal. Si no tienes un nombre de dominio asociado con tu servidor, puedes utilizar la dirección IP del servidor:

```
ServerName dominio_del_servidor_o_IP
```

Ahora podemos revisar los errores de sintaxis:

```
$sudo apache2ctl configtest
```

Puesto que hemos añadido la directiva global `ServerName`, todo lo que debe ver es:

Output

Syntax OK

Reinicie Apache para implementar los cambios:

```
$sudo systemctl restart apache2
```

## Ajustar el Firewall para Permitir el Tráfico Web

---

Ahora, asumiendo que ha seguido las instrucciones previas de configuración del servidor para habilitar el firewall UFW, asegúrate de que el firewall permita el tráfico HTTP y HTTPS. Puedes asegurarte de que UFW tiene un perfil de aplicación para Apache así:

```
$sudo ufw app list
```

Output

Available applications:

```
Apache
Apache Full
Apache Secure
Bind9
OpenSSH
```

Si observas el perfil `Apache Full`, debería mostrar que habilita el tráfico a los puertos 80 y 443:  
`$sudo ufw app info "Apache Full"`

Output

Profile: `Apache Full`

Title: `Web Server (HTTP,HTTPS)`

Description: `Apache v2 is the next generation of the omnipresent Apache web server.`

Ports:

`80,443/tcp`

Permitir el tráfico entrante para ese perfil:  
`$sudo ufw allow in "Apache Full"`

Comprueba ahora desde un navegador desde una máquina cliente que puedes acceder a tu página web.

## Instalación de MariaDB

Sigue el siguiente enlace de [instalación de MariaDB](#). **Nota:** antes de lanzarte a seguir los pasos advierte que la clave para Ubuntu ha cambiado. También ten en cuenta que cuando toque añadir el repositorio adecuado se hace necesario saber la versión Linux instalada:

```
$cat /etc/*-release  
$lsb_release -a
```

Cuando hayas instalado mariaDB, prueba su funcionamiento:

```
$mysql -u root -p  
$MariaDB[(none)]>status
```



**Nota:** Crea usuarios y permite que puedan acceder desde un cliente.

Algunos enlaces de ayuda:

[Usando MariaDB](#)

[Listado de usuarios en mysql](#)

[¿Por qué no me puedo conectar a mi servidor?](#)