# Cómo instalar Apache en Ubuntu 18.04 LTS

En esta entrada veremos **cómo instalar Apache en Ubuntu 18.04 LTS** paso a paso, con lo que tendremos el servidor web más popular de Internet en uno de los sistemas cada vez más utilizados en los servicios **VPS Ubuntu** profesionales.

- Antes de instalar Apache en Ubuntu 18.04 LTS
- Cómo instalar Apache en Ubuntu 18.04 LTS
  - o Cómo configurar el firewall UFW para Apache en Ubuntu 18.04
  - Cómo probar el servicio Apache
  - <u>Cómo saber la versión de Apache instalada</u>
- <u>Cómo configurar Apache en Ubuntu 18.04 LTS</u>
  - <u>Cómo activar módulos de Apache en Ubuntu 18.04</u>
- <u>Cómo crear servidores virtuales de Apache en Ubuntu 18.04 LTS</u>
  - <u>Crear un nuevo servidor virtual de Apache en Ubuntu</u>
  - Error 403 Fobidden en servidores virtuales de Apache en Ubuntu
  - o Configuración adicional de Apache
  - o Cómo desactivar un servidor virtual de Apache en Ubuntu
  - Servidor Virtual por defecto de Apache en Ubuntu
- Conclusión

## Antes de instalar Apache en Ubuntu 18.04 LTS

Conviene que antes de empezar esta **guía de instalación y configuración de Apache en Ubuntu 18.04 LTS** cumplas los siguientes requisitos básicos:

- Una instalación de Ubuntu 18.04 lo más reciente y acualizada posible.
- Un usuario con privilegios de *sudo* para poder instalar paquetes y modificar archivos del sistema.
- Conexión a Internet.

Con todo esto, ¡ya podemos empezar a instalar Apache en Ubuntu 18.04!

# Cómo instalar Apache en Ubuntu 18.04 LTS

Para **instalar Apache en Ubuntu 18.04 LTS** emplearemos los repositorios y réplicas del sistema, así que previamente actualizaremos las listas de paquetes:

• ~\$ sudo apt update

Concretamente instalaremos *apache2*, el paquete principal, y automáticamente se añadirán a la descarga e instalación todas las dependencias necesarias:

• ~\$ sudo apt install -y apache2

El servicio *apache2* queda en ejecución tras la instalación y, además, se configura para iniciar automáticamente en cada arranque de Ubuntu 18.04. Para comprobar que Apache está iniciado usaremos el comando systemctl status apache2:



### Cómo configurar el firewall UFW para Apache en Ubuntu 18.04

Si quieres que el servidor web Apache sea accedido desde la red y tienes activado el firewall UFW en tu sistema Ubuntu 18.04, necesitarás añadir las reglas que permitan las conexiones:

• ~\$ sudo ufw allow http

Si vas a utilizar páginas seguras con conexión SSL tendrás que permitir también el servicio HTTPS:

• ~\$ sudo ufw allow https

### Cómo probar el servicio Apache

Seguro que quieres comprobar el servicio a través de un navegador, así que si la máquina Ubuntu es un sistema de escritorio, abrimos un navegador en la propia máquina, con la dirección *http://l27.0.0.1* ó *http://localhost*.

Si accedemos desde otra máquina (como se hará en este ejemplo) se debe introducir la dirección de la interfaz de red con que se conecta la máquina Ubuntu a la red local, o un dominio que apunte a esa dirección:



Web por defecto de Apache en Ubuntu 18.04, accediedo por IP y por dominio

### Cómo saber la versión de Apache instalada

Para saber qué versión de Apache tenemos instalada en Ubuntu, aunque ya intuirás que se trata de Apache 2 por el nombre del paquete que hemos instalado, usaremos el comando *apache2ctl*:

- ~\$ apachectl -v
- Server version: Apache/2.4.29 (Ubuntu)
- Server built: 2019-09-16T12:58:48

Veremos la versión y revisión de Apache y la fecha de compilación de los binarios.

# Cómo configurar Apache en Ubuntu 18.04 LTS

Al **configurar Apache en Ubuntu 18.04** debemos tener en cuenta que toda la configuración se guarda en el directorio /*etc/apache2*/, siendo /*etc/apache2/apache2.conf* el archivo principal de configuración de Apache.

### Cómo activar módulos de Apache en Ubuntu 18.04

La instalación estándar de Apache en Ubuntu 18.04 viene con muchos módulos con diversas funcionalidades para el servidor web, pero no todos están cargados o activos.

Si quieres saber qué módulos están cargados en tu configuración, lanza el comando apachectl -M:

🛃 chacho@ubuntu64: ~	
chacho@ubuntu64:~\$ apachect1 -M	-
Loaded Modules:	
core_module (static)	
so_module (static)	
watchdog_module (static)	
http_module (static)	
log_config_module (static)	
logio_module (static)	
version_module (static)	
unixd_module (static)	
access_compat_module (shared)	
alias_module (shared)	
auth_basic_module (shared)	
authn_core_module (shared)	
authn_file_module (shared)	
authz_core_module (shared)	
authz_host_module (shared)	
authz_user_module (shared)	
autoindex_module (shared)	
deflate_module (shared)	
dir_module (shared)	
env_module (shared)	
Tilter_module (shared)	
mime_module (shared)	
mpm_pretork_module (shared)	
negotiation_module (shared)	
pnp/_module (shared)	
reqtimeout_module (shared)	
setenvil_module (shared)	
status_modute (snared)	
Chachowubuncuo4:~5	-

Si necesitas algún módulo concreto y no está en esa lista deberás activarlo. Por ejemplo, si quisieras activar el módulo de CGI podrías activarlo con el comando *a2enmod*:

• ~\$ sudo a2enmod cgi

Por el contrario, si tuvieras activo este módulo y quisieras desactivarlo podrías hacerlo con el comando *a2dismod*:

• ~\$ sudo a2dismod cgi

En cualquier caso, tras activar o desactivar módulos, necesitarás reiniciar el servidor Apache:

• ~\$ sudo systemctl restart apache2

### Cómo crear servidores virtuales de Apache en Ubuntu 18.04 LTS

Una de las funcionalidades más interesantes a la hora de configurar Apache en Ubuntu es la de servidores virtuales o Virtual Hosts, que nos permitirá alojar varios sitios web independientes referenciados por nombres de dominio distintos.

En Ubuntu la funcionalidad de servidores virtuales ya viene activada y configurada por defecto. De hecho, la página web que hemos visto es el servidor virtual por defecto, que es el que responde ante cualquier dirección IP o dominio que apunte a este servidor y no tenga configurado un servidor virtual propio.

### Crear un nuevo servidor virtual de Apache en Ubuntu

Para empezar necesitaremos un dominio que apunte al servidor Ubuntu, ya sea en Internet, ya sea en local, según en qué entorno estemos trabajando.

En este tutorial vamos a configurar un servidor virtual para el dominio *blog.ubuntu18.local*, configurado para apuntar a la dirección IP del mismo servidor Ubuntu 18.04 en el que estamos trabajando. Antes de crear el nuevo servidor virtual, esa dirección en un navegador devolverá la página web por defecto que vimos antes:



Dominio sin configurar el servidor virtual de Apache en Ubuntu

Las configuraciones de los servidores virtuales se guardan en archivos independientes con extensión .*conf* en el directorio */etc/apache2/sites-available/*.

Usando nuestro editor de texto favorito, crearemos un archivo en ese directorio. Lo ideal es darle como nombre el dominio al que va a responder, terminando con la extensión.*conf.* Esto es especialmente útil cuando tenemos varios servidores virtuales configurados en el sistema.

El archivo podría ser /etc/apache2/sites-available/blog.ubuntu18.local.conf:

• ~\$ sudo nano /etc/apache2/sites-available/blog.ubuntu18.local.conf

Y podría tener este contenido:

- <VirtualHost \*:80>
- ServerName blog.ubuntu18.local
- DocumentRoot /home/instalador/public\_html/blog.ubuntu18.local
- LogLevel info
- ErrorLog \${APACHE\_LOG\_DIR}/blog.ubuntu18.local-error.log
- CustomLog \${APACHE\_LOG\_DIR}/blog.ubuntu18.local-access.log combined
- </VirtualHost>

En la directiva ServerName indicamos el dominio al que responde el nuevo servidor virtual.

En la directiva *DocumentRoot* indicamos el directorio en el que residirán los archivos del sitio web. Un buen sitio para crear estos directorios podría ser */var/www/*, pero podría ser interesante y cómodo tenerlos en el directorio personal del usuario que mantiene el sitio web. Es lo que haremos en este ejemplo:

• ~\$ mkdir -p public\_html/blog.ubuntu18.local

Pondremos algún contenido para el sitio web, como el archivo índice por defecto (*index.htm*l):

• ~\$ nano public\_html/blog.ubuntu18.local/index.html

El contenido de podría ser un simple mensaje identificativo:

• <h1>Bienvenido a blog.ubuntu18.local</h1>

Para activar el nuevo servidor virtual tendremos que usar el comando a2ensite:

• ~\$ sudo a2ensite blog.ubuntu18.local.conf

A continuación, se aplicarán los cambios recargando la configuración de Apache con *systemctl*:

• ~\$ sudo systemctl reload apache2

Ya estaríamos en condiciones de probar el sevidor virtual recargando la página web a la que accedimos hace un momento:



Error 403 Forbidden. Acceso no permitido al directorio web

Un momento, ¿qué ha ocurrido?

### Error 403 Fobidden en servidores virtuales de Apache en Ubuntu

La configuración por defecto de Apache en Ubuntu no permite el acceso a contenido web fuera de */var/www/*, pero en este ejemplo sí lo hemos configurado fuera, concretamente bajo */home/*. De ahí el error 403.

Para configurar servidores virtuales debemos permitir expresamente el acceso a ubicaciones alternativas dentro del archivo de configuración del servidor virtual.

Los cambios que haremos serán añadir un bloque *Directory* con una directiva *Require* en el archivo de configuración del servidor virtual, en este ejemplo /*etc/apache2/sites-available/blog.ubuntu18.local.conf*:

- <VirtualHost \*:80>
- ServerName blog.ubuntu18.local
- DocumentRoot /home/instalador/public\_html/blog.ubuntu18.local
- •
- <Directory /home/instalador/public\_html/blog.ubuntu18.local>
- Require all granted
- </Directory>
- •

- LogLevel info
- ErrorLog \${APACHE\_LOG\_DIR}/blog.ubuntu18.local-error.log
- CustomLog \${APACHE\_LOG\_DIR}/blog.ubuntu18.local-access.log combined
- </VirtualHost>

Recargamos la configuración de Apache con *systemctl* y probamos de nuevo a acceder a la página web del servidor virtual:



Servidor virtual de Apache configurado y funcionando

### Configuración adicional de Apache

Si borramos el archivo *index.html* del directorio web del servidor virtual, obtendríamos un error 403 al intentar acceder desde el navegador. Esto ocurre porque por defecto no se pemite listar el contenido de directorios web.

Para modificar este comportamiento y permitir el listado, habría que añadir una directiva *Options* en el bloque *Directory* del archivo de configuración del servidor web:

- <Directory /home/instalador/public\_html/blog.ubuntu18.local>
- Options +Indexes
- Require all granted
- </Directory>

Tras recargar la configuración de Apache, si accedemos al directorio vacío esta vez sí recibiremos una página de listado elaborada por el servidor web:



Este comportamiento en general no es deseable, ya que se expone a la vista de cualquiera la estructura del sitio, archivos que podrían dar más información de la deseada, etc.

Sin embargo podríamos querer tener un directorio donde sí se listara el contenido, como una zona de descargas.

Para lograr esta configuración modificaremos de nuevo el bloque *Directory* del servidor virtual:

- <Directory /home/instalador/public\_html/blog.ubuntu18.local>
- Options -Indexes
- Require all granted
- AllowOverride all
- </Directory>

Eliminamos la opción de indexar o, mejor aún, la desactivamos explícitamente. Y añadimos la directiva *AllowOverride* que permite la modificación de la configuración de Apache mediante archivos *.htaccess*.

Recargamos la configuración de Apache y creamos un directorio nuevo, por ejemplo, *descargas/* dentro de la estructura del sitio web:

• ~\$ mkdir public\_html/blog.ubuntu18.local/descargas

Dentro de este directorio crearemos un archivo .htaccess:

• ~\$ nano public\_html/blog.ubuntu18.local/descargas/.htaccess

Con este contenido:

• Options +Indexes

Para los cambios en los archivos *.htaccess* no es necesario recargar la configuración de Apache, porque la configuración se modifica al vuelo, cada vez que se accede al directorio que los contiene.

Ahora tenemos vacíos de contenido web tanto el directorio principal del sitio web como el directorio *descargas/*. La configuración del servidor virtual por defecto no permite el listado de directorios sin archivo índice, pero en el directorio *descargas/* sí se permite gracias al archivo *.htaccess.* 

Accediendo tanto a la raíz del sitio como al directorio *descargas*/ comprobamos el comportamiento personalizado:



Configuración personalizada mediante archivos .htaccess

Justo lo que queríamos.

### Cómo desactivar un servidor virtual de Apache en Ubuntu

Igual que para activar un servidor virtual en Apache tenemos un comando, para desactivarlo tenemos otro, el comando *a2dissite*:

• ~\$ sudo a2dissite blog.ubuntu18.local.conf

Al igual que para el caso de la activación, debemos recargar la configuración de Apache para que los cambios tomen efecto y el servidor virtual no sea accesible.

Si el dominio del servidor virtual que acabamos de desactivar sigue apuntando a la dirección IP del servidor Ubuntu, se servirá la página web por defecto.

### Servidor Virtual por defecto de Apache en Ubuntu

La configuración de Apache en Ubuntu tal y como viene tras la instalación incluye un servidor virtual por defecto. Su archivo de configuración es /*etc/apache2/sites-available/000-default.conf*, donde puedes hacer los cambios que creas necesarios. El directorio web de este servidor es /*var/www/html/*.

Como ya he dicho anteriormente, este servidor virtual responde a las conexiones a través de dirección IP y de cualquier dominio que se resuelva a la dirección IP del servidor y no tengan configurado su propio servidor virtual.

# Conclusión

Ahora que sabes cómo instalar y configurar Apache en Ubuntu 18.04 LTS con servidores virtuales incluidos puedes alojar varios sitios en una misma máquina, lo que hará que rentabilices mucho más los recursos de la máquina Ubuntu. Esto es especialmente interesante si usas un **VPS Ubuntu** de algún servicio profesional.

Si tienes alguna pregunta, indicar algún error o corrección, o tienes sugerencias para otros temas, déjame un comentario.