

APACHE

Javier Hermoso



RESUMEN

En este documento se va a hablar sobre el servidor HTTP **Apache** de una manera clara para que sea entendido por usuarios primerizo. En primer lugar se redactará un poco sobre su historia y origen, seguidamente cual es el propósito de **Apache**.

Seguidamente se explicará como instalar **Apache** en nuestro sistema **Ubuntu** indicando detalladamente los pasos a seguir para evitar cualquier tipo de errores.

Por último la guía de **configuración** para poder iniciar nuestro servidor con nuestra propia web en local, pasando antes por la configuración de puertos y de la **URL** personalizada.

PALABRAS CLAVE

- Servidor
- HTTP
- Linux
- C
- WWW

ÍNDICE

1. Introducción
 - 1.1. ¿Qué es Apache?
 - 1.2. Historia
2. Tutorial
 - 2.1. Instalación

- 2.1.1. Instalar Apache
- 2.1.2. Configurar cortafuegos Apache
- 2.1.3. Configurar cortafuegos Apache
- 2.1.4. Verificar servidor web.

2.2. Configuración

3. Conclusión

4. Bibliografía

1. INTRODUCCIÓN

1.1 ¿QUÉ ES APACHE?

Apache es un servidor web de código abierto, multiplataforma y gratuito. Este web server es uno de los más utilizados en el mundo, actualmente el 43% de los sitios webs funcionan con él.

1.2 HISTORIA

En marzo de 1989 [Tim Berners Lee](#), un científico británico que trabajaba en el [CERN](#). Berners propone una nueva forma de gestionar toda la información de los proyectos del **CERN**.

Este se trataba de una red de documentos enlazados la **WWW**. Sin embargo no fue hasta 1996 cuando ya se estrenó la versión 1.0 de Apache la cual ya incluyó funciones muy interesantes como [MySQL](#) que apareció en la **1.1**.

El nombre de Apache se debe a la filosofía de desarrollo y de organización. Al igual que la tribu de los apaches.

2. TUTORIAL

El siguiente tutorial nos va a explicar como instalar el servidor Apache en nuestro sistema **Ubuntu**.

2.1. INSTALACIÓN

Para poder instalar Apache vamos a tener que ejecutar los siguientes comandos en nuestro terminal.

Instalar Apache

```
sudo apt update
```

```
sudo apt install apache2
```

Configurar cortafuegos

```
sudo ufw app list
```

Una vez ingresado ese comando nos deberá aparecer un listado como este:

```
Available applications:
Apache
```

```
Apache Full
Apache Secure
OpenSSH
```

Ahora lo que tenemos que hacer es habilitar el Apache con este código:

```
sudo ufw allow 'Apache'
```

Y para comprobar que todo esté correcto:

```
sudo ufw status
```

Si todo ha salido bien la salida deberá de ser la siguiente:

```
Status: active
```

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
Apache	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)

Verificar el servidor web

Para saber si Apache está funcionando podremos ejecutar lo siguiente:

```
sudo systemctl status apache2
```

Como podremos ver, en la 5ª línea debe de aparecer *active (running)*.


```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
   enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Tue 2018-04-24 20:14:39 UTC; 9min ago
   Main PID: 2583 (apache2)
     Tasks: 55 (limit: 1153)
    CGroup: /system.slice/apache2.service
            └─2583 /usr/sbin/apache2 -k start
              └─585 /usr/sbin/apache2 -k start
                └─2586 /usr/sbin/apache2 -k start
```

Si todo ha funcionado correctamente ahora podremos acceder a nuestra web con el comando:

```
hostname -I
```

Y deberemos escribir esa IP en el navegador .

Veremos algo parecido a esto:



Apache2 Ubuntu Default Page

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
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to any file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

- *Apagar Apache:*

```
sudo systemctl stop apache2
```

- *Iniciar Apache:*

```
sudo systemctl start apache2
```

- *Detener y reiniciar Apache:*

```
sudo systemctl restart apache2
```

- *Impedir que Apache se inicie cuando enciendes el servidor:*

```
sudo systemctl disable apache2
```

- *Permitir que Apache se inicie cuando enciendes el servidor:*

```
sudo systemctl enable apache2
```

2.1. CONFIGURACIÓN

Primero que todo deberemos crear una carpeta con el nombre **gci** dentro de la siguiente ruta

```
/var/www/
```

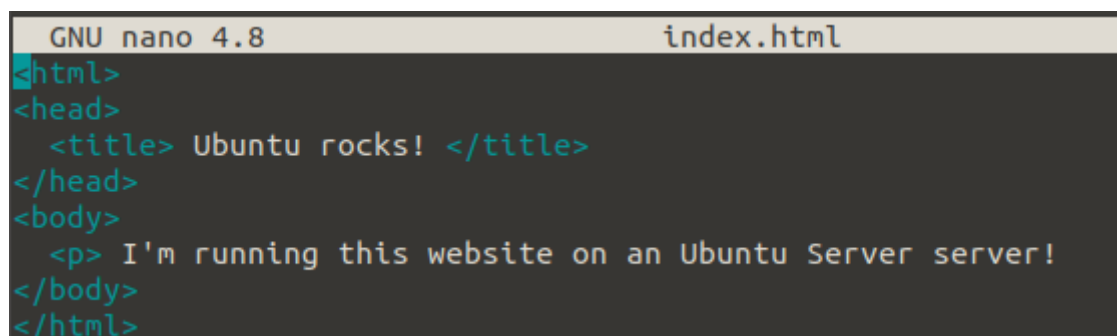
Una vez creada deberemos colocarnos dentro de ella con el siguiente comando:

```
cd /var/www/gci
```

Una vez dentro, deberemos de crear el archivo **index.html** de la siguiente manera:

```
nano index.html
```

Y escribir algo como esto:



```
GNU nano 4.8 index.html
<html>
<head>
  <title> Ubuntu rocks! </title>
</head>
<body>
  <p> I'm running this website on an Ubuntu Server server!
</body>
</html>
```

A continuación queremos editar el archivo **gci.conf**, pero para ello primero debemos dirigirnos a la ruta donde se encuentra:

```
cd /etc/apache2/sites-available/
```

Una vez ahí queremos copiar un archivo y cambiarle el nombre a **gci.conf**:

```
cp 000-default.conf gci.conf
```

Ahora para lo deberemos editar con la instrucción **nano**.

```

root@usuario-OptiPlex-380: /etc/apache2/sites-available
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 4.8 gci.conf
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin monpersonalth@gmail.com
DocumentRoot /var/www/gci/
ServerName gci.example.com

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

Deberemos cambiar el **ServerAdmin** y poder nuestro mail.

Seguidamente deberemos iniciar la configuración:

```
a2ensite gci.conf
```

Y el resultado debe de ser este:

```

Enabling site gci.
To activate the newq configuration, you need to run:
Systemctl reload apache2

```

Le hacemos caso y ejecutamos el ultimo comando.

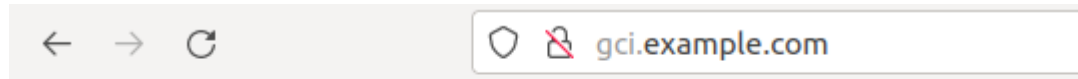
Sin embargo esto no funciona todavía, tenemos que incluir nuestra URL dentro del fichero **hosts** el cual se encuentra en la carpeta **etc**:

```

Abrir hosts Guardar
/etc
1 127.0.0.1 localhost
2 127.0.1.1 usuario-OptiPlex-380
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1 ip6-localhost ip6-loopback
6 fe00::0 ip6-localnet
7 ff00::0 ip6-mcastprefix
8 ff02::1 ip6-allnodes
9 ff02::2 ip6-allrouters
10 127.0.0.1 localhost.localdomain localhost
11 ::1 localhost6.localdomain6 localhost6
12 127.0.0.1 gci.example.com

```

Ahora introducimos la URL en nuestro navegador y si todo funciona veremos algo así:



I'm running this website on an Ubuntu Server server!

Para quitar la configuración y volver al estado inicial debermos escribiri lo siguiente y posteriormente reiniciar el PC:

```
a2disite gci.conf
```

3. CONCLUSIÓN

Para concluir, en mi opinión, aún estando explicado detalladamente todo el proceso de instalación y configuración de Apache, puede llegar a ser algo engorroso para un usuario medio, por lo tanto solo recomiendo realizar esto si eres un usuario avanzado y sabes solucionar problemas dentro del entorno de Linux, si no es ese el caso, recomiendo la instalación de algún paquete como **XAMP** que incluyó todo lo que vamos a necesitar para el servidor de nuestra modesta web.

4. BIBLIOGRAFÍA

- [QUÉ ES APACHE](#)
- [HISTORIA APACHE](#)
- [IMÁGENES DE CONFIGURACIÓN CEDIDAS POR RAÚL CUBERO DEBIDO A PROBLEMAS TÉCNICOS](#)