

0.- Servidores Web

Un servidor web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de petición, el servidor web busca una página web o bien ejecuta un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

El **servidor HTTP Apache** es un servidor HTTP de software libre para plataformas Unix (BSD, GNU/Linux, ...), Microsoft Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y el concepto de sitio virtual.

La versión 2.x de Apache tiene varias mejoras claves sobre Apache 1.x. Estas mejoras incluyen threads de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPv6.

NGINX es un servidor web HTTP (segunda alternativa más utilizada) de código abierto que incluye servicios de correo electrónico con acceso al Internet Message Protocol (IMAP) y al servidor Post Office Protocol (POP). Además, NGINX está listo para ser utilizado como un proxy inverso. En este modo, se utiliza para equilibrar la carga entre los servidores back-end, como también para ser utilizado como caché en un servidor back-end lento.

Su arquitectura, es diferente al modelo tradicional, de crear una instancia por cada request. NGINX procesa decenas de miles de conexiones simultáneas en un proceso compacto y con varios núcleos de CPU.

Además se compone de módulos que se incluyen en tiempo de compilación. Eso significa que el usuario descarga el código fuente y selecciona qué módulos quiere utilizar, haciendo más liviano que la competencia. Hay módulos para la conexión a clones de aplicaciones, balanceo de carga, servidor proxy, y otros. No hay módulo para PHP, ya que Nginx puede interpretar código PHP en sí mismo.

1.- INSTALACIÓN LAMP

En las tecnologías **LAMP** esto se consigue mediante la unión de las siguientes tecnologías:

- **Linux**, el sistema operativo; En algunos casos también se refiere a LDAP.
- **Apache**, el servidor web;
- **MySQL**, el gestor de bases de datos;
- **Perl, PHP, o Python**, los lenguajes de programación.

Para desinstalar un paquete Linux

Apt-get remove (Borrar todos los paquetes instalados)

Apt-get purge (Borrar todos los paquetes instalados y sus ficheros de configuración)

Vamos a instalar los paquetes necesarios para tener un entorno LAMP (Permiso de administración).

Antes de instalar se recomienda

apt-get update (actualizar repositorios)

apt-get upgrade (actualizar el sistema)

✓ Apache: **apt-get install apache2**

Además del paquete apache2, se instalarán los siguientes paquetes:

apache2-common apache2-mpm-worker apache2-utils libapr0 ssl-cert

✓ MySQL: **apt-get install mysql-common mysql-client mysql-server**

Durante la instalación del servicio se nos pedirá la contraseña del usuario root del servidor mysql.

✓ PHP: **apt-get install php libapache2-mod-php php-mysql**

✓ Como gestor de la base de datos podemos utilizar phpmyadmin, que es una aplicación php que me permite acceder a todas las funciones de MySQL:

apt-get install phpmyadmin

En la instalación del paquete phpmyadmin nos preguntará el servidor web que estamos utilizando, para realizar la configuración automática. En nuestro caso escogeremos la opción Apache2 (señalando la opción con la barra espaciadora).

2.- CONFIGURACIÓN

2.1 Ficheros de configuración de apache

- **/etc/apache2/apache2.conf** Es el fichero de configuración principal del servidor. Por medio de directivas incluye en este fichero los módulos necesarios y el resto de la configuración de los restantes ficheros de configuración.
- **/etc/apache2/httpd.conf** Este fichero se mantiene por razones de compatibilidad con la primera versión de Apache. Además se recomienda que la configuración de los módulos de terceros vayan en este fichero.
- **/etc/apache2/sites-available** Es el directorio donde se tienen los ficheros de configuración de los diferentes sitios virtuales. El fichero default-000 contiene la configuración del sitio por defecto.
- **/etc/apache2/sites-enabled** Es un directorio donde existe enlaces simbólicos a los ficheros de configuración de los sitios virtuales (en el directorio /etc/apache2/sites-available) que están activos.

- **/etc/apache2/ports.conf** Es el fichero de configuración donde se indica el puerto por el que escucha el servidor y en su caso la dirección IP,
- inicialmente viene configurado para escuchar sólo en el 80/tcp.
- **/etc/apache2/mods-available** Directorio que aloja todos los módulos instalados en el servidor.
- **/etc/apache2/mods-enabled** Directorio que aloja mediante enlaces simbólicos al directorio anterior, todos los módulos activos en el servidor.

Las líneas que comienzan con el símbolo # en cualquiera de los ficheros de configuración, son comentarios y el servidor no las tiene en cuenta.

La configuración del servidor por defecto define un host virtual default cuyo directorio de trabajo es **/var/www/html**, por lo tanto para probar que el servidor está instalado podemos poner en un navegador **http://IP**, con lo que se visualizará la página web guardada en este directorio.

El fichero index.php (plataforma) os puede servir para comprobar que funciona el PHP en el directorio **/var/www/html** con el siguiente contenido:

También puedes editarlo con nano con este contenido:
`<?php phpinfo(); ?>`

2.2 Host virtual (no seguro - HTTP)

Si nuestro servidor dispone de varios portales web diferentes (Ej Web y drupal) lo guardaremos la página web en **/var/www/html** y el drupal en la ruta **/var/www/drupal** y accederemos a través de dominios diferentes (**host virtuales**).

En los ficheros

-> **/etc/apache2/sites-available/000-default.conf** (No seguro – HTTP 80)
 están configurado la ruta por defecto para todas las páginas web **/var/www/html** que **NUNCA** deberemos modificar si no queremos cambiar la ruta por defecto. A través de los **HOST VIRTUALES** accederemos a cada sitio web almacenados en carpetas diferentes

- Tenemos que comprobar que en el fichero **/etc/apache2/ports.conf** aparecen los puertos por los que queremos que se acceda a nuestro servidor, los puertos por defectos son: **80**.
1. Comprobar que los ficheros del sitio web este dentro de una **carpeta** en **/var/www/html/** y tiene los permisos adecuados
 2. Crear nuestro fichero de configuración (**/etc/apache2/sites-available**) para nuestro sitio web (ej. drupal): copiando el fichero por defecto con otro nombre (Ej drupal.conf).

```
cp 000-default.conf drupal.conf
```

Por cada carpeta del sitio web tenemos un fichero de configuración en sites-available

Las páginas web se alojarán dentro de

/var/www/misitio → /etc/apache2/sites-available/misitio.conf

/var/www/misitio2 → /etc/apache2/sites-available/misitio2.conf

/var/www/drupal → /etc/apache2/sites-available/drupal.conf

.....

Cada carpeta y ficheros del sitio web tienen que tener los siguientes permiso (permiso usuario root y usuario apache)

chmod -R 771 /var/www/misitio

chown -R www-data:www-data /var/www/misitio

ls -all

- Modificar el fichero .conf (sites-availables) del nuevo host virtual que hemos creado

```
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port to
# 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 webmaster@localhost
DocumentRoot /var/www/html
<Directory /var/www/html>
# 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

</Directory>
</VirtualHost>
```

- Habilitamos el nuevo sitio que acabamos de crear :

```
a2ensite drupal.conf
```

(nota: el comando a2ensite es equivalente a

cp /etc/apache2/sites-available/drupal.conf /etc/apache2/sites-enabled)

Comprueba que el fichero esté en la carpeta sites-enabled

- Reiniciamos el servidor Apache2:

```
systemctl restart apache2.service
```

- Configurar el archivo hosts cliente o configurar un servidor DNS para el nombre de dominio Ej. (drupal.dominio.com)

Debe existir un servidor DNS que asocie la IP correcta al nombre del servidor o en el **ordenador del cliente** debería aparecer en el archivo

Windows 7: C:\Windows\System32\drivers\etc\hosts (abrir wordpad en modo administrador)

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
ip drupal.dominio.com
```

→ Pon IP y el dominio conectado

2.3 Host virtual (seguro -HTTPS)

En los ficheros

-> /etc/apache2/sites-available/default-ssl.conf (Seguro – HTTPS 443)

- Tenemos que comprobar que en el fichero /etc/apache2/ports.conf aparecen los puertos por los que queremos que se acceda a nuestro servidor, los puertos por defecto son: **443 (web segura)**.

```
GNU nano 2.2.6      Fichero: /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf
Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

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

[ 15 líneas leídas ]
^G Ver ayuda  ^O Guardar   ^R Leer Fich ^Y Pág Ant   ^K CortarTxt ^C Pos actual
^X Salir      ^J Justificar ^W Buscar    ^V Pág Sig   ^U PegarTxt  ^T Ortografía
```

Crear nuestro fichero de configuración para nuestro sitio web (ej. drupal):

Los pasos previos para instalar nuestro CMS (Ej drupal) de manera segura consiste en crear primero el certificado y después crear el fichero de configuración de manera parecida al host virtual no seguro

Generar certificado

- Instalar openssl

```
apt-get install openssl
```

- Comprobaremos si está o no instalado consultando el directorio **/etc/apache2/mods-enabled**, y en el caso de que no esté lo habilitaremos con:

```
a2enmod ssl
```

Reiniciamos el servidor Apache2:

```
systemctl restart apache2.service
```

- Creamos el directorio dónde almacenaremos los certificados y generamos el certificado

```
mkdir -p /etc/ssl/certificados
```

```
openssl req -new -x509 -days 365 -nodes -out  
/etc/ssl/certificados/apache.pem -keyout  
/etc/ssl/certificados/apache.key
```

- Respondemos a las preguntas que se nos hacen en la generación del certificado (no pongas nombre adicional).
- Damos permisos de lectura al certificado

```
chmod 600 /etc/ssl/certificados/apache*
```

Configurar nuestro host virtual

- Crear nuestro fichero de configuración (**/etc/apache2/sites-available**) para nuestro sitio web (ej. drupal): copiando el fichero por defecto con otro nombre (Ej drupal.conf).

```
cp default-ssl.conf drupalseguro.conf
```

- Busca las líneas siguientes y modifica o añade las que necesites

```

<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/drupal
    ServerName drupal.dominio.com
    DirectoryIndex index.html index.php
    SSLEngine on
    SSLCertificateFile /etc/ssl/certificados/apache.pem
    SSLCertificateKeyFile /etc/ssl/certificados/apache.key

    # Available loglevels: trace8, ..., trace1, debug, info, notices
    # 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
    [ 140 líneas leídas ]

```

NUEVAS [

Añade el dominio
Añade la carpeta

Nota: Busca información de cada línea que no entiendas

- Habilitamos el nuevo sitio que acabamos de crear:

```
a2ensite drupalseguro.conf
```

(nota: el comando **a2ensite** es equivalente a **cp /etc/apache2/sites-available/drupalseguro.conf /etc/apache2/sites-enabled**)

- Reiniciamos el servidor Apache2:

```
systemctl restart apache2.service
```

- Configurar el archivo hosts cliente o configurar un servidor DNS para el nombre de dominio Ej. (drupal.dominio.com)

Debe existir un servidor DNS que asocie la IP correcta al nombre del servidor o en el **ordenador del cliente** debería aparecer en el archivo

Windows 7: C:\Windows\System32\drivers\etc\hosts (abrir wordpad en modo administrador)

```

# localhost name resolution is handled within DNS itself.
# 127.0.0.1 localhost
# ::1 localhost
ip drupal.dominio.com

```

→ Pon IP y el dominio correcto

Cliente

← → ↻ ⚠ No es seguro | https://drupal.dominio.com

