MANUAL INSTALACIÓN LAMP

Miguel Ángel Aranda García

INSTALACIÓN GENERAL

CML-WX	
Sistema Operativo	Ubuntu Server 20.04
RAM	2GB
Tamaño del disco	80GB
Particiones	/ 40GB ext4
	/var 35.997GB ext4
	swap 4GB
	1M BIOS Boot
Usuarios	miadmin/paso
	operadorweb/paso
Configuración de red	IP : 192.168.1.200
	GATEWAY : 192.168.1.1
	MASCARA : 255.255.255.0
	DNS: 192.168.1.1 - 8.8.8.8 - 8.8.4.4

Configuración de red

¡ATENCIÓN!

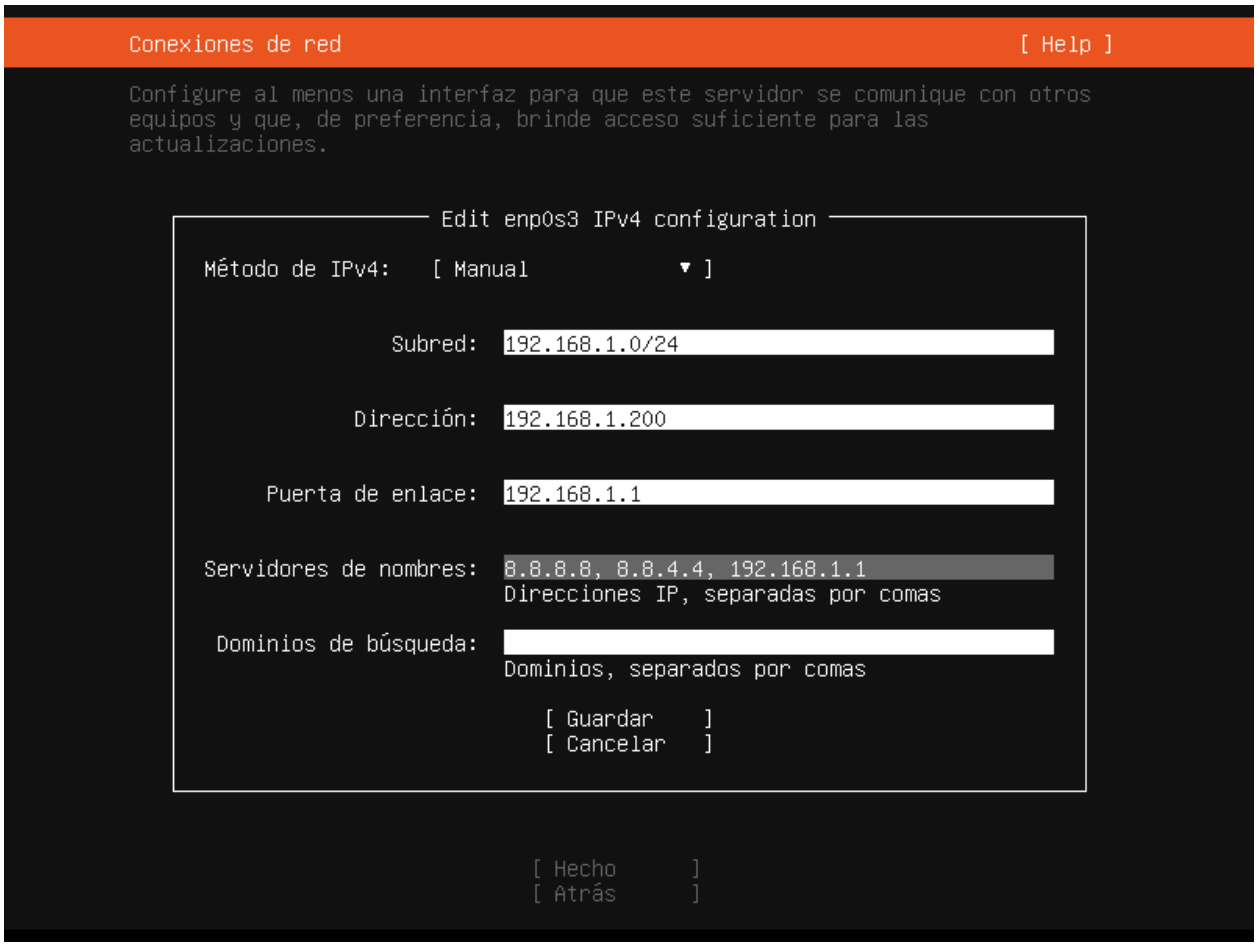
En caso de hacerse en una máquina virtual es necesario establecer una conexión puente para que la máquina sea detectada en la red local. Estos ajustes se han hecho en una máquina virtual con dicha conexión.

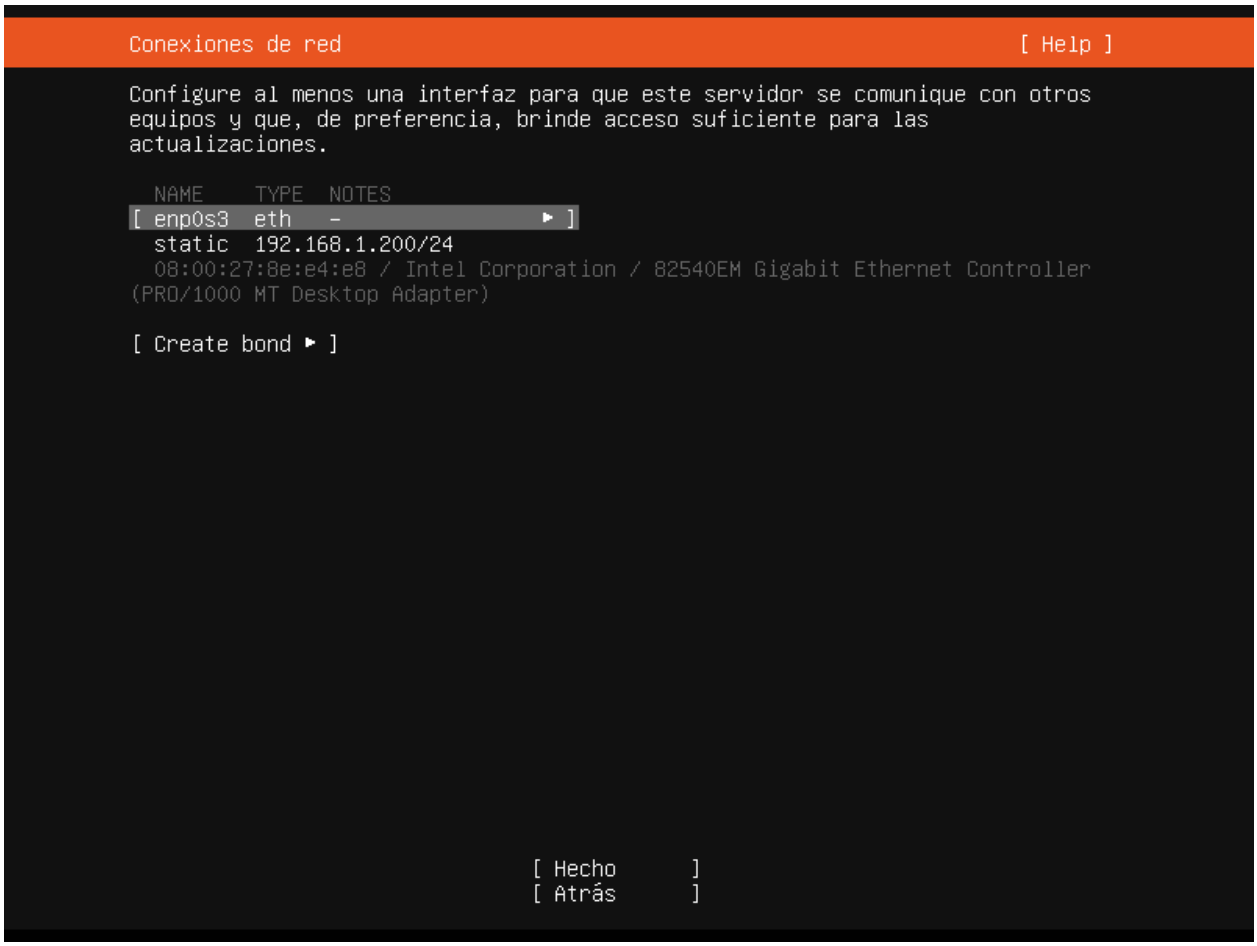
La subred corresponde a la dirección de red, la cual por defecto suele ser 192.168.x.0.

La dirección será la dirección que nosotros queramos darle a la máquina.

La puerta de enlace (normalmente la dirección del router), es la dirección del dispositivo por la cual se conecta nuestro equipo a internet.

Los servidores de nombres serán las direcciones IP de los servidores donde se hará la resolución de nombres (Servidores que buscan direcciones IP de los servidores mediante su dominio de internet).

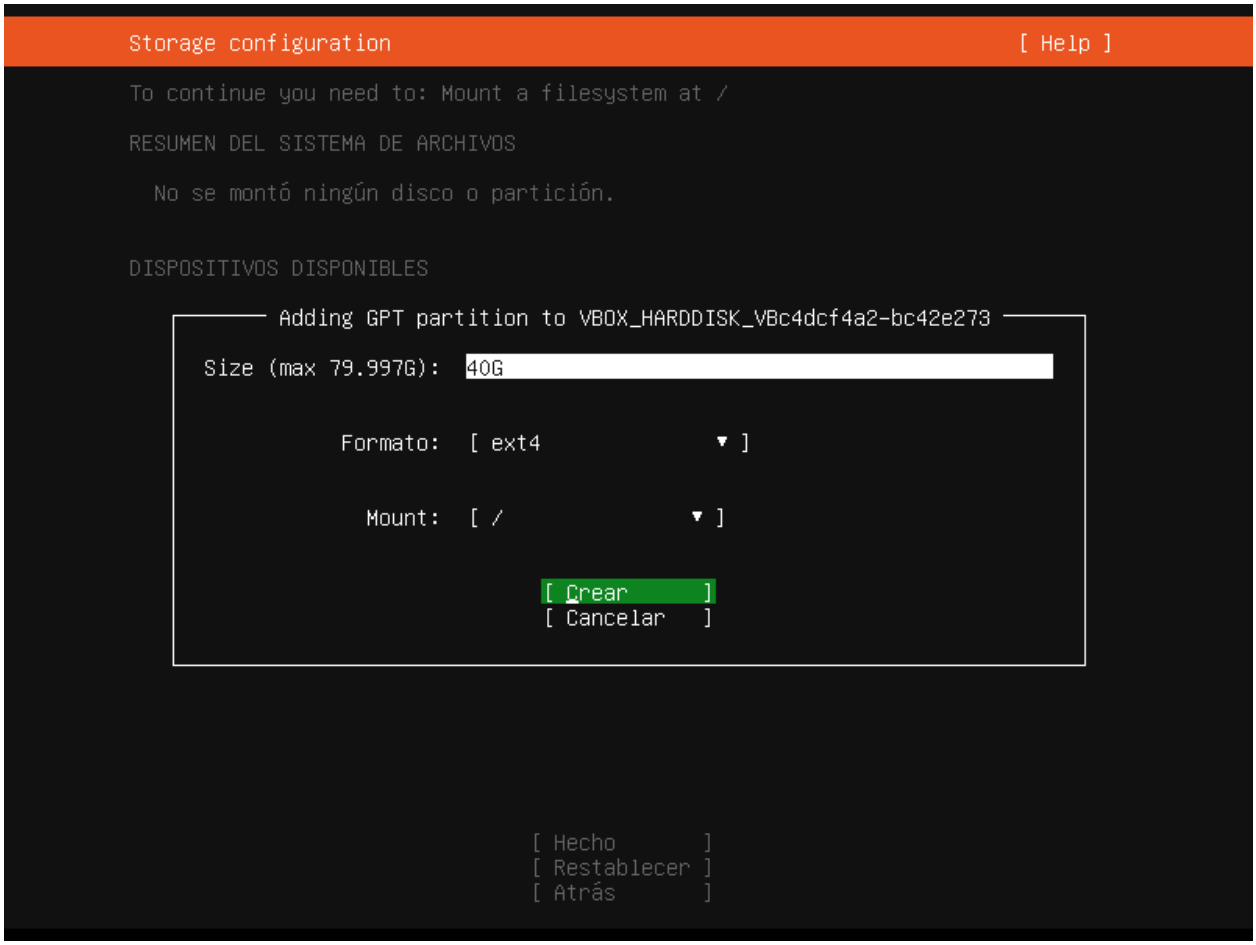

Conexiones de red
[Help]
Configure al menos una interfaz para que este servidor se comunique con otros equipos y que, de preferencia, brinde acceso suficiente para las actualizaciones.
Edit enp0s3 IPv4 configuration
Método de IPv4: [Manual ▾]
Subred: 192.168.1.0/24
Dirección: 192.168.1.200
Puerta de enlace: 192.168.1.1
Servidores de nombres: 8.8.8.8, 8.8.4.4, 192.168.1.1
Direcciones IP, separadas por comas
Dominios de búsqueda:
Dominios, separados por comas
[Guardar]
[Cancelar]
[Hecho]
[Atrás]

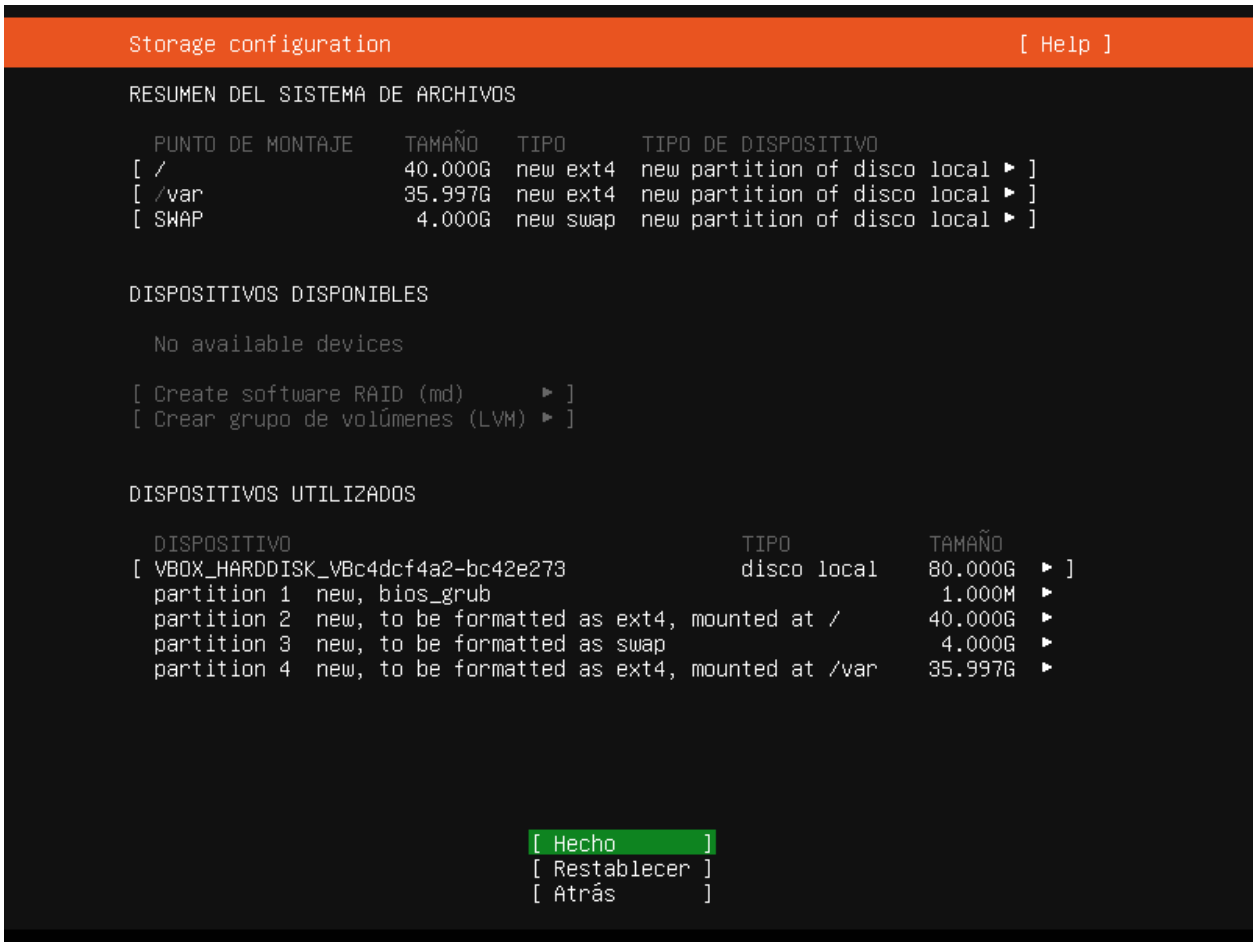


Particionado

El disco de 80GB se dividirá en 4 particiones:

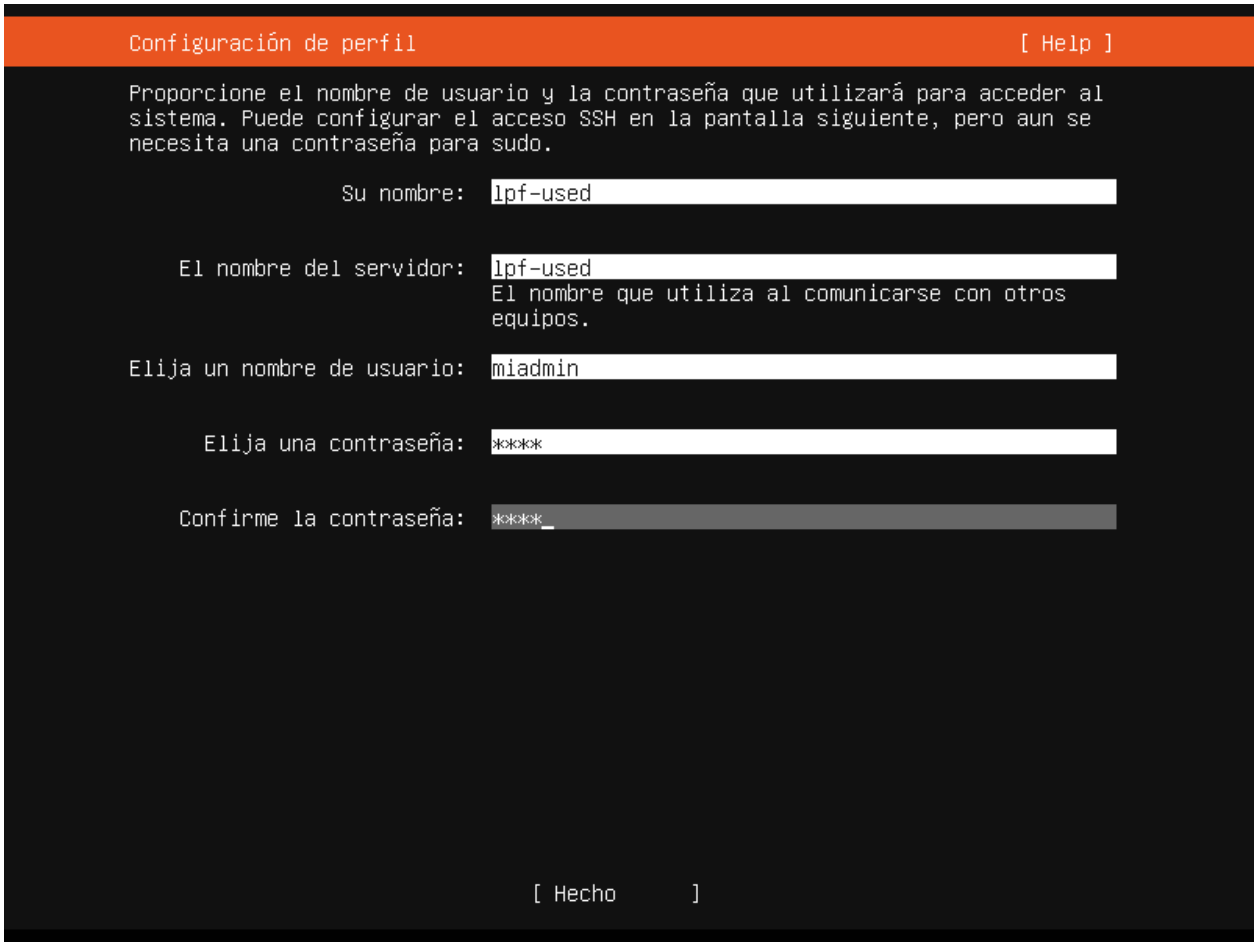
- La primera de 40GB, formato ext4 será la partición principal.
- La segunda de 35.997GB, formato ext4 será la partición de /var
- La tercera de 4GB será la partición swap
- La cuarta de 1MB será la partición para BIOS


Storage configuration
[Help]
To continue you need to: Mount a filesystem at /
RESUMEN DEL SISTEMA DE ARCHIVOS
No se montó ningún disco o partición.
DISPOSITIVOS DISPONIBLES
Adding GPT partition to VBOX_HARDDISK_VBc4dcf4a2-bc42e273
Size (max 79.997G): 40G
Formato: [ext4 ▾]
Mount: [/ ▾]
[Crear]
[Cancelar]
[Hecho]
[Restablecer]
[Atrás]


Storage configuration
[Help]
RESUMEN DEL SISTEMA DE ARCHIVOS
PUNTO DE MONTAJE TAMAÑO TIPO TIPO DE DISPOSITIVO
[/ 40.000G new ext4 new partition of disco local ▸]
[/var 35.997G new ext4 new partition of disco local ▸]
[SWAP 4.000G new swap new partition of disco local ▸]
DISPOSITIVOS DISPONIBLES
No available devices
[Create software RAID (md) ▸]
[Crear grupo de volúmenes (LVM) ▸]
DISPOSITIVOS UTILIZADOS
DISPOSITIVO TIPO TAMAÑO
[VBOX_HARDDISK_VBc4dcf4a2-bc42e273 disco local 80.000G ▸]
partition 1 new, bios_grub 1.000M ▸
partition 2 new, to be formatted as ext4, mounted at / 40.000G ▸
partition 3 new, to be formatted as swap 4.000G ▸
partition 4 new, to be formatted as ext4, mounted at /var 35.997G ▸
[Hecho]
[Restablecer]
[Atrás]

Creación de usuario

El nombre del servidor será lpf-used, el usuario miadmin y la contraseña será paso



CONFIGURACIÓN INICIAL

Configuración de red

En caso de necesitar modificar la configuración de red, habrá que modificar el archivo /etc/netplan/00-installer-config.yaml (el nombre del archivo puede tener variaciones en el nombre, dependiendo de la versión). Es necesario ser superusuario para realizar la modificación y aplicarla. Para aplicar la configuración será con:

sudo netplan apply o sudo netplan --debug apply

para mostrar los errores de forma más detallada

```
miadmin@lpf-used:~$ cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses:
      - 192.168.1.200/24
      gateway4: 192.168.1.1
      nameservers:
        addresses:
        - 8.8.8.8
        - 8.8.4.4
        - 192.168.1.1
  version: 2
miadmin@lpf-used:~$ _
```

Nombre de equipo

El cambio del nombre del equipo se puede hacer de dos formas, la primera haciendo cambios sobre los archivos /etc/hostname y /etc/hosts (en este último solo hay que sustituir tu antiguo nombre de host por el nuevo, puede aparecer varias veces).

Aunque hay un comando que lo hace de forma automática:

sudo hostnamectl set-hostname nuevoHostName.

Una vez hecho el cambio es necesario editar el archivo /etc/cloud/cloud.cfg y establecer preserve_hostname a true, para que el cambio se mantenga cuando se reinicie

Usuarios

```
# The top level settings are used as module
# and system configuration.

# A set of users which may be applied and/or used by various modules
# when a 'default' entry is found it will reference the 'default_user'
# from the distro configuration specified below
users:
   - default

# If this is set, 'root' will not be able to ssh in and they
# will get a message to login instead as the default $user
disable_root: true

# This will cause the set+update hostname module to not operate (if true)
preserve_hostname: true

# Example datasource config
# datasource:
#    Ec2:
#      metadata_urls: [ 'blah.com' ]
#      timeout: 5 # (defaults to 50 seconds)
#      max_wait: 10 # (defaults to 120 seconds)



# The modules that run in the 'init' stage
cloud_init_modules:
 - migrator
 - seed_random
 - bootcmd
 - write-files
 - growpart
 - resizefs
 - disk_setup
 - mounts
 - set_hostname
"/etc/cloud/cloud.cfg" [Modified] 133 lines --11%--                    15,23         Top
```

Usuario miadmin

En la instalación del sistema operativo se creó este usuario con su contraseña. Es el administrador del sistema

Usuario operadorweb

Se encargará de la gestión del servidor web (Apache). Será el administrador de la carpeta donde se subirán los archivos del servidor.

```
sudo useradd -d /var/www/html -s /bin/bash -G www-data operadorweb
```

-d: establece su directorio home

-s: establece su shell por defecto

-G: añade al usuario a grupos mediante su nombre, además de añadirlo a su grupo propio

operadorweb: nombre del usuario.

Y para establecer la contraseña usaremos:

```
sudo passwd operadorweb
```

Una vez creado el usuario necesitaremos establecer los permisos correspondientes y establecerlo como propietario:

```
sudo chmod -R 2775 /var/www/html
```

-R: modifica todos los permisos de los subdirectorios y archivos de la carpeta.

2: Para que el grupo al que el usuario pertenece sea propietario de todos los archivos y carpetas de ese directorio.

7: Da permisos de escritura, lectura y ejecución al usuario propietario

7: Da permisos de escritura, lectura y ejecución al grupo propietario

5: Da permisos de escritura y lectura a los demás usuarios.

/var/www/html: carpeta a la que se va a modificar los permisos

```
sudo chown -R operadorweb:www-data /var/www/html
```

-R: modifica también al grupo y usuario propietario de todas las carpetas y subcarpetas.

operadorweb: cambia el usuario propietario por, en este caso operadorweb.

www-data: cambia el grupo propietario por, en este caso www-data.

En caso de querer cambiar solo el usuario sería:

```
sudo chown -R operadorweb /var/www/html
```

El grupo se puede omitir, pero en caso de ser el grupo lo único a cambiar, será necesario usar el comando chgrp:

```
sudo chgrp -R ww-data /var/www/html
```

SSH

Instalación

La instalación de ssh puede hacerse al instalar el sistema operativo, una de las opciones de configuración te permite instalar ssh e incluso importa, si tienes, una clave ssh. En caso de no haber marcado esa opción, o necesitar instalarlo. Primero será recomendable actualizar los repositorios con:

```
sudo apt update
```

Una vez actualizado los repositorios, lo instalamos con:

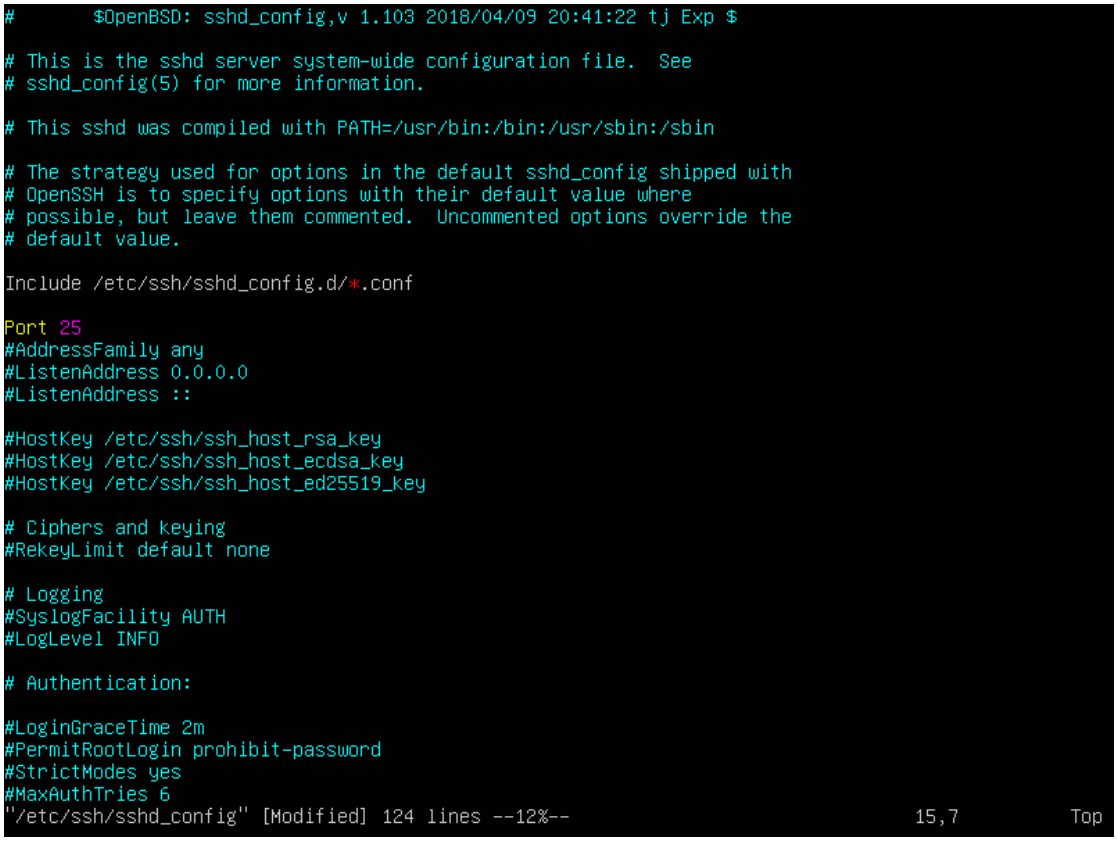
```
sudo apt install openssh-server
```

Para establecer una conexión ssh es necesario un cliente ssh, windows hace tiempo que implementó el comando, aunque en caso de no tenerlo existen varios clientes, putty el más famoso. En linux el cliente ssh es bastante común que venga instalado por defecto. Para establecer una conexión por comando será:

```
ssh nombreUsuario@direccionIP [-p nºPuerto]
```

El nº de puerto es opcional y por defecto se conectará al 22 pero en caso de cambiarlo en el archivo fallará y será necesario especificarlo.

Para cambiar el nº de puerto es necesario modificar el fichero /etc/ssh/sshd_config y descomentar o incluir la directiva Port y establecer el puerto que desees. También puedes crear un archivo acabado en .conf en /etc/ssh/sshd_config.d e incluir ahí las directivas. Por ejemplo /etc/ssh/sshd_config.d/puertos.conf donde le insertas Port 25.



*Si cambias el puerto, el firewall, aunque actives la regla ssh, solo contempla los puertos por defecto, es por eso que necesitarás activar el puerto correspondiente. Más adelante se explica cómo hacerlo.

SSH permite conectarse como un usuario determinado sin necesidad de introducir su contraseña, mediante las claves ssh. Estas claves constan de una clave pública que funciona a modo de cerradura y una clave privada que funciona a modo de llave.

ssh-keygen nos generará una clave pública y privada, en el home, en una carpeta oculta llamada .ssh, la clave privada estará en id_rsa, la pública en id_rsa.pub.

En esta carpeta se almacenarán también known_hosts y authorized_keys

Known_hosts, que corresponde a las claves de los equipos que se han conectado al menos una vez.

Authorized_keys es el archivo donde se almacenan todas las claves públicas, es decir, se copia el contenido de id_rsa.pub y se inserta aquí. Otra forma es:

ssh-copy-id [-p puerto] usuario@direccionIP
Que lo hará automáticamente. El archivo donde se almacenan las claves públicas puede ser cambiado en /etc/ssh/sshd_config.d con la directiva AuthorizedKeysFile

Configuración

Añadir usuario a www-data

Si bien el operadorweb es el encargado de administrar los contenidos del directorio en el que se almacenarán todos los ficheros del servidor web, podemos necesitar más usuarios para la administración, ahí es donde entra el grupo www-data, donde todo el que pertenezca al grupo dispondrá de los mismos permisos.

Para añadir un usuario a un grupo hay varias formas, una de ellas es modificar el fichero de configuración /etc/group, sin embargo es bastante peligroso modificar este archivo o similares.

Agregar un usuario a un grupo:

sudo usermod -a -G nombreDelGrupo nombreUsuario

El comando usermod permite hacer modificaciones sobre un usuario, para añadir un grupo puede.

-a: Añade a los grupos a los que pertenece el usuario, los grupos introducidos en la opción -G.

-G: En esta opción se especifican la lista de los grupos a los que el usuario pertenece.

gpasswd -a nombreUsuario nombreDelGrupo

El comando gpasswd permite administrar grupos y sus usuarios.

-a: Añade un usuario al grupo especificado

Para eliminar usuarios de un grupo:

sudo usermod -G nombreDeLosGrupo nombreUsuario

En usermod si solo se usa -G sin -a todos los grupos que se pasen por parámetros pasarán a ser los únicos grupo a los que pertenece el usuario. Por ejemplo el usuario “miadmin” pertenece a los grupos: miadmin, adm, cdrom, sudo, dip, plugdev y lxd en caso de que -G solo especificará el grupo sudo, pasará a formar parte de los grupos miadmin(el grupo personal del usuario aunque no se especifique no sale de él) y sudo.

gpasswd -d nombreUsuario nombreDelGrupo

-d: Borra al usuario especificado del grupo que se le pase por parametro

Acciones con SFTP sobre SSH

Para conectar mediante sftp a un servidor, (es necesario tener un servidor, en la máquina, instalado), necesitaremos un cliente sftp, windows y varias distribuciones linux suelen tenerlo instalado por defecto, sin embargo su uso es mediante comandos. En caso de quererlo existen clientes con interfaz gráfica, por ejemplo filezilla. Para el modo comandos:

Comando	Descripción
nombreUsuario@direccionIP	Establece un conexión con el servidor
help o ?	Muestra los comandos y ayuda de estos
bye o exit o quit	Cierra la conexión
cd	cambia de directorio
lcd	cambia de directorio en local
ls o dir	lista los archivos en el directorio actual
lls	lista los archivos de local
pwd	ruta actual
lpwd	ruta actual en local
chgrp	cambia el grupo del archivo/directorio
chown	cambia el propietario del archivo/directorio
chmod	cambia los permisos del archivo/directorio
put	sube un archivo
mput	similar a put, pero permite múltiples archivos
reput	en caso de error permite reanudar la subida
get	descarga un archivo
mget	similar a get, pero permite múltiples archivos
reget	en caso de error permite reanudar la descarga
mkdir	crea una carpeta

lmkdir	crea una carpeta en local
rename	renombra archivos y directorios *
rm	Borra el fichero indicado
rmdir	Borra el directorio indicado
progress	Activa o desactiva la barra de progreso
!	Abre la consola local, no sale de sftp
!comando	ejecuta el comando en la consola local
ln	Crea un link entre dos ficheros
df	muestra estadísticas del directorio actual

*También puede mover archivos, ejemplo:

```
rename pruebas/texto.txt pruebas2/texto.txt
```

*Algunos comandos como lumask que permite establecer el umask (permisos por defecto que tiene un archivo o directorio al crearse) se han omitido por no ser apenas utilizados

Si por algun motivo se necesitara cambiar el puerto sftp usa el mismo puerto que ssh en caso de cambiar el puerto por defecto de ssh cambiará el puerto de sftp

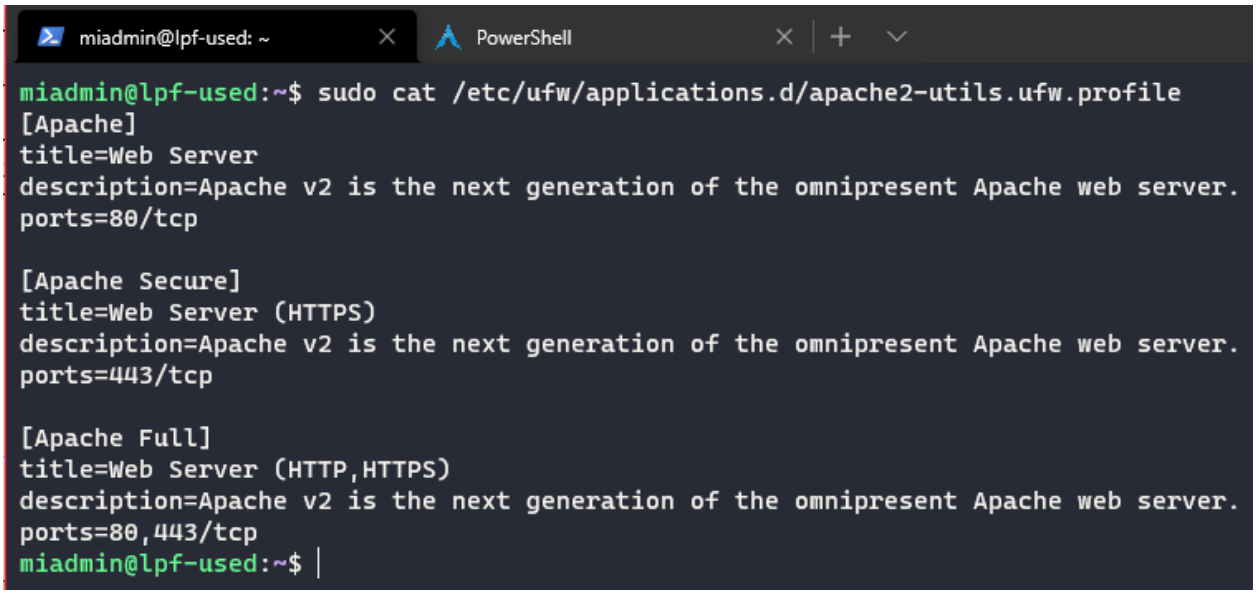
Cortafuegos

*Todos los comandos del cortafuegos se han de hacer siendo superusuario(sudo).

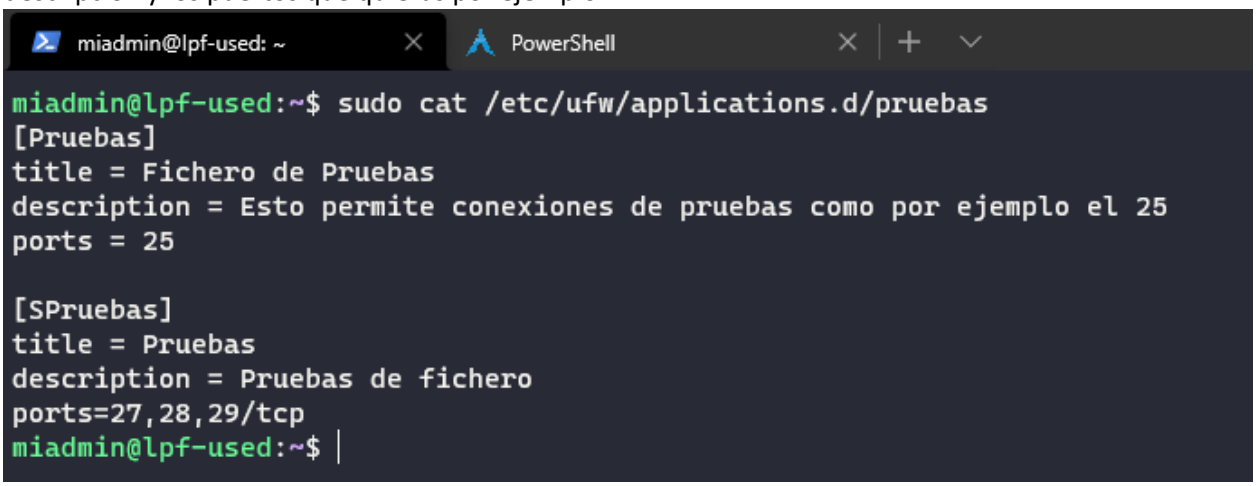
El cortafuegos se encarga de gestionar qué puertos están abiertos y quien puede entrar por dichos puertos. El cortafuegos tiene también unos perfiles o apps que suelen utilizarse para agrupar un conjunto de puertos que usa dicha aplicación.

Para crear un perfil en /etc/ufw/applications.d/ y creas un fichero.

Por ejemplo el fichero de perfil para apache tiene tres perfiles.



Cada nombre de perfil se define con corchetes, y va seguido de un título una descripción y los puertos que quieras por ejemplo



Una vez acabado hay que hacer ufw app update perfil. Es necesario actualizar cada vez que se modifica un perfil.

Con ufw app list nos muestra todos los perfiles disponible y si queremos saber informacion sobre algun perfil en concreto con ufw app info perfil y mostrará el título, la descripción y los puertos

Comando	Descripción
ufw enable	Activa el cortafuegos
ufw enable	

service ufw stop	Para el cortafuegos
ufw disable	Desactiva el cortafuegos
service ufw status	Estado del cortafuegos
ufw status	Muestra las reglas del cortafuegos
ufw reload	Reinicia el cortafuegos
ufw reset	Establece los valores por defecto
Reglas	
ufw allow puerto/app	Permite el acceso a un puerto para todos
ufw allow from IP	Permite el acceso a todos los puertos para la IP
ufw deny puerto/app	Impide el acceso a un puerto para todos
ufw deny from IP	Impide el acceso a todos los puertos desde una ip
ufw reject puerto/app	Rechaza el acceso a un puerto para todos
ufw reject from IP	Rechaza el acceso a todos los puertos desde una ip
ufw delete regla/num	Borra una regla, ejemplo delete allow 80
ufw insert núm regla	Inserta una regla en cierta posición

En el cortafuegos las posiciones son importantes y en ciertas ocasiones pueden generar problemas, sobretodo a la hora de establecer prohibiciones de acceso a IPs, por ejemplo, en caso de tener una IP bloqueada y tener un puerto accesible desde cualquier IP, es necesario que la regla del bloqueo esté antes, en caso contrario le permitirá conectarse por dicho puerto.

Cuando usamos una regla seguido de from pero solo queremos prohibir cierto puerto o perfil ejecutaremos el siguiente comando para un perfil o app:

```
ufw allow from ip to any port puerto
```

Y para un perfil o app

```
ufw allow from ip to any app nombreApp
```

APACHE

Instalación

sudo apt install apache2

Configuración

La configuración de apache en linux se encuentra en el directorio “etc”.

En este directorio se encuentran distintos archivos y ficheros de configuración. Los archivos de configuración son 3, aunque divididos en 2, los sitios/mods/configuraciones disponibles y los activados. En los directorios disponibles (available) se encuentran todos los sitios, mods o configuraciones pero puede estar sin activar. En la carpeta de enable(activados) se encuentran link simbólicos de los sitios,mods o configuraciones a sus respectivos ficheros de configuración en la carpeta available. Los comandos para habilitar o deshabilitar dichos ficheros son:

a2enconf “nombre_fichero_configuración” / a2disconf “nombre_fichero_configuración”

a2enmod “nombre_módulo” / a2dismod “nombre_módulo”

a2ensite “nombre_sitio” / a2dissite “nombre_sitio”

Apache separa los archivos de configuración en varios archivos de texto plano, que el archivo principal apache2.conf se encarga de incluir y leer, los directorios available son un sistema centralizado para luego poder incluir y leer las configuraciones. Por ejemplo los puertos por los que apache la escucha están en un fichero externo.

Para incluir un archivo hay dos formas **Include** e **IncludeOptional** ambas buscarán e incluirán el fichero pero en caso de no encontrarlo, IncludeOptional no mostrará error

```
IncludeOptional conf-enabled/*.conf
```

Las propiedades más interesantes son:

TimeOut

Va seguido de un número establece el número de segundos que tiene el servidor para responder a la solicitud.

KeepAlive

Establece si es necesario una nueva conexión por cada solicitud, KeepAlive off, o se permite dejar abierta la conexión para nuevas solicitudes.

MaxKeepAliveRequests

Número de solicitudes que se permite por cada conexión antes de que esta muera.

KeepAliveTimeout

Tiempo de espera entre cada solicitud, en caso de que se alcance la conexión morirá

Virtual Host

Un virtual host nos permitirá disponer de varios dominios o sitios virtuales. Por defecto en apache se establece en /var/www/html. Si queremos establecer un host virtual crearemos un directorio en /var/www y le daremos el nombre que queramos y establecemos permisos al usuario que lo administra.

```
drwxrwsr-x 5 operadorweb www-data 4096 Nov  4 19:50 html
drwxrwsr-x 2 operadorweb www-data 4096 Nov  7 19:48 pruebas
```

Una vez creado creamos o copiamos de alguno un archivo de configuración en etc/apache2/sites-available/, el nombre no importa pero el final tiene que acabar en .conf

```
<VirtualHost *:80>
        ServerAdmin luis.puefer@edua.jcyl.es
        DocumentRoot /var/www/pruebas_
        ServerName pruebas.local
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

En ServerName establecemos el nombre del servidor. Guardamos y ejecutamos sudo apache2ctl configtest para verificar los archivos, una vez ejecutado hay que habilitar el sitio con sudo a2ensite demo.conf (el archivo que he creado se llama demo.conf). Y reiniciamos con sudo service apache2 restart.

Si bien es cierto que el sitio ya esta configurado es necesario, configurar un dns para la resolución del nombre, también es posible modificar el fichero host de linux (/etc/hosts/) o windows (C:\Windows\System32\drivers\etc\hosts). Y añadimos:

192.168.1.204 pruebas.local

192.168.1.204 -> es la IP de la maquina.

pruebas.local -> ha de ser igual al **ServerName**, en caso contrario nos mostrará la página por defecto, si está activada o un mensaje de error al no encontrar la página.

dir.conf

Nos permite establecer que fichero se cargará con la carga de la página.

```
<IfModule mod_dir.c>
        DirectoryIndex pruebas.html index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

En este caso he establecido que el fichero pruebas.html sea el primero, si no se encuentra este archivo se buscará el siguiente, así hasta el final. Si no encuentra ninguno por defecto lista el directorio actual.

Para evitar que se pueda listar un directorio se ha de modificar el .htaccess(fichero de configuración de apache que se deposita en cada servidor virtual) o el apache2.conf y añadir un **"-"** al **Indexes**.

Por defecto si no se indica nada delante de cada opción se trata como activado **"+"**. Al introducir el "-" delante de una opción, salta error, para solucionarlo hay que indicar en cada opción si quieres activarla o desactivarla.

```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>

<Directory /var/www/pruebas>
        Options -Indexes +FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

ports.conf

En este archivo se establece por donde escuchan los puertos, por defecto se establece el 80 y varía en función de los módulos activados, por ejemplo en caso de tener el modulo ssl establece la escucha en el 443

```
Listen 80

<IfModule ssl_module>
        Listen 443
</IfModule>

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

Mensajes de error personalizados

En caso de querer establecer un mensaje de error personalizado puedes hacerlo modificando el htaccess o el fichero de configuración del host virtual. Los errores pueden ser **4xx**, error del cliente, o **5xx** del servidor. Para redirigir un error a una página personalizada será con **ErrorDocument <código> <ruta>**. La ruta es absoluta sobre el directorio donde se almacena nuestro virtual host.

```
<VirtualHost *:80>
        ServerAdmin luis.puefer@edua.jcyl.es
        DocumentRoot /var/www/pruebas
        ServerName pruebas.local
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log
        ErrorDocument 404 /error.html
</VirtualHost>
```

En este caso la ruta es **/var/www/pruebas/error.html**

.htaccess

Es un fichero de configuración específico para cada host. Antes de enviar la página al cliente apache leerá este fichero de configuración. Si está habilitado.

Para habilitarlo es necesario modificar una directiva **AllowOverride** en apache2.conf

```
<Directory /var/www/pruebas>
        Options -Indexes +FollowSymLinks
        AllowOverride all
        Require all granted
</Directory>
```

El valor de dicha directiva puede ser none, all o especificar qué directivas pueden ser sobreescritas, en caso de sobreescribir alguna otra, provocará un error del servidor. Por ejemplo con **AllowOverride DirectoryIndex** solo permitiré que cada host indique el orden de carga para su archivo inicial.

El nombre .htaccess puede ser cambiado con la directiva AccessFileName, en el fichero de configuración del host.

```
<VirtualHost *:80>
        ServerAdmin luis.puefer@edua.jcyl.es
        DocumentRoot /var/www/pruebas
        ServerName pruebas.local
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log
        AccessFileName .p1
</VirtualHost>
```

El fichero .htaccess está protegido contra la lectura pero al cambiar de nombre el fichero de configuración es posible acceder a él. Para evitarlo en el fichero de configuración principal se puede añadir una directiva **filesMatch expresiónRegural** y dentro de esta las opciones que se le vayan a aplicar a los ficheros que coincidan

```
<FilesMatch "^\.ht">
        Require all denied
</FilesMatch>

<FilesMatch "^\.p">
        Require all denied
</FilesMatch>
```

Mantenimiento

Si bien apache se puede reiniciar con service o systemctl, apache dispone de **apache2ctl restart** que permite reiniciar o en caso de fallo de sintaxis avisar por pantalla.

También están los archivos log en /var/log/apache2/ que muestran distinta información como los errores, accesos ...