

SERVIDORES APACHE

Instalación

Paso 1:

```
# Primero: es posible que no nos permita instalar porque tengamos que actualizar
sudo apt-get update

# Segundo: volvemos a ejecutar el install
sudo apt-get install [systemd] apache2 apache2-utils

# Tambien puede que tengamos que usar la opcion
sudo apt-get install --fix-missing apache2 apache2-utils
```

Los comandos de información y servicio.

Apache nos provee de una serie de opciones:

- Mostrar el status: `sudo service apache2 status`.
- Detener el servidor hasta el siguiente arranque: `sudo service apache2 stop`.
- Arrancar el servidor: `sudo service apache2 start`.
- Reiniciar el servidor: `sudo service apache2 restart`.
- Más información: `apachectl status`.
- Información de la versión: `apache2 -v`.

Paso 2:

```
# Instalar PHP:
sudo apt-get install php libapache2-mod-php php-mysql

# Instalar MySQL:
sudo apt-get install default-mysql-server
```

Paso 3: Realizamos un `ip a` para comprobar nuestra IP y la apuntamos, en nuestro caso: 1.25.

Paso 4: Es importante que Apache arranque solo, para ello ejecutamos: `systemctl enable apache2`.

Configuración

Como en FTP, trabajaremos todo el tema de la configuración en su mayoría en el archivo `/etc/apache2/apache2.conf`. En las carpetas `sites-available/enabled` se almacenan los ficheros de configuración de los hosts virtuales.

Crear hosts virtuales

En este caso vamos a crear un sitio para la web dreddsoft.com, para ello: **Paso 1:**

```
# Creamos el fichero de configuracion del virtualhost
sudo nano /etc/apache2/sites-available/dreddsoft.com.conf
```

Paso 2: Crear configuracion del virtualhost y activarlo.

```
# Tenemos que escribir las lineas de configuracion:
<VirtualHost *:80>
  # ServerAdmin webmaster@dreddsoft.com
  DocumentRoot "/var/www/html/dreddsoft.com"
  ServerName dreddsoft.com
  ServerAlias www.dreddsoft.com www.dreddsoft.es
  ErrorLog "dreddsoft.com-error.log"
  CustomLog "dreddsoft.com-access-log" combined
</VirtualHost>

# Activar el virtualhost
sudo a2ensite dreddsoft.com
# Desactivar
sudo a2dissite dreddsoft.com
```

Paso 3: Crear el directorio donde alojaremos el index y los ficheros de log

```
# Vamos con cd y comprobamos que no esta el directorio
cd /var/www/html
ls

# Si no aparece
sudo mkdir /var/www/html/dreddsoft.com
ls
```

No es necesario crear manualmente los archivos de log especificados en la configuración del VirtualHost, Apache los creará automáticamente.

- Nosotros los hemos definido con una ruta relativa, por lo que se crearan en el directorio predeterminado de logs `/var/log/apache2/`.
- Si deseamos especificar una dirección diferente, usamos rutas absolutas

```
ErrorLog "/var/www/html/dreddsoft.com/logs/dreddsoft.com-error-log"
CustomLog "/var/www/html/dreddsoft.com/logs/dreddsoft.com-access-log" combined
```

Pero para ello, debemos asegurarnos los permisos

```
sudo mkdir -p /var/www/html/dreddsoft.com/logs
sudo chown www-data:www-data /var/www/html/dreddsoft.com/logs
sudo chmod 755 /var/www/html/dreddsoft.com/logs
```

Paso 4: Los archivos hosts En Linux:

```
sudo nano /etc/hosts

# Hacemos que la ip apunte a dreddsoft.com
192.168.1.25 dreddsoft.com
```

En Windows:

1. Abrimos el bloc de notas como administrador.
2. Nos dirigimos a la ruta: `C:\Windows\System32\drivers\etc`.
3. Todos los archivos.
4. Abrimos el hosts.
5. Introducimos al final la linea de la IP y el dominio.
6. Guardamos.

Paso 5: Archivos index.html en el directorio de nuestro sitio.

```
cd /var/www/html/dreddsoft.com
ls

# Creamos el fichero index.html
nano /var/www/html/dreddsoft.com/index.html
# o con touch
touch /var/www/html/dreddsoft.com/index.html
```

Paso 6: Reiniciar / Recargar Apache.

```
/etc/init.d/apache2 restart || systemctl restart apache2
/etc/init.d/apache2 reload || systemctl reload apache2
```

Instalación y configuración de módulos.

PHP

```
sudo apt-get install php8.2
```

MySQL

```
# Usamos mariadb-server porque mysql-server no funciona
sudo apt-get install mariadb-server
```

Controlador MySQL de PHP

```
sudo apt-get install php8.2-mysql
```

Para comprobar esto, podemos crear un fichero **info.php** en el directorio de nuestro dominio **/var/www/html/dreddsoft/** con una sentencia de código muy corta de PHP:

```
<?php
phpinfo();
```

Ahora si en el navegador escribimos: *dreddsoft.com/info.php*, se nos debe mostrar la página de información de PHP.

Algunas librerías de PHP y el PHPMysqlAdmin:

```
sudo apt-get install php-gd php-imagick php-apcu
sudo apt-get install phpmyadmin
```

Soporte para Python:

```
# Instalar mod_wsgi
sudo apt-get update
sudo apt-get install apache2 libapache2-mod-wsgi-py3

# Verificar que mod_wsgi esta habilitado y reiniciar Apache
sudo a2enmod wsgi
sudo systemctl restart apache2

# Insertar estas lineas en el archivo de configuracion del virtualhost
WSGIDaemonProcess myapp python-path=/var/www/html/dreddsoft
WSGIScriptAlias / /var/www/html/dreddsoft/app.wsgi

<Directory /var/www/html/dreddsoft>
    WSGIProcessGroup myapp
    WSGIApplicationGroup %{GLOBAL}
    Require all granted
</Directory>

# Crear un archivo WSGI
sudo nano /var/www/html/dreddsoft/app.wsgi
```

```
# Introducir este código
import sys
sys.path.insert(0, '/var/www/myapp')

from app import application

# Reload
sudo systemctl reload apache2
```

Autenticación HTTP

Apache utils.

```
sudo apt-get install apache2-utils
```

Creamos primero un usuario con su contraseña y almacenamos el fichero en la ruta de configuración de nuestro servidor web. Como el fichero todavía no existe usamos la opción **-c**.

```
sudo htpasswd -c /etc/apache2/.htpasswd andres
# Nos pedirá la contraseña, a lo que pondremos 1234

# Añadimos un segundo usuario, como el fichero ya existe:
sudo htpasswd /etc/apache2/.htpasswd otro_usuario
# contraseña 4321

# Para ver cómo quedan los usuarios, primero podemos usar ls -a que es un ls con
esteroides
cd /etc/apache2
ls -a
nano /etc/apache2/.htpasswd

# Asignamos la propiedad del fichero a www-data
chown www-data:www-data /etc/apache2/.htpasswd
```

Ahora, en el fichero de configuración de nuestros hosts virtuales vamos a proteger uno de ellos. Vamos a añadir la directiva **Directory** para proteger un directorio al que llamaremos usuarios, donde se supone que hay cierta información que no todo el mundo debería acceder, para ello:

```
<Directory "/var/www/html/dreddsoft.com/usuarios">
AuthType Basic
AuthName "Acceso Restringido a Usuarios"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
</Directory>

# Reiniciar
sudo systemctl restart apache2
```

```
# Recargar
sudo systemctl reload apache2
```

Autenticación HTTP con .htaccess

Para permitir que las propiedades sobre el directorio raíz puedan ser modificadas tenemos que:

[/etc/apache2/apache2.conf](#)

```
# Se modifican estas líneas, cambiando el valor AllowOverride None por All
<Directory /var/www/>
Options Indexes FollowSymLinks
AllowOverride All
Require all granted
</Directory>
```

En la carpeta que queremos proteger, crearemos un fichero .htaccess. La ruta:

[/var/www/html/dreddsoft.com/usuarios](#)

```
sudo nano /var/www/html/dreddsoft.com/usuarios/.htaccess

# Añadimos las líneas:
AuthType Basic
AuthName "Acceso Restringido a Usuarios"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user

# Reiniciar
/etc/init.d/apache2 restart

# Recargar
/etc/init.d/apache2 reload
```

Para impedir que se liste un directorio, tenemos que añadir a nuestro archivo .htaccess

```
Options -Indexes
```

Limitar el ancho de banda.

```
sudo apt-get install libapache2-mod-bw
```

Configuramos el ancho de banda del sitio creado, modificando en su archivo de configuración

[/etc/apache2/sites-available/dreddsoft.com.conf](#).

```
# Tenemos que añadir:
BandwidthModule On
ForceBandWidthModule On
Bandwidth all 64
MinBandwidth all -1
MaxConnection all 2

# También es posible limitar la velocidad de un directorio específico:
<Directory "/var/www/example1.com/public_html/media">
LargeFileLimit * 10240 102400
</Directory>
```

Personaliza las páginas de error.

Para personalizar las páginas de error, primero tenemos que crear los html de los errores personalizados:

```
cd /var/www/html/dreddsoft.com
ls

# Creamos el error 404
nano /var/www/html/dreddsoft.com/error_404.html

# Creamos el error 500
nano /var/www/html/dreddsoft.com/error_500.html

# Luego en el archivo de configuracion
nano /etc/apache2/sites-available/dreddsoft.com.conf
ErrorDocument 404 /error_404.html
ErrorDocument 500 /error_500.html
```

Seguridad

Configurar SSL/TLS en Apache

El primer paso es crear el certificado y la clave, que quedarán almacenados en `/etc/apache2/certs`.

```
# Creamos el directorio
sudo mkdir /etc/apache2/certs

# crear el certificado SSL
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/apache2/certs/apache2.key -out /etc/apache2/certs/apache2.crt

# Activamos el módulo SSL de Apache:
sudo a2enmod ssl

# Reiniciamos
systemctl restart apache2
```

Ahora, modificamos el archivo de configuración de un host virtual para que soporte una conexión segura.

```
# Cambiamos el puerto al 443
<VirtualHost *:443>

# Esta es la parte que introducimos para el certificado SSL
SSLEngine On
SSLCertificateFile /etc/apache2/certs/apache2.crt
SSLCertificateKeyFile /etc/apache2/certs/apache2.key
SSLProtocol All -SSLv3
</VirtualHost>
```

Estadísticas Web.

Instalar webalizer

```
sudo apt-get install webalizer

# Archivo de configuracion del webalizer
sudo nano /etc/webalizer/webalizer.conf

# Importante que estas líneas estén correctas
LogFile /var/log/apache2/access.log
OutputDir /var/www/html/webalizer

# También que la carpeta esté en la dirección correcta, en nuestro caso
/var/www/html/
# Si no lo está, podemos moverla con:
mv /var/www/webalizer /var/www/html/

# Y con eso me ha funcionado.
```

Monitorización

Usamos GoAccess, que es un analizador visual de logs de apache en tiempo real.

```
# Tenemos que ejecutar los siguientes comandos
sudo echo "deb http://deb.goaccess.io/ stretch main" >> /etc/apt/sources.list
sudo apt-get update
sudo apt-get install goaccess

# Para visualizar por consola basta con el comando goaccess y la ruta del
access.log
sudo goaccess /var/log/apache2/access.log -c

# Para monitorizar en html
```



```
sudo goaccess /var/log/apache2/access.log -o /var/www/html/report.html --log-format=COMBINED --real-time-html
```

Dependiendo de si queremos que el GoAccess nos de los registros generales o si queremos la información de un dominio concreto, cambiaremos la ruta de los logs en el comando del goaccess. Ejemplo:

```
# CustomLog "/var/www/html/dreddsoft.com/logs/dreddsoft.com-access-log" combined
sudo goaccess /var/www/html/dreddsoft.com/logs/dreddsoft.com-access-log -o
/var/www/html/report.html --log-format=COMBINED --real-time-html
```