Trabajo Final (DAW)

Christian Ramos Ortiz

Índice

Apache:

Apache es el servidor web más popular para servir contenido en Internet. Cuenta con más de la mitad de los sitios activos en Internet y es extremadamente poderoso y flexible.

Apache rompe su funcionalidad y componentes en unidades separadas que pueden ser personalizadas y configuradas de manera independiente. La unidad básica que describe a un sitio o dominio es denominada `virtual host` (ó alojamiento virtual en español).

Esta designación permite al administrador hacer uso de un servidor para alojar múltiples dominios o sitios en una única interfaz o IP utilizando un mecanismo de coincidencias. Esto es relevante para cualquiera que desee alojar más de un sitio en un mismo VPS.

Fuente (introducción):

- https://www.digitalocean.com/community/tutorials/como-configurar-virtual-hosts-de-apache-en-ubuntu-16-04-es

Virtual Hosting

Para configurar un virtual hosting con Apache deberemos de hacer lo siguiente:

Primero deberemos de crear la estructuras de los directorios para el servidor web, usaremos el comando:

```
sudo mkdir -p /var/www/2daw.com
```

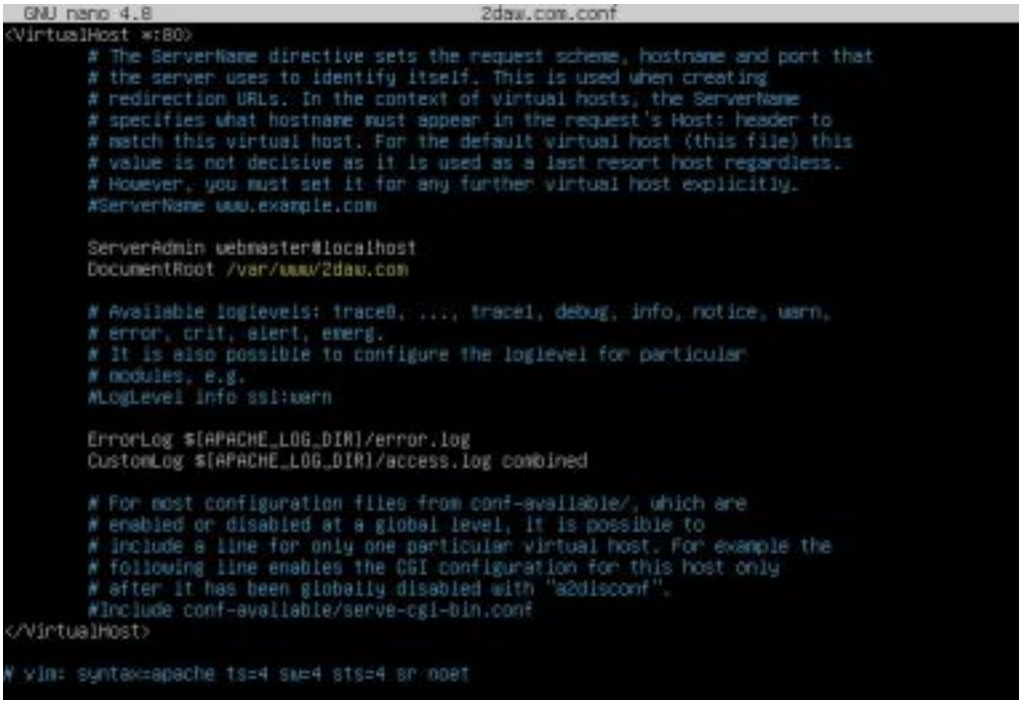
Ahora le daremos los permisos:

```
sudo chmod -R 755 /var/www
```

Creamos el archivo de configuración para el Virtual Host (ejemplo 2daw.com.conf):

```
sudo cp /etc/apache2/sites-available/000-default.conf
/etc/apache2/sites-available/2daw.com.conf
```

Editaremos el “DocumentRoot”, el archivo de configuración deberá de quedar así:



Mapeo de URL

Directiva “Options” para el mapeo de URL:

- *All:* Todos los directorios
- *FollowSymLinks:* Para ver directorios fuera de la carpeta default con enlaces simbólicos
- *Indexes:* Para no poner el “index.html” cada vez que se entre a una URL
- *MultiViews:* Ofrecer varias páginas diferentes (Por ejemplo para diferentes idiomas)
- *SymLinksIfOwnerMatch:* Para los enlaces simbólicos que haya hecho el propietario
- *ExecCGI*: Permite que el servidor ejecute scripts por ejemplo de Perl

Para el mapeo de URL editaremos el archivo de configuración de apache “apache2.conf”, un ejemplo de mapeo de URL sería:

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

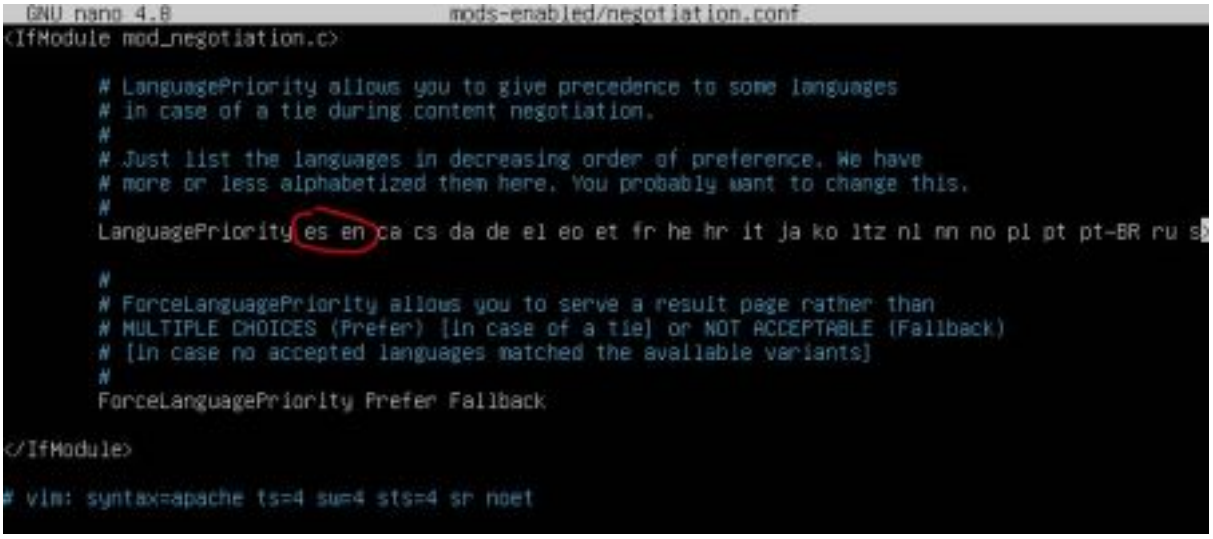
“Indexes” muestra el listado de las carpetas, y los enlaces simbólicos.
“AllowOverride”, para el archivo .htaccess
“Require all granted”, permite el acceso a todo el mundo

Para la negociación de contenidos, editaremos el archivo “negotiation.conf”, deberemos usar la opción "Options +Multiviews" en el “apache2.conf”, de esta manera:

Archivo “apache2.conf”

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

Archivo “mods-enabled/negotiation.conf”



Para que nos redirija al idioma que deseamos, deberemos añadir al final del nombre de los archivos el siguiente formato:

```
christian@server:/etc/apache2$ ls /var/www/html/
index.html.es
```

Control de acceso

Directivas "Require":

- **Require all granted:** Concede acceso a todos
- **Require all dined:** Deniega el acceso a todos
- **Require user userid [userid] ... :** Acceso para un usuario concreto
- **Require group group-name [group-name] ... :** Acceso para un grupo concreto
- **Require valid-user:** Acceso para un usuario/contraseña válido
- **Require ip [ip] ... :** Acceso para una ip específica
- **Require host [dominio]:** Acceso mediante un dominio específico
- **Require local:** Acceso solamente local

Un ejemplo de directiva que negara el acceso a todos los usuarios sería:

```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all denied
</Directory>
```

Un ejemplo de autenticación básica sería:

```
<Directory /var/www/apache2/privado>
        AuthUserFile "/etc/apache2/claves/passwd.txt"
        AuthName "Palabra de paso"
        AuthType Basic
        Require valid-user
</Directory>
```

- *AuthUserFile:* Archivo con contraseñas
- *AuthName*: Nombre de usuario para acceder
- *AuthType:* Tipo de acceso
- *Require:* Requisito de un usuario/contraseña válido

Existe otro tipo de "AuthType" más seguro llamado "Digest" en vez de "Basic"

VSFTPD:

Vsftpd es uno de los servidores FTP más potentes y completos disponibles para la mayoría de distribuciones de Linux. Este servidor FTP es el favorito de muchos administradores de sistemas por la configurabilidad que es capaz de proporcionarnos, y por la facilidad de configuraciones avanzadas en el propio servidor FTP. Hoy en RedesZone os vamos a enseñar cómo podemos instalarlo, configurarlo y también cómo habilitar el protocolo FTPES para que toda la comunicación esté cifrada

Fuente (introducción):

- https://www.redeszone.net/tutoriales/servidores/vsftpd-configuracion-servidor-ftp/

Instalación

Para llevar a cabo la instalación solo tendremos que usar el comando:

```
sudo apt install vsftpd
```

Y lanzar el servicio con:

```
sudo systemctl start vsftpd
```

```
sudo systemctl enable vsftpd
```

Configuración

1. **Crearemos el directorio de los usuarios**

   ```
   sudo adduser christian
   ```

   ```
   sudo mkdir /home/christian/ftp
   ```

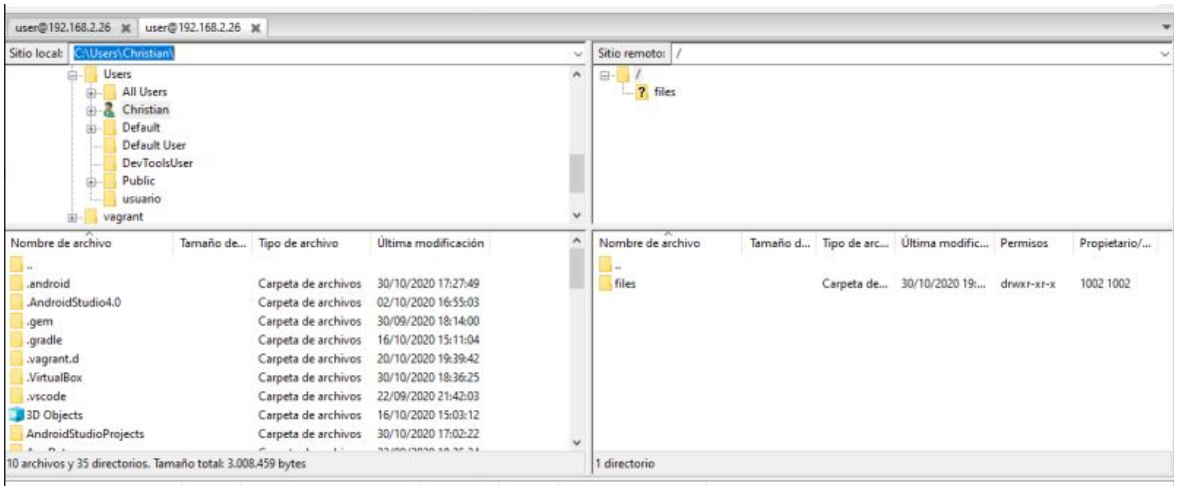
2. **Configuramos el archivo “vsftpd.conf”**

   Tendremos que cambiar las opciones:

   ```
   local_enable=YES
   ```

   ```
   write_enable=YES
   ```

   ```
   chroot_local_user=YES
   ```



Configuración con SSL

1. **Creamos el certificado SSL**

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048
-keyout /etc/ssl/private/vsftpd.pem -out
/etc/ssl/private/vsftpd.pem
```

2. **Abrimos el archivo “vsftpd.conf”, y editamos estas líneas con el lugar donde hayamos guardado los certificados**

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem
```

```
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

3. **Habilitamos el SSL (Archivo “vsftpd.conf”)**

```
ssl_enable=YES
```

GIT:

Hoy en día, Git es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto. Los desarrolladores que han trabajado con Git cuentan con una buena representación en la base de talentos disponibles para el desarrollo de software, y este sistema funciona a la perfección en una amplia variedad de sistemas operativos e IDE (entornos de desarrollo integrados).

Fuente (introducción):

- **https://www.atlassian.com/es/git/tutorials/what-is-git**

Configuración inicial

Deberemos de añadir el usuario y el email:

```
usuario@hobbit:~$ git config --global user.name "Christian Ramos"
usuario@hobbit:~$ git config --global user.email "christian.ramos-ortiz@iesruizgijon.com"
```

La podemos visualizar con el siguiente comando:

```
usuario@hobbit:~$ git config --global --list
user.name=Christian Ramos
user.email=christian.ramos-ortiz@iesruizgijon.com
```

Creación de repositorio local y remoto

REPOSITORIO LOCAL

Inicializamos un nuevo repositorio de Git de la siguiente manera “git init nombre_repo”:

```
usuario@hobbit:~$ git init hello-world
Initialized empty Git repository in /home/usuario/hello-world/.git/
```

Con “git add .” pasaremos a la zona de STAGING los cambios que realizemos

```
usuario@hobbit:~/hello-world$ git add README.md
usuario@hobbit:~/hello-world$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
```

Para hacer el commit usamos el comando “git commit -m ‘mensaje’”

```
usuario@hobbit:~/hello-world$ git commit -m "Primera revisión"
[master (root-commit) 16a236f] Primera revisión
 1 file changed, 2 insertions(+)
 create mode 100644 README.md
```

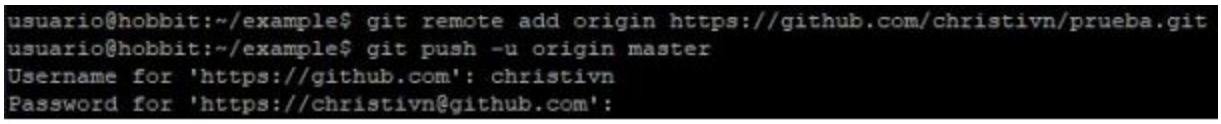
REPOSITORIO REMOTO

Realizaremos los mismos pasos que el anterior cuando trabajemos de manera local, y a la hora de subir el repositorio a remoto usaremos el siguiente comando:

```
git remote add origin https://github.com/christivn/prueba.git
```

```
git push -u origin master
```

Deberemos de introducir el usuario y contraseña de nuestro Github



Comandos básicos

git help

Muestra una lista con los comandos más utilizados en GIT.

git init

Podemos ejecutar ese comando para crear localmente un repositorio con GIT y así utilizar todo el funcionamiento que GIT ofrece. Basta con estar ubicados dentro de la carpeta donde tenemos nuestro proyecto y ejecutar el comando. Cuando agreguemos archivos y un commit, se va a crear el branch master por defecto.

git add -A

Agrega al repositorio TODOS los archivos y carpetas que estén en nuestro proyecto, los cuales GIT no está siguiendo.

git commit -m "mensaje" + archivos

Hace commit a los archivos que indiquemos, de esta manera quedan guardados nuestras modificaciones.

git commit -am "mensaje"

Hace commit de los archivos que han sido modificados y GIT los está siguiendo.

git checkout -b NombreDeBranch

Crea un nuevo branch y automaticamente GIT se cambia al branch creado, clonando el branch desde donde ejecutamos el comando.

git branch

Nos muestra una lista de los branches que existen en nuestro repositorio.

git checkout NombreDeBranch

Sirve para moverse entre branches, en este caso vamos al branch que indicamos en el comando.

git merge NombreDeBranch

Hace un merge entre dos branches, en este caso la dirección del merge sería entre el branch que indiquemos en el comando, y el branch donde estémos ubicados.

git status

Nos indica el estado del repositorio, por ejemplo cuales están modificados, cuales no están siendo seguidos por GIT, entre otras características.

git clone URL/name.git NombreProyecto

Clona un proyecto de git en la carpeta NombreProyecto.

git push origin NombreDeBranch

Luego de que hicimos un git commit, si estamos trabajando remotamente, este comando va a subir los archivos al repositorio remoto, específicamente al branch que indiquemos.

git pull origin NombreDeBranch

Hace una actualización en nuestro branch local, desde un branch remoto que indicamos en el comando.