

Práctica: Instalación y configuración de servidor web Nginx

Tabla de Contenido

1. Instalación servidor web Nginx	1
2. Creación de las carpeta del sitio web	1
3. Configuración de servidor web NGINX.....	2
3.1. Comprobaciones	4
4. FTP	4
4.1. Transferir archivos al servidor	5
4.2. Configurar servidor FTPS en Debian	5
5. Tarea	7
6. Cuestiones finales.....	7
6.1. Evaluación	7

1. Instalación servidor web Nginx

Para instalar el servidor nginx en nuestra Debian, primero actualizamos los repositorios y después instalamos el paquete correspondiente:

```
sudo apt update
sudo apt install nginx
```

Comprobamos que nginx se ha instalado y que está funcionando correctamente:

```
systemctl status nginx
```

2. Creación de las carpeta del sitio web

Igual que ocurre en Apache, todos los archivos que formarán parte de un sitio web que servirá nginx se organizarán en carpetas. Estas carpetas, típicamente están dentro de `/var/www`.

Así pues, vamos a crear la carpeta de nuestro sitio web o dominio:

```
sudo mkdir -p /var/www/nombre_web/html ①
```

① Donde el `nombre_web` puede ser la palabra que queráis, sin espacios.

Dentro de esa carpeta html, debéis clonar el siguiente repositorio:

```
https://github.com/cloudacademy/static-website-example
```

Además, haremos que el propietario de esta carpeta y todo lo que haya dentro sea el usuario `www-data`, típicamente el usuario del servicio web.

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

Y le daremos los permisos adecuados para que no nos de un error de acceso no autorizado al entrar en el sitio web:

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

Para comprobar que el servidor está funcionando y sirviendo páginas correctamente, podéis acceder desde vuestro cliente a:

```
http://IP-maq-virtual
```

Y os deberá aparecer algo así:

:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Figura 1. Página inicial por defecto de Nginx

Lo que demuestra que todo es correcto hasta ahora.

3. Configuración de servidor web NGINX

En Nginx hay dos rutas importantes. La primera de ellas es `sites-available`, que contiene los archivos de configuración de los hosts virtuales o bloques disponibles en el servidor. Es decir, cada uno de los sitios webs que alberga el servidor. La otra es `sites-enabled`, que contiene los archivos de configuración de los sitios habilitados, es decir, los que funcionan en ese momento.

Dentro de `sites-available` hay un archivo de configuración por defecto (`default`), que es la página que se muestra si accedemos al servidor sin indicar ningún sitio web o cuando el sitio web no es encontrado en el servidor (debido a una mala configuración por ejemplo). Esta es la página que nos ha aparecido en el apartado anterior.

Para que Nginx presente el contenido de nuestra web, es necesario crear un bloque de servidor con las directivas correctas. En vez de modificar el archivo de configuración predeterminado directamente, crearemos uno nuevo.

```
sudo nano /etc/nginx/sites-available/vuestro_dominio
```

Con el siguiente contenido:

`/etc/nginx/sites-available/vuestro_dominio`

```
server {
    listen 80;
    listen [::]:80;
    root /ruta/absoluta/archivo/index;
    index index.html index.htm index.nginx-debian.html;
    server_name nombre_web;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

Aquí la directiva `root` debe ir seguida de la ruta absoluta absoluta dónde se encuentre el archivo `index.html` de nuestra página web, que se encuentra entre todos los que habéis descomprimido.

Aquí tenéis un ejemplo de un sitio webs con su ruta (directorios que hay) antes del archivo `index.html`:

Ruta `/var/www/ejemplo2/html/2016_soft_landing`

```
ejemplo2/
├── html/
│   └── 2016_soft_landing/
│       ├── ABOUT THIS TEMPLATE.txt
│       ├── css/
│       ├── ...
│       ├── fonts/
│       ├── ...
│       ├── images/
│       ├── ...
│       ├── index.html
│       ├── js
│       └── ...
```

Y crearemos un archivo simbólico entre este archivo y el de sitios que están habilitados, para que se dé de alta automáticamente.

```
sudo ln -s /etc/nginx/sites-available/nombre_web /etc/nginx/sites-enabled/
```

Y reiniciamos el servidor para aplicar la configuración:

```
sudo systemctl restart nginx
```

3.1. Comprobaciones

3.1.1. Comprobación del correcto funcionamiento

Como aún no poseemos un servidor DNS que traduzca los nombres a IPs, debemos hacerlo de forma manual. Vamos a editar el archivo `/etc/hosts` de nuestra máquina anfitriona para que asocie la IP de la máquina virtual, a nuestro `server_name`. Este archivo, en Linux, está en `/etc/hosts` y en Windows: `C:\Windows\System32\drivers\etc\hosts`

Y deberemos añadirle la línea:

```
192.168.X.X nombre_web
```

donde debéis sustituir la IP por la que tenga vuestra máquina virtual.

3.1.2. Comprobar registros del servidor

Comprobad que las peticiones se están registrando correctamente en los archivos de logs, tanto las correctas como las erróneas:

`/var/log/nginx/access.log`

cada solicitud a su servidor web se registra en este archivo de registro, a menos que Nginx esté configurado para hacer algo diferente.

`/var/log/nginx/error.log`

cualquier error de Nginx se asentará en este registro.



Si no os aparece nada en los logs, podría pasar que el navegador ha cacheado la página web y que, por tanto, ya no está obteniendo la página del navegador sino de la propia memoria. Para solucionar esto, podéis acceder con el *modo privado* del navegador y ya os debería registrar esa actividad en los logs.

4. FTP

Si queremos tener varios dominios o sitios web en el mismo servidor nginx (es decir, que tendrán la

misma IP) debemos repetir todo el proceso anterior con el nuevo nombre de dominio que queramos configurar.

4.1. Transferir archivos al servidor

¿Cómo transferir archivos desde nuestra máquina local/anfitrión a nuestra máquina virtual Debian/servidor remoto?

A día de hoy el proceso más sencillo y seguro es a través de Github. No obstante, el currículum nos obliga a enseñaros un método un tanto obsoleto (aunque se usa) a día de hoy, así que vamos a ello, os presento al FTP.

El **FTP** es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP. Como su nombre indica, se trata de un protocolo que permite transferir archivos directamente de un dispositivo a otro. Actualmente, es un protocolo que poco a poco va abandonándose, pero ha estado vigente más de 50 años.

El protocolo FTP tal cual es un protocolo inseguro, ya que su información no viaja cifrada. Sin embargo, esto se solucionó con el protocolo **FTPS**, que le añade una capa para hacerlo más seguro y privado. FTPS no es más que el mismo protocolo FTP pero implementado por un canal seguro.

La seguridad que nos aporta FTPS es importante para la transferencia de archivos porque, si no disponemos de ella, los archivos viajarán tal cual por la red, sin ningún tipo de encriptación. Así pues, usando FTP tradicional, si algún agente consigue escuchar las transferencias, podría ocurrir que la información quedase al descubierto. Esto sería especialmente importante si los archivos que subimos contienen información confidencial o datos personales.

Dado que usar FTPS aporta mayor seguridad a las transmisiones, es recomendable utilizarlo, más aún sabiendo que realmente no hay mucha dificultad en establecer las conexiones por el protocolo seguro.

4.2. Configurar servidor FTPS en Debian

En primer lugar, lo instalaremos desde los repositorios:

```
sudo apt-get update
sudo apt-get install vsftpd
```

Ahora vamos a crear una carpeta en nuestro *home* en Debian:

```
mkdir /home/nombre_usuario/ftp
```

En la configuración de **vsftpd** indicaremos que este será el directorio al cual **vsftpd** se cambia después de conectarse el usuario.

Ahora vamos a crear los certificados de seguridad necesarios para aportar la capa de cifrado a nuestra conexión (algo parecido a HTTPS)

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

Y una vez realizados estos pasos, procedemos a realizar la configuración de **vsftpd** propiamente dicha. Se trata, con el editor de texto que más os guste, de editar el archivo de configuración de este servicio, por ejemplo con **nano**:

```
sudo nano /etc/vsftpd.conf
```

En primer lugar, buscaremos las siguientes líneas del archivo y las eliminaremos por completo:

```
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem  
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key  
ssl_enable=NO
```

Tras ello, añadiremos estas líneas en su lugar

```
rsa_cert_file=/etc/ssl/certs/vsftpd.crt  
rsa_private_key_file=/etc/ssl/private/vsftpd.key  
ssl_enable=YES  
allow_anon_ssl=NO  
force_local_data_ssl=YES  
force_local_logins_ssl=YES  
ssl_tlsv1=YES  
ssl_sslv2=NO  
ssl_sslv3=NO  
require_ssl_reuse=NO  
ssl_ciphers=HIGH  
  
local_root=/home/nombre_usuario/ftp
```

Y, tras guardar los cambios, reiniciamos el servicio para que coja la nueva configuración:

```
sudo systemctl restart vsftpd
```

Tras acabar esta configuración, ya podremos acceder a nuestro servidor mediante un cliente FTP adecuado, como por ejemplo **Filezilla** de dos formas, a saber:

1. Mediante una conexión FTP sin cifrar en el puerto 21.
2. Mediante una conexión FTPS, utilizando los certificados. Esta conexión puede a su vez ser de dos tipos:
 - a. FTP implícito sobre SSL. Esta variante significa que la conexión cifrada es implícita. No es necesario solicitarla. No obstante, esta variante está desaconsejada hoy en día.

- b. FTP explícito sobre SSL. También se le conoce simplemente como FTPES. Se establece una conexión estándar mediante el puerto 21 con el servidor y se solicita explícitamente la negociación SSL.

5. Tarea

Configura un nuevo dominio para un sitio web que hayas desarrollado tú. En este caso debéis transferir los archivos a vuestra Debian mediante FTPES.

Recordemos que debemos tener nuestro sitio web en la carpeta `/var/www` y darle los permisos adecuados, de forma similar a cómo hemos hecho con el otro sitio web.

6. Cuestiones finales

¿Qué pasa si no hago el link simbólico entre `sites-available` y `sites-enabled` de mi sitio web?

...

¿Qué pasa si no le doy los permisos adecuados a `/var/www/nombre_web`?

...

6.1. Evaluación

Criterio	Puntuación
Configuración correcta del servidor web	1 puntos
Comprobación del correcto funcionamiento del primer sitio web	3 puntos
Configuración correcta y comprobación del funcionamiento de una segunda web	2 puntos
Cuestiones finales	2 puntos
Se ha prestado especial atención al formato del documento, utilizando la plantilla actualizada y haciendo un correcto uso del lenguaje técnico	2 puntos