

# Instalación y configuración de servidor web Nginx:

Para instalar y configurar el servidor deberemos actualizar repositorios. Una vez actualizados instalaremos el servidor Nginx. Utilizaremos el siguiente comando: “sudo apt install nginx”

```
a24gadilu@debian:~$ sudo apt install nginx
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  nginx-common
Paquetes sugeridos:
  fcgiwrap nginx-doc ssl-cert
Se instalarán los siguientes paquetes NUEVOS:
  nginx nginx-common
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Comprobamos que este instalado y funcione correctamente con el comando: “systemctl status nginx”

```
a24gadilu@debian:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-09-23 10:11:45 CEST; 7min ago
     Docs: man:nginx(8)
  Process: 823 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 824 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 849 (nginx)
    Tasks: 3 (limit: 2306)
   Memory: 2.5M
      CPU: 23ms
   CGroup: /system.slice/nginx.service
           └─849 'nginx: master process /usr/sbin/nginx -g daemon on; master_process on;'
             └─851 'nginx: worker process'
               └─852 'nginx: worker process'

sep 23 10:11:45 debian systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
sep 23 10:11:45 debian systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
a24gadilu@debian:~$
```

Creamos las carpetas del sitio web de la siguiente manera:

“sudo mkdir -p /var/www/html”

```
a24gadilu@debian:/$ sudo mkdir -p /var/www/a24gadilu/html
a24gadilu@debian:/$ cd /var/www/a24gadilu/html
```

Dentro de esta carpeta tenemos que clonar el siguiente repositorio: “git clone <https://github.com/cloudacademy/static-website-example>”

```
a24gadilu@debian:/var/www/a24gadilu/html$ sudo git clone https://github.com/cloudacademy/static-website-example
Clonando en 'static-website-example'...
remote: Enumerating objects: 69, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 69 (delta 8), reused 8 (delta 8), pack-reused 45 (from 1)
Recibiendo objetos: 100% (69/69), 668.08 KiB | 1.45 MiB/s, listo.
Resolviendo deltas: 100% (9/9), listo.
a24gadilu@debian:/var/www/a24gadilu/html$
```

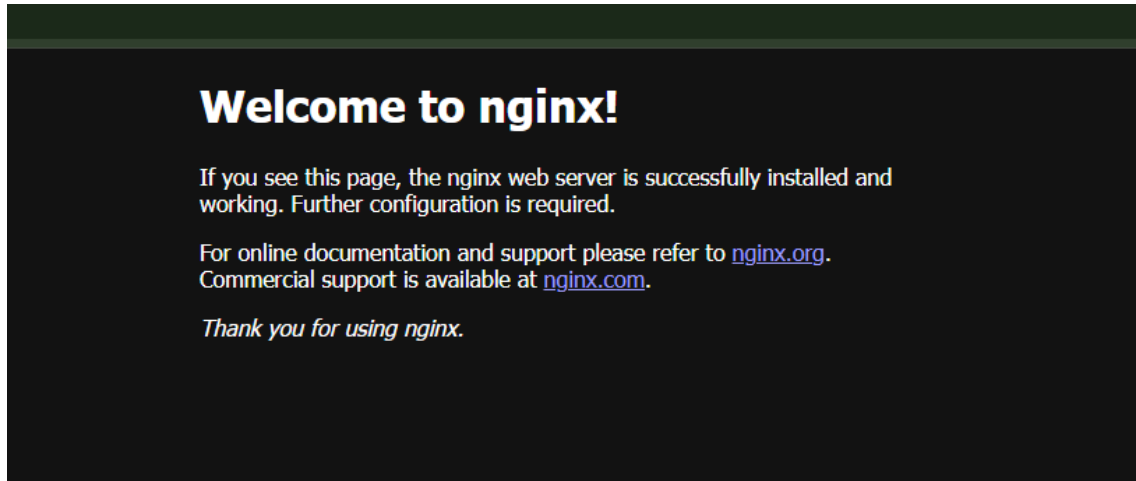
Ahora pondremos el propietario de la carpeta y todo lo que haya dentro, será el usuario a24gadilu que será el usuario del servidor web. Con el siguiente comando: “sudo chown -R www-data:www-data /var/www/a24gadilu/html”

```
a24gadilu@debian:/var/www/a24gadilu/html$ sudo chown -R a24gadilu:a24gadilu /var/www/a24gadilu/html
[sudo] contraseña para a24gadilu:
a24gadilu@debian:/var/www/a24gadilu/html$
```

Y le daremos permisos para que no nos de error de acceso no autorizado al entrar al sitio web. Con el siguiente comando: “sudo chmod -R 755 /var/www/nombre\_web”

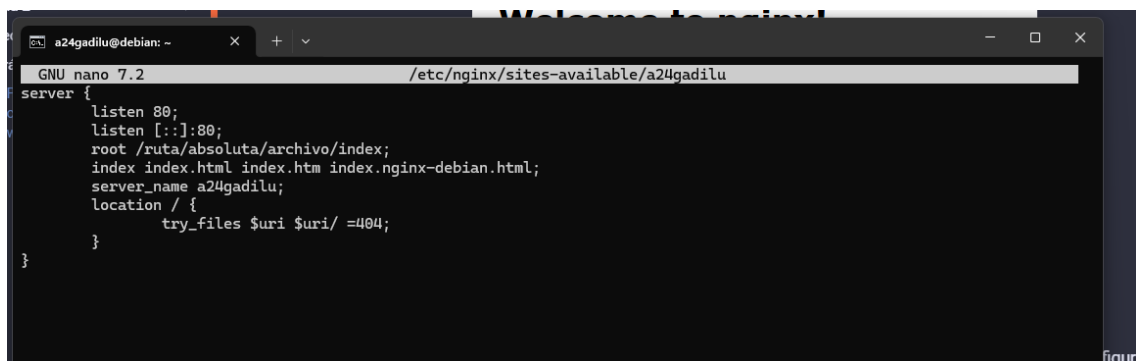
```
a24gadilu@debian:/$ sudo chmod -R 755 /var/www/a24gadilu
a24gadilu@debian:/$
```

Para ver que esta correcta la instalación y configuración deberemos acceder desde nuestro navegador a través de la dirección ip en este caso 192.168.0.2 que es la red) de nuestra máquina.



Ahora configuraremos el servidor para ello deberemos crear un dominio web con nuestro nombre, con el siguiente comando “sudo nano /etc/nginx/sites-available/a24gadilu”.

Deberemos configurarlo en el puerto 80 y ponerle nuestro nombre al servidor.



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

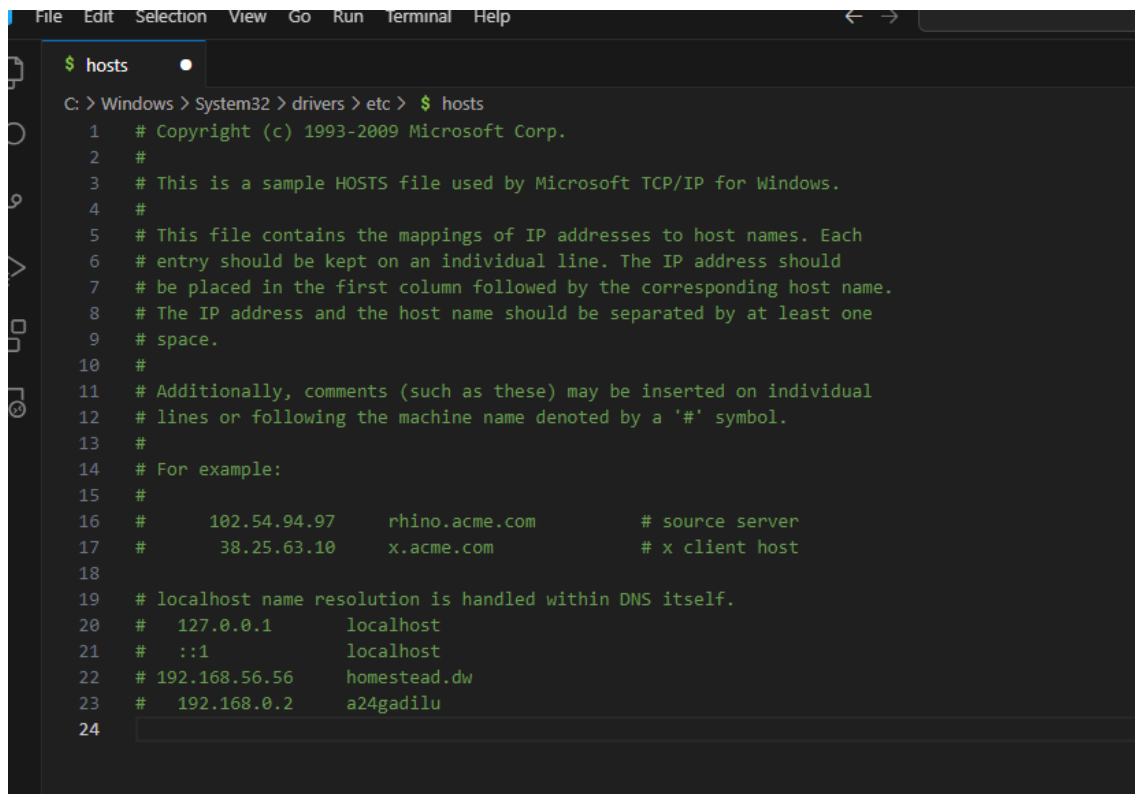
“sudo ln -s /etc/nginx/sites-available/a24gadilu /etc/nginx/sites-enabled/”

```
a24gadilu@debian:~$ sudo ln -s /etc/nginx/sites-available/a24gadilu /etc/nginx/sites-enabled/
```

Una vez creado reiniciamos el servicio con esxte comando: “sudo service nginx restart”

```
a24gadilu@debian:~$ sudo service nginx restart
Restarting nginx: nginx.
a24gadilu@debian:~$
```

Ahora debemos configurar nuestra maquina anfitriona porque no tenemos DNS y para ello deberemos asignar la ip en el archivo C:\Windows\System32\drivers\etc\hosts

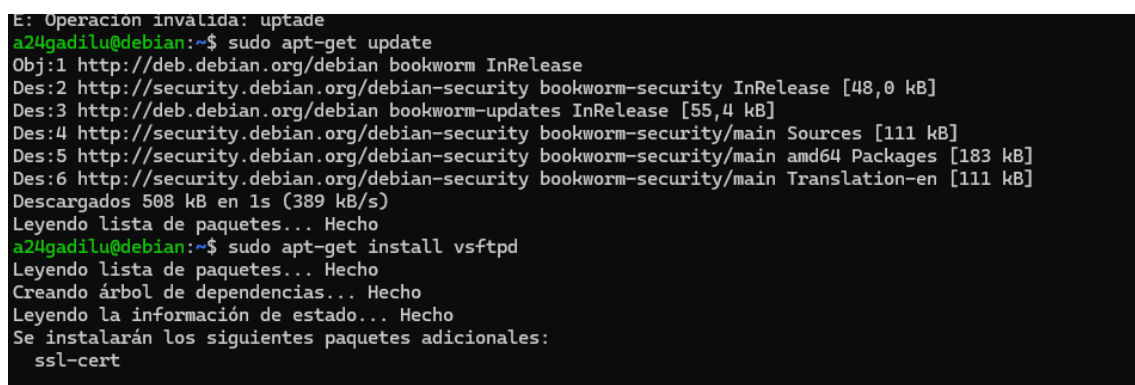


```
File Edit Selection View Go Run Terminal Help
$ hosts
C: > Windows > System32 > drivers > etc > $ hosts
1  # Copyright (c) 1993-2009 Microsoft Corp.
2  #
3  # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4  #
5  # This file contains the mappings of IP addresses to host names. Each
6  # entry should be kept on an individual line. The IP address should
7  # be placed in the first column followed by the corresponding host name.
8  # The IP address and the host name should be separated by at least one
9  # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #         102.54.94.97       rhino.acme.com   # source server
17 #          38.25.63.10      x.acme.com       # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1       localhost
21 #   ::1            localhost
22 # 192.168.56.56     homestead.dw
23 # 192.168.0.2      a24gadilu
24
```

Ahora configuraremos el servidor SFTP en Debian. Actualizamos repositorios e instalamos SFTP.

“sudo apt-get update ”

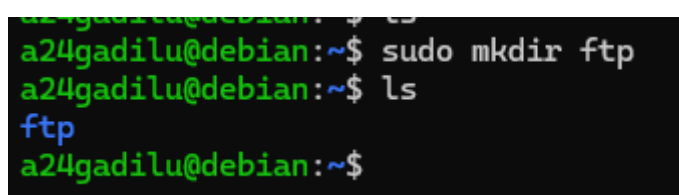
“sudo apt-get install vsftpd ”



```
E: Operación inválida: uptade
a24gadilu@debian:~$ sudo apt-get update
Obj:1 http://deb.debian.org/debian bookworm InRelease
Des:2 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Des:3 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Des:4 http://security.debian.org/debian-security bookworm-security/main Sources [111 kB]
Des:5 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [183 kB]
Des:6 http://security.debian.org/debian-security bookworm-security/main Translation-en [111 kB]
Descargados 508 kB en 1s (389 kB/s)
Leyendo lista de paquetes... Hecho
a24gadilu@debian:~$ sudo apt-get install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
ssl-cert
```

Creamos una carpeta en nuestro /home de Debian:

“mkdir /home/a24gadilu /ftp”



```
a24gadilu@debian:~$ sudo mkdir ftp
a24gadilu@debian:~$ ls
ftp
a24gadilu@debian:~$
```

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

Ahora configuraremos vsftpd :

```
“sudo nano /etc/vsftpd.conf”
```

Debemos eliminar estas líneas:

Y añadir estas:

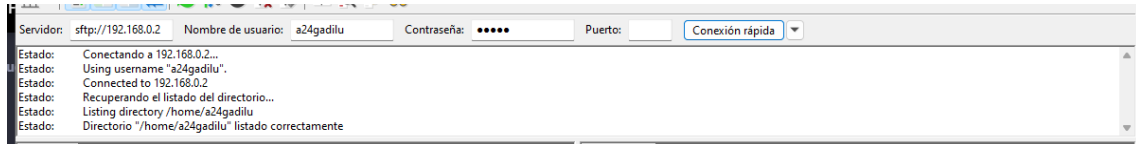
```
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
require_ssl_reuse=NO
ssl_ciphers=HIGH

local_root=/home/a24qadilu/ftp
```

Reiniciamos el servicio: “sudo service vsftpd restart --now”

```
a24gadilu@debian:~/ftp$ sudo nano /etc/vsftpd.conf
a24gadilu@debian:~/ftp$ sudo service vsftpd restart --now
ice vsftpd restart --now
Stopping FTP server: vsftpd.
Starting FTP server: vsftpd.
a24gadilu@debian:~/ftp$
```

Ahora deberemos descargar un cliente de FTP, en este caso hemos elegido Filezilla e introducimos los datos de nuestra maquina virtual.



Una vez conectados ya podemos subir los ficheros que queramos a nuestro servidor.

