

Instal·lació i configuració de servidor web Nginx

Guillermo Vidal Frasquet

Desplegament
d'Aplicacions Web
Pràctica



Continguts

1 Instal·lació i configuració de servidor web Nginx	2
1.1 Instal·lació servidor web Nginx	2
1.2 Creació de les carpeta del lloc web	2
1.3 Configuració de servidor web NGINX	3
1.4 Comprovacions	5
1.4.1 Comprovació del correcte funcionament	5
1.4.2 Comprovar registres del servidor	5
1.5 FTP	6
1.5.1 Com transferir arxius des de la nostra màquina local/amfitrió a la nostra màquina virtual Debian/servidor remot?	6
1.5.2 Configurar servidor SFTP en Debian	6

1 Instal·lació i configuració de servidor web Nginx

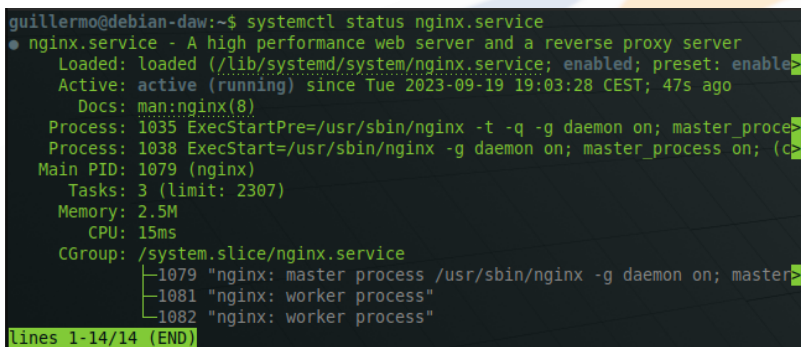
1.1 Instal·lació servidor web Nginx

Per a instal·lar el servidor **Nginx** en la nostra **Debian**, primer actualitzem els repositoris i després instal·lem el paquet corresponent:

```
1 sudo apt update
2 sudo apt install nginx
```

Comprovem que **Nginx** s'ha instal·lat i que està funcionant correctament:

```
1 systemctl status nginx
```



```
guillermo@debian-daw:~$ systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enable>
   Active: active (running) since Tue 2023-09-19 19:03:28 CEST; 47s ago
     Docs: man:nginx(8)
   Process: 1035 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proce>
   Process: 1038 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (c>
   Main PID: 1079 (nginx)
    Tasks: 3 (limit: 2307)
   Memory: 2.5M
      CPU: 15ms
   CGroup: /system.slice/nginx.service
           └─1079 "nginx: master process /usr/sbin/nginx -g daemon on; master>
             └─1081 "nginx: worker process"
               └─1082 "nginx: worker process"

lines 1-14/14 (END)
```

1.2 Creació de les carpeta del lloc web

Igual que ocorre en **Apache**, tots els arxius que formaran part d'un lloc web que servirà **Nginx** s'organitzaran en carpetes. Aquestes carpetes, típicament estan dins de `/var/www`.

Així doncs, crearem la carpeta del nostre lloc web o domini:

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

On el **nom_web** pot ser la paraula que vulgueu, sense espais.

Ací, dins d'aquesta carpeta HTML, heu de clonar el següent repositori:

<https://github.com/cloudacademy/static-website-example>

A més, farem que el propietari d'aquesta carpeta i tot el que hi haja dins siga l'usuari `www-data`, típicament l'usuari del servei web.

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

I li donarem els permisos adequats perquè no ens de un error d'accés no autoritzat en entrar en el lloc web:

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

Per a comprovar que el servidor està funcionant i servint pàgines correctament, podeu accedir des del vostre client a:

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

I us haurà d'aparèixer una cosa així:

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.

El que demostra que tot és correcte fins ara.

1.3 Configuració de servidor web NGINX

En **Nginx** hi ha dues rutes importants. La primera d'elles és `sites-available`, que conté els arxius de configuració dels **hosts virtuals** o blocs disponibles en el servidor. És a dir, cadascun dels llocs webs que alberga el server. L'altra és `sites-enabled`, que conté els arxius de configuració dels llocs habilitats, és a dir, els que funcionen en eixe moment.

Dins de `sites-available` hi ha un arxiu de configuració per defecte (**default**), que és la pàgina que es mostra si accedim al servidor sense indicar cap lloc web o quan el lloc web no és trobat en el servidor (a causa d'una mala configuració per exemple). Aquesta és la pàgina que ens ha aparegut en l'apartat anterior.

Perquè **Nginx** presente el contingut de la nostra web, és necessari crear un bloc de servidor amb les directives correctes. En comptes de modificar l'arxiu de configuració predeterminat directament, crearem un nou en `/etc/nginx/sites-available/nom_web`:

```
1 sudo nano /etc/nginx/sites-available/nom_web
```

I el contingut d'aquest arxiu de configuració:

```
1 server {
2     listen 80;
3     listen [::]:80;
4     root /ruta/absoluta/archivo/index;
5     index index.html index.htm index.nginx-debian.html;
6     server_name nombre_web;
7     location / {
8         try_files $uri $uri/ =404;
9     }
10 }
```

Ací la directiva **root** ha d'anar seguida de la ruta absoluta on es trobe l'arxiu `index.html` de la nostra pàgina web, que es troba entre tots els que heu descomprimit.

Ací teniu un exemple d'un lloc webs amb la seua ruta (directoris que hi ha) abans de l'arxiu `index.html`:



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

I crearem un arxiu simbòlic entre aquest arxiu i el de llocs que estan habilitats, perquè es done d'alta automàticament.

```
1 sudo ln -s /etc/nginx/sites-available/nom_web /etc/nginx/sites-enabled/
```

I reiniciem el servidor per a aplicar la configuració:

```
1 sudo systemctl restart nginx
```

1.4 Comprovacions

1.4.1 Comprovació del correcte funcionament

Com encara no posseïm un servidor **DNS** que tradueixi els noms a **IPs**, hem de fer-ho de manera manual. Editarem l'arxiu `/etc/hosts` de la nostra màquina amfitriona perquè associi la IP de la màquina virtual, al nostre `server_name`.

Aquest arxiu, en Linux, està en: `/etc/hosts`

I en Windows: `C:\Windows\System32\drivers\etc\hosts`

I haurem d'afegir-li la línia:

```
192.168.X.X nom_web
```

on heu de substituir la **IP** per la qual tinga la vostra màquina virtual.

1.4.2 Comprovar registres del servidor

Comproveu que les peticions s'estan registrant correctament en els arxius de logs, tant les correctes com les errònies:

- `/var/log/nginx/access.log`: cada sol·licitud al seu servidor web es registra en aquest arxiu de registre, llevat que Nginx estiga configurat per a fer una cosa diferent.
- `/var/log/nginx/error.log`: qualsevol error de Nginx s'assentarà en aquest registre.



Si no us apareix res en els logs, podria passar que el navegador ha escorcollat la pàgina web i que, per tant, ja no està obtenint la pàgina del navegador sinó de la pròpia memòria. Per a solucionar això, podeu accedir amb la manera privada del navegador i ja us hauria de registrar aqueixa activitat en els logs.

1.5 FTP

Si volem tindre diversos dominis o llocs web en el mateix servidor Nginx (és a dir, que tindran la mateixa IP) hem de repetir tot el procés anterior amb el nou nom de domini que vulguem configurar.

1.5.1 Com transferir arxius des de la nostra màquina local/amfitrió a la nostra màquina virtual Debian/servidor remot?

Hui dia el procés més senzill i segur és a través de **Github** com hem vist abans. No obstant això, el currículum de la **Conselleria d'Educació** m'obliga a ensenyar-vos un mètode una mica obsolet hui dia, així que anem a això, us presente al **FTP**.

El FTP és un protocol de transferència d'arxius entre sistemes connectats a una xarxa TCP. Com el seu nom indica, es tracta d'un protocol que permet transferir arxius directament d'un dispositiu a un altre. Actualment, és un protocol que a poc a poc va abandonant-se, però ha estat vigent més de 50 anys.

El protocol FTP tal qual és un protocol insegur, ja que la seua informació no viatja xifrada. Tanmateix, en 2001 això es va solucionar amb el protocol **SFTP**, que li afegia una capa SSH per a fer-ho més segur i privat.

SFTP no és més que el mateix protocol FTP però implementat per un canal segur. Són les sigles de SFTP, Secure File Transfer Protocol, i consisteix en una extensió de Secure Shell Protocol (SSH) creada per a poder fer transmissions d'arxius.

La seguretat que ens aporta SFTP és important per a la transferència d'arxius perquè, si no disposem d'ella, els arxius viatjaran tal qual per la xarxa, sense cap mena d'encryptació. Així doncs, usant FTP tradicional, si algun agent aconseguís escoltar les transferències, podria ocórrer que la informació quedara al descobert. Això seria especialment important si els arxius que pugem contenir informació confidencial o dades personals.

Atés que usar SFTP aporta major seguretat a les transmissions, és recomanable utilitzar-ho, més encara sabent que realment no hi ha molta dificultat a establir les connexions pel protocol segur.

1.5.2 Configurar servidor SFTP en Debian

En primer lloc, ho instal·larem des dels repositoris:

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


Ara crearem una carpeta en el nostre **home** en Debian:

```
1 mkdir /home/nom_usuari/ftp
```

En la configuració de **vsftpd** indicarem que aquest serà el directori al qual **vsftpd** es canvia després de connectar-se l'usuari.

Ara crearem els certificats de seguretat necessaris per a aportar la capa de xifratge a la nostra connexió (alguna cosa semblant a HTTPS)

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

I una vegada realitzats aquests passos, procedim a realitzar la configuració de **vsftpd** pròpiament dita. Es tracta, amb l'editor de text que més us agrade, d'editar l'arxiu de configuració d'aquest servei, per exemple amb **nano**:

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

En primer lloc, buscarem les següents línies de l'arxiu i les eliminarem per complet:

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

Després d'això, afegirem aquestes línies en el seu lloc

```
1 rsa_cert_file=/etc/ssl/private/vsftpd.pem
2 rsa_private_key_file=/etc/ssl/private/vsftpd.pem
3 ssl_enable=YES
4 allow_anon_ssl=NO
5 force_local_data_ssl=YES
6 force_local_logins_ssl=YES
7 ssl_tlsv1=YES
8 ssl_sslv2=NO
9 ssl_sslv3=NO
10 require_ssl_reuse=NO
11 ssl_ciphers=HIGH
12
13 local_root=/home/nombre_usuario/ftp
```

I, després de guardar els canvis, reiniciem el servei perquè agafe la nova configuració:

```
1 sudo systemctl restart --now vsftpd
```

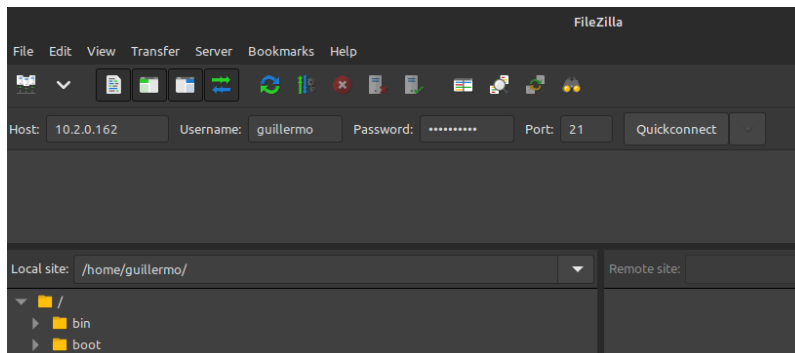
Després d'acabar aquesta configuració, ja podem accedir al nostre servidor mitjançant un client FTP adequat, com per exemple Filezilla de dues formes, a saber:

- Mitjançant el port per defecte del protocol insegur FTP, el 21, però utilitzant certificats que xifren

l'intercanvi de dades convertint-lo així en segur.

- Fent ús del protocol SFTP, dedicat a l'intercanvi de dades mitjançant una connexió similar a SSH, utilitzant el port 22.

Després de descarregar **el client FTP** en el nostre ordinador, introduïm les dades necessàries per a connectar-nos al nostre servidor FTP en Debian:

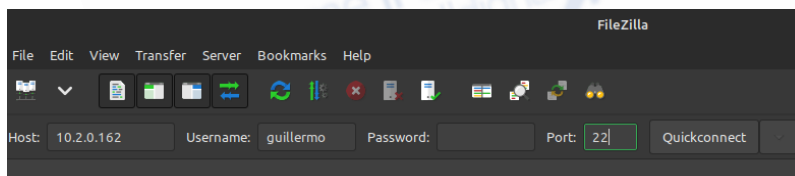


- La IP de Debian
- El nom d'usuari de Debian
- La contrasenya d'aqueix usuari
- El port de connexió, que serà el 21 per a connectar-nos utilitzant els certificats generats prèviament

Després de donar-li al botó de **Connexió ràpida**, ens saltarà un avís a propòsit del certificat, li donem a acceptar perquè no comporta perill ja que ho hem generat nosaltres mateixos.

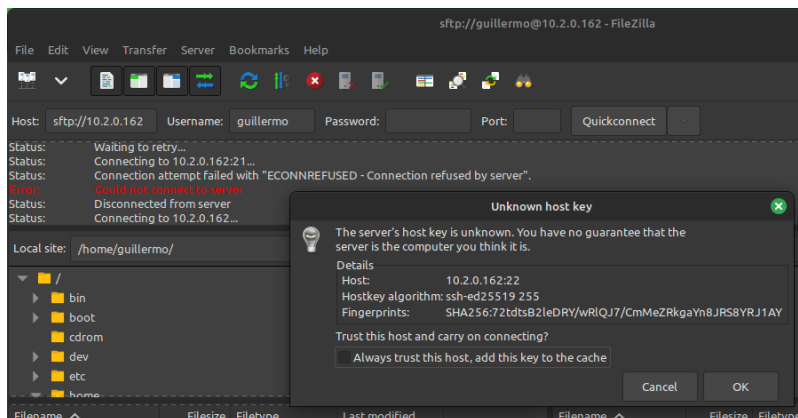
Ens connectarem directament a la carpeta que li havíem indicat en l'arxiu de configuració `/home/guillermo/ftp`

Si el que volguérem és connectar-nos per SFTP, exactament igual de vàlid, faríem:

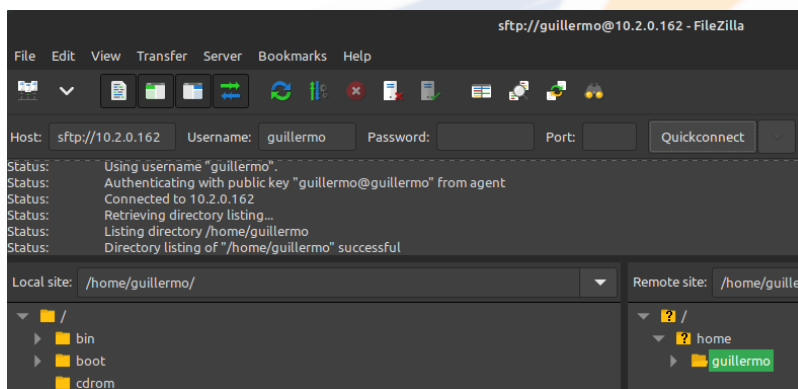


Fixeu-vos que en utilitzar les claus de SSH que ja estem utilitzant des de la Pràctica 1, no s'ha d'introduir la contrasenya, únicament el nom d'usuari.

Com que ens estem connectant usant les claus FTP, ens ix el mateix avís que ens eixia en connectar-nos per primera vegada per SSH a la nostra Debian, que acceptem perquè sabem que no comporta cap perill en aquest cas:



I veiem que a l'èsser una espècie de connexió SSH, ens connecta al **home** de l'usuari, en lloc de la carpeta ftp. A partir d'ací ja procediríem igual que en l'altre cas.



Una vegada connectats, busquem la carpeta del nostre ordinador on hem descarregat el .zip (en la part esquerra de la pantalla) i en la part dreta de la pantalla, busquem la carpeta on volem pujar-la. Amb un doble clic o utilitzant botó dret > pujar, la pugem al servidor.

Recordem que hem de tindre el nostre lloc web en la carpeta **/var/www** i donar-li els permisos adequats, de forma pareguda a com hem fet amb l'altre lloc web.

El comando que ens permet descomprimir un .zip en un directori concret és:

```
1 unzip archivo.zip -d /nombre/directorio
```

Si no teniu **unzip** instal·lat, ho instal·leu:

```
1 sudo apt update && sudo apt install unzip
```



Tasca

Configura un nou domini (nom web) per al **.zip** amb el nou lloc web que us proporcionat. **En aquest cas heu de transferir els arxius a la vostra Debian mitjançant SFTP.**