Servidor de Ficheros

Preguntas

1. ¿En qué consiste FTP? ¿y SFTP? ¿y TFTP? ¿Qué puertos y protocolo en la capa de transporte utilizan?

FTP (*File Transfer Protocol*) es un protocolo de transferencia de ficheros. Utiliza los puertos TCP 21 (para el envío de comandos) y 20 (para la transferencia de los datos de respuesta).

SFTP (*SSH FTP*) es una versión segura de FTP que cifra la información utilizando el protocolo SSH. Utiliza entonces el puerto TCP 22. También existe FTPS (*SSL FTP*), que es una versión segura de FTP que cifra la información utilizando SSL. Utiliza los puertos TCP 990 (para el envío de comandos) y 989 (para la transferencia de los datos de respuesta).

TFTP (*Trivial FTP*) es una versión muy sencilla de FTP, con sólo dos comandos: get y put. Utiliza el puerto UDP 69.

A diferencia de SMB y NFS, que se utilizan más en entornos de red local, el protocolo FTP se suele utilizar para transferencia de ficheros entre ordenadores remotos. A diferencia de BitTorrent, que es un protocolo de transferencia P2P, FTP tiene arquitectura cliente-servidor.

2. ¿En qué consiste SAMBA (SMB/CIFS)? ¿Qué puertos y protocolo en la capa de transporte utilizan?

Samba es la implementación en software libre del protocolo de transferencia de ficheros SMB o CIFS (*Server Message Block* / *Common Internet File System*), típicamente utilizado en las redes locales de Microsoft Windows para compartir ficheros e impresoras. Utiliza los puertos UDP 137 y 138, y TCP 139 y 445.

Con Samba desde Linux podemos acceder a recursos compartidos de ordenadores Windows, podemos compartir nuestras carpetas e impresoras con ordenadores Windows, nos podemos conectar a un servidor de dominio, o actuar nosotros como servidor de dominio Windows, e incluso actuar como servidor de Active Directory.

3. ¿En qué consiste NFS? ¿Qué puertos y protocolo en la capa de transporte utilizan?

NFS (*Network File System*) es un protocolo de transferencia de ficheros, típico en sistemas Unix. Utiliza el puerto UDP/TCP 2049.

Un escenario típico de uso es una red local donde los usuarios tienen su carpeta personal /home en un ordenador remoto accesible desde cualquier ordenador de la red, de tal manera que pueden "loggearse" en cualquier ordenador de la red teniendo siempre disponible su perfil personal.

4. ¿Que ventajas ofrecen ownCloud o Alfresco sobre los servidores de ficheros tradicionales?

Alfresco (http://www.alfresco.com/community/) , al igual que Nuxeo o Knowledgetree, es un gestor documental accesible vía web. Permite organizar los documentos generados en una empresa. Para dicha tarea, ofrece muchas más opciones que un servidor de ficheros. Sin embargo, ni de lejos tienen la velocidad y potencia de un servidor de ficheros. No es más que un gestor de contenidos.

OwnCloud (http://owncloud.org/) , al igual que Google Drive o Dropbox, es un gestor de contenido para almacenar ficheros en la nube, e incluso crear nuestra propia nube. Permite acceder a dichos ficheros a través de un navegador. Tampoco tiene la velocidad y potencia de un servidor de ficheros. No es más que un gestor de contenidos.

5. Imagina el caso de un servidor web donde varios usuarios tienen páginas. Para que los usuarios puedan subir y modificar páginas se instala un servidor de FTP.

a) ¿Necesitarán nombre de usuario y contraseña para entrar?

b) ¿Una vez dentro a qué carpeta apuntará el servidor FTP?

c) ¿Tendrán permisos de escritura o sólo permisos de lectura en dicha carpeta?

a) Sí. b) A la carpeta personal del usuario. c) Escritura y lectura.

Ahora, imagina el caso de un servidor FTP para que cualquier persona pueda descargar ficheros

d) ¿Necesitarán nombre de usuario y contraseña para entrar?

e) ¿Una vez dentro a qué carpeta apuntará el servidor FTP?

f) ¿Tendrán permisos de escritura o sólo permisos de lectura en dicha carpeta?

d) No. e) A la carpeta raiz donde estén los ficheros. f) Sólo lectura.

6. Viene algún software de servidor de FTP con Windows Server? ¿Desde dónde se instala y desde dónde se administra? ¿Y para Windows XP?

Con Windows Server viene el Internet Information Server (IIS). Los usuarios de dicho servicio deben existir usuarios del sistema. Si queremos más funcionalidad, disponemos soluciones más flexibles e incluso gratuitas, como por ejemplo *Filezilla FTP Server*.

Se instala desde "Agregar y quitar programas → Agregar y quitar componentes de Windows → Servidor de aplicaciones → Instalar Internet Information Services → Servicio World Wide Web → Servicio World Wide Web".

Se administra desde "Herramientas Administrativas → Internet Information Services"

7. ¿Cuál es el software de servidor de FTP más conocido para Unix/Linux?

Hay varios bastante buenos. No hay uno que sea el más conocido. Están *pure-FTPd*, *proFTPd*, *vsFTPd*, *wu-FTPd*, etc: http://en.wikipedia.org/wiki/List_of_FTP_server_software

8. Al instalar un servidor de FTP, ¿Qué parámetros a configurar piensas que serán los más importantes?

- Puerto por el que escucha.

- Límites del números de conexiones, de capacidad de transferencias, de ancho de banda, de tiempo sin transferir datos, ...

- Lista de usuarios, con sus contraseñas, y sus correspondientes carpetas.

- Permisos de los usuarios en las carpetas.

Aviso: si el acceso del usuario falla, comprueba que la carpeta que les asignaste existe, que el usuario es el propietario de dicha carpeta, y que tiene permisos correctos para las operaciones

sobre dicha carpeta. Además, en sistemas Unix donde los usuarios de FTP sean usuarios del sistema, comprueba que le asignaste una shell válida. Por último, comprueba que el usuario esté "encerrado" en la carpeta y no pueda "escapar" subiendo a carpetas de más alto nivel para navegar por todo el sistema de ficheros.

9. En un cliente de FTP que no utiliza interfície gráfica (ventanas, carpetas, iconos para ficheros, ...), sino línea de comandos ¿Cuáles son los comandos "imprescindibles" y para qué sirven?

- *open*: inicia una conexión.

- *close*: cierra una conexión.

- *quit*: finaliza el cliente.

- *dir*: lista el directorio en el servidor.

- *cd* y *lcd*: cambia de directorio en el servidor y en el cliente, respectivamente.

- *put* y *mput*: transfiere al servidor uno o múltiples archivos del cliente, respectivamente.

- *get* y *mget*: transfiere al cliente uno o múltiples archivos del servidor, respectivamente.

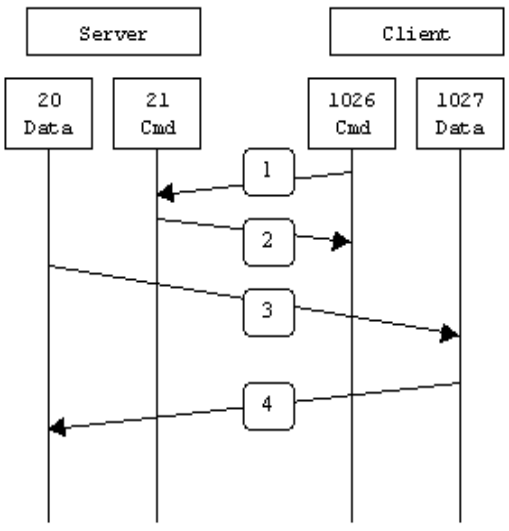
http://es.wikipedia.org/wiki/File_Transfer_Protocol#Gu.C3.ADa_de_comandos_FTP

10. ¿Cómo funciona FTP en modo activo? ¿Cómo funciona FTP en modo pasivo?

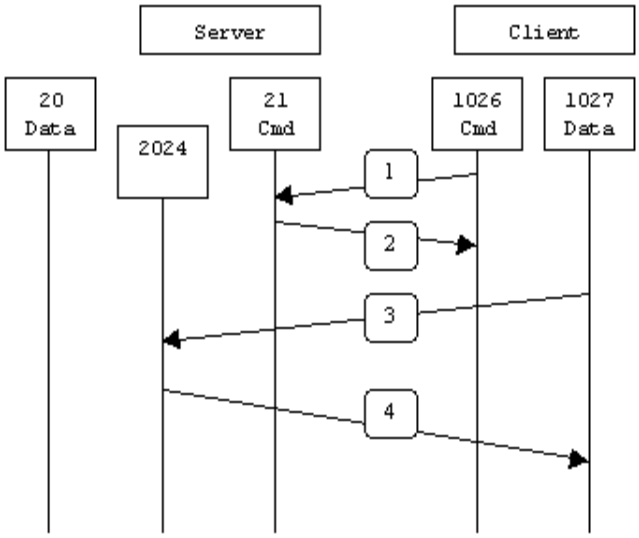
FTP permite dos modos de transferencia de datos entre el cliente y el servidor:

- Activo: el cliente envía un comando hacia el puerto 21 del servidor e indica qué puerto superior al 1024 abrirá para que el servidor le envíe los datos. El servidor desde su puerto 20 inicia una conexión contra el cliente hacia el puerto especificado. ¡Cuidado con NAT!

- Pasivo: el cliente envía un comando hacia el puerto 21 del servidor. El servidor responde especificando a qué nuevo puerto se debe conectar el cliente para recibir los datos. El cliente inicia una conexión contra el servidor hacia el puerto especificado.



FTP activo



FTP pasivo

Más información sobre los modos activo y pasivo en http://slacksite.com/other/ftp.html.

Datos de la práctica

Configuraremos nuestro servidor FTP para que los usuarios Ana y Bob puedan subir las páginas de las dos webs que instalamos en la práctica de HTTP: www.mired.org y intranet.mired.org, respectivamente.

También daremos acceso anónimo a una carpeta para lectura de documentos pero sin escritura.

Práctica con Windows

Instalaremos Internet Information Server para Windows Server. Exploraremos la interfaz gráfica de administración del servidor FTP, configurando los parámetros básicos para una pequeña red local, y probaremos el servidor. También podemos probar a instalar y configurar Filezilla FTP Server, que es más flexible, gratuito, y esta disponible para otras versiones de Windows.

Configuraremos el servidor web para que aloje páginas de diferentes usuarios, en una carpeta propia cada uno. A continuación configuraremos el servidor FTP para que dichos usuarios existan y al acceder con su nombre de usuario y contraseña el servidor FTP les lleve automáticamente a su carpeta.

Paso a paso en Windows Server 2016:

- https://proyectopub.wordpress.com/2018/01/27/configuracion-ftp-windows-server-2016/

- https://www.solvetic.com/tutoriales/article/9768-instalar-ftp-e-iis-en-windows-server-2022/

Práctica con Linux: servidor FTP

1. Primero de todo creamos la carpeta donde irán los ficheros de los usuarios. En el caso de las webs, dichas carpetas ya están creadas, y en el caso de usuarios anónimos podría ser */var/ftp*:

   ```
   # mkdir /var/ftp
   ```

   A continuación vamos a crear usuarios con shell falsa */bin/false* para que puedan entrar a nuestro servidor, sin que puedan acceder al resto del sistema. Alternativamente a */bin/false* podríamos utilizar */usr/bin/nologin* para denegar acceso al sistema:

   ```
   # echo "/bin/false" >> /etc/shells
   ```

   ```
   # groupadd ftp_users
   ```

   El proceso, en general, sería este:

   (1) crearemos una carpeta donde un usuario accederá por ftp;

mkdir /var/ftp/*usuario*

(2) crearemos el usuario con una shell falsa;

useradd -g ftp_users -d /var/ftp/*usuario* -s /bin/false *usuario*

cat /etc/passwd

(3) le pondremos una contraseña al usuario;

passwd *usuario*

(4) y le daremos propiedad y permisos sobre su carpeta.

chown -R *usuario*:ftp_users /var/ftp/*usuario*

chmod -R *755* /var/ftp/*usuario*

Pero debemos modificar dichos pasos para que Ana y Bob accedan a las carpetas donde están sus webs. ¿Te atreves a hacerlo?

2. En el caso de querer instalar el servidor *proFTPd*:

   # **apt install proftpd-basic**

   El instalador nos preguntará si deseamos ejecutar el servidor desde el demonio *inetd* (el servidor ftp solo se carga en memoria cuando existan peticiones) o como un servicio independiente (el servidor ftp esta permanentemente en memoria). Escogeremos el funcionamiento como servicio independiente ya que es más eficiente.

   La configuración del servidor se guardará en el archivo */etc/proftpd/proftpd.conf* . Editaremos el archivo de configuración para indicarle qué usuarios pueden entrar al servidor y cuales no, y también que no será necesaria una shell validada en */etc/shells*:

   `sudo nano /etc/proftpd/proftpd.conf`

```
# Por si la shell no es válida

RequireValidShell off

# Para "encerrar" los usuarios en su directorio personal
# ( otro ejemplo:   DefaultRoot /var/ftp/%u )

DefaultRoot ~

# Ejemplo en caso que quieras usuario anónimo

<Anonymous /var/ftp/>
  User ftp
  Group nogroup
  UserAlias anonymous ftp
  <Directory *>
    <Limit WRITE>
      DenyAll
    </Limit>
  </Directory>
</Anonymous>


# No hace falta la configuración que sigue, ya que en /etc/passwd
# ya especificamos a qué directorio acceden los usuarios al loguearse

<Directory /var/www/html/site1/>
```

```
  Umask 022 022
  AllowOverwrite on
  <Limit LOGIN>
    AllowUser usuario1
    DenyAll
  </Limit>
</Directory>

<Directory /var/www/html/site2/>
  Umask 022 022
  AllowOverwrite on
  <Limit LOGIN>
    AllowUser usuario2
    DenyAll
  </Limit>
</Directory>
```

Y reiniciamos el servidor para que se apliquen los cambios en la configuración:

```
# systemctl restart proftpd
```

Podemos ver el log del servidor para visualizar quién ha podido entrar correctamente o no, así como las transferencias de archivos:

```
# less /var/log/proftpd/proftpd.log
```

```
# less /var/log/proftpd/xferlog
```

3. En el caso de querer instalar el servidor *vsFTPd*:

```
# apt install vsftpd
```

La configuración del servidor se guardará en el archivo */etc/vsftpd.conf* . Editaremos el archivo de configuración para indicarle qué usuarios pueden entrar al servidor y cuales no, y también que no será necesaria una shell validada en */etc/shells*:

```
# nano /etc/vsftpd.conf
```

```
check_shell=NO
chroot_local_user=YES
local_enable=YES
local_write_enable=YES
local_umask=022

anonymous_enable=YES
anon_root = /var/ftp
anon_write_enable=NO
no_anon_password=YES

xferlog_enable=YES
```

Y reiniciamos el servidor para que se apliquen los cambios en la configuración:

```
# systemctl restart vsftpd
```

Podemos ver una explicación de todas las opciones del fichero de configuración con:

```
$ man vsftpd.conf
```

Podemos ver el log del servidor para visualizar quién ha podido entrar correctamente o no, así como las transferencias de archivos:

```
# less /var/log/vsftpd.log
```

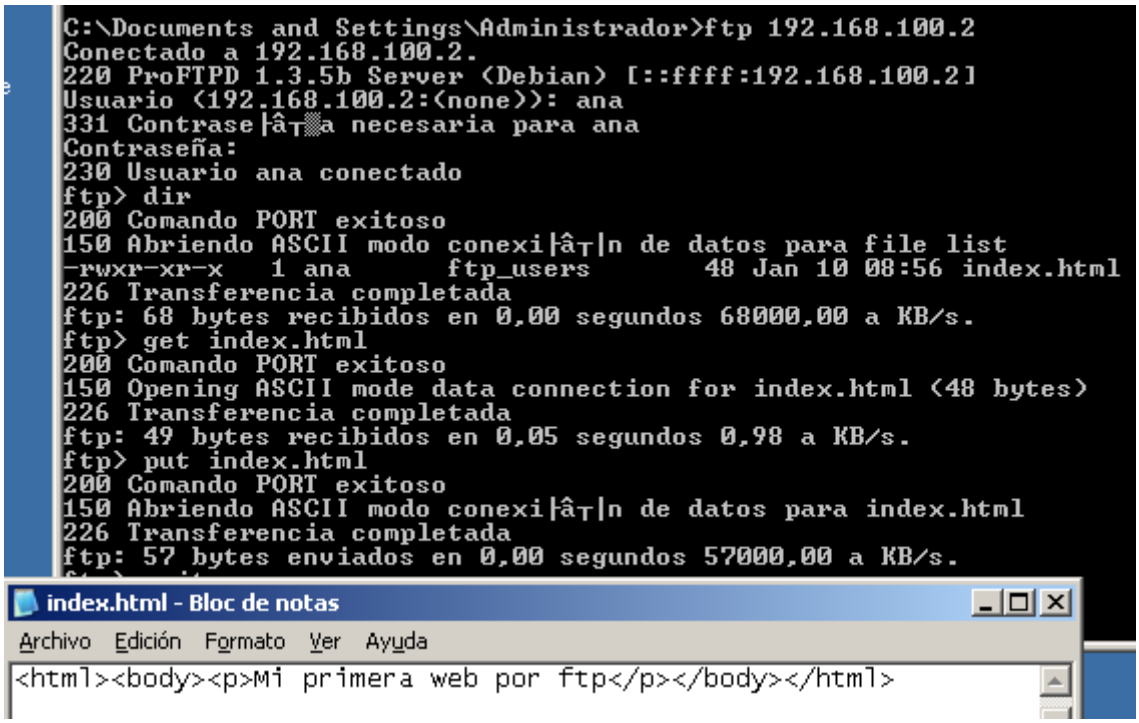
4. Comprueba que los servicios están escuchando en sus puertos correspondientes (tcp/21):

   `# ss -tplna`

   Y prueba el servicio des de la línea de comandos de un cliente:

   `$ ftp IP_servidor`

   O todavía mejor, prueba el servicio navegando des de el popular cliente de FTP *Filezilla*.



5. Si quisieramos activar certificados para *proFTPd* o *vsFTPd*, consulta estos tutoriales:

   https://www.howtoforge.com/how-to-install-proftpd-with-tls-on-ubuntu-1804/

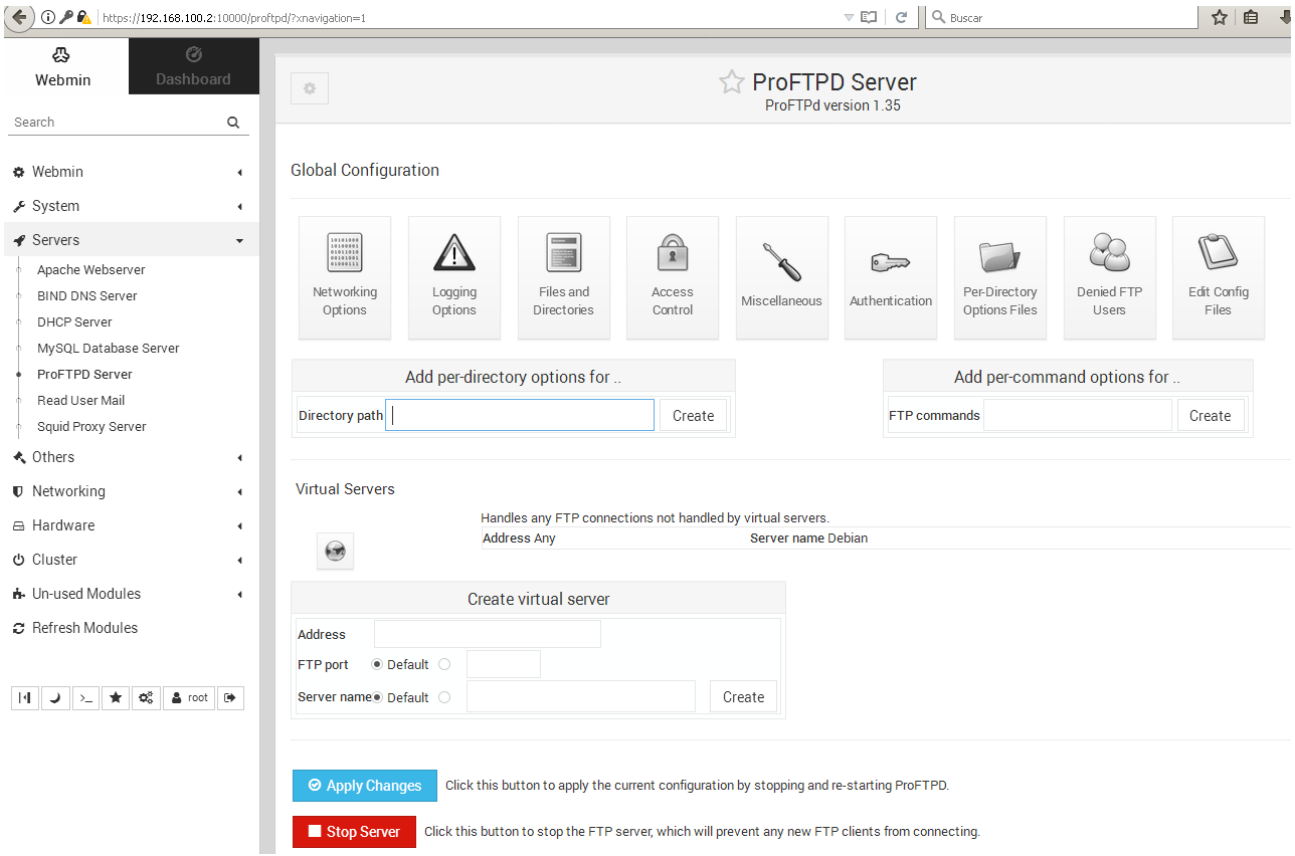
   https://www.howtoforge.com/tutorial/ubuntu-vsftpd/

6. Si queremos configurar el proxy-caché desde un entorno gráfico instalaremos el módulo de *webmin* correspondiente:

   https://*IP_servidor*:10000/ → Un-used modules → proFTPd Server

   https://*IP_servidor*:10000/ → Refresh modules

   https://*IP_servidor*:10000/ → Servers → proFTPd Server



Práctica adicional con Linux: SFTP al servidor en Oracle Cloud

1. Primero entramos por SSH al servidor para instalar SFTP (en realidad ya estaba instalado):

   `$ ssh ubuntu@IPservidor -i clave_privada_servidor`

   `ubuntu@IPservidor:~$ sudo apt update`

   `ubuntu@IPservidor:~$ sudo apt install openssh-sftp-server`

   No nos hace falta abrir el cortafuegos personal ni el cortafuegos de red, ya que funciona sobre el puerto 22/TCP por el que ya teníamos SSH funcionando en dicho servidor.

2. Su fichero de configuración es el mismo que el del servidor SSH, y permitirá acceder por SFTP a los usuarios que ya tenían acceso por SSH y con el mismo modo de autentificación que tienen con SSH.

   `ubuntu@IPservidor:~$ cat /etc/ssh/sshd_config`

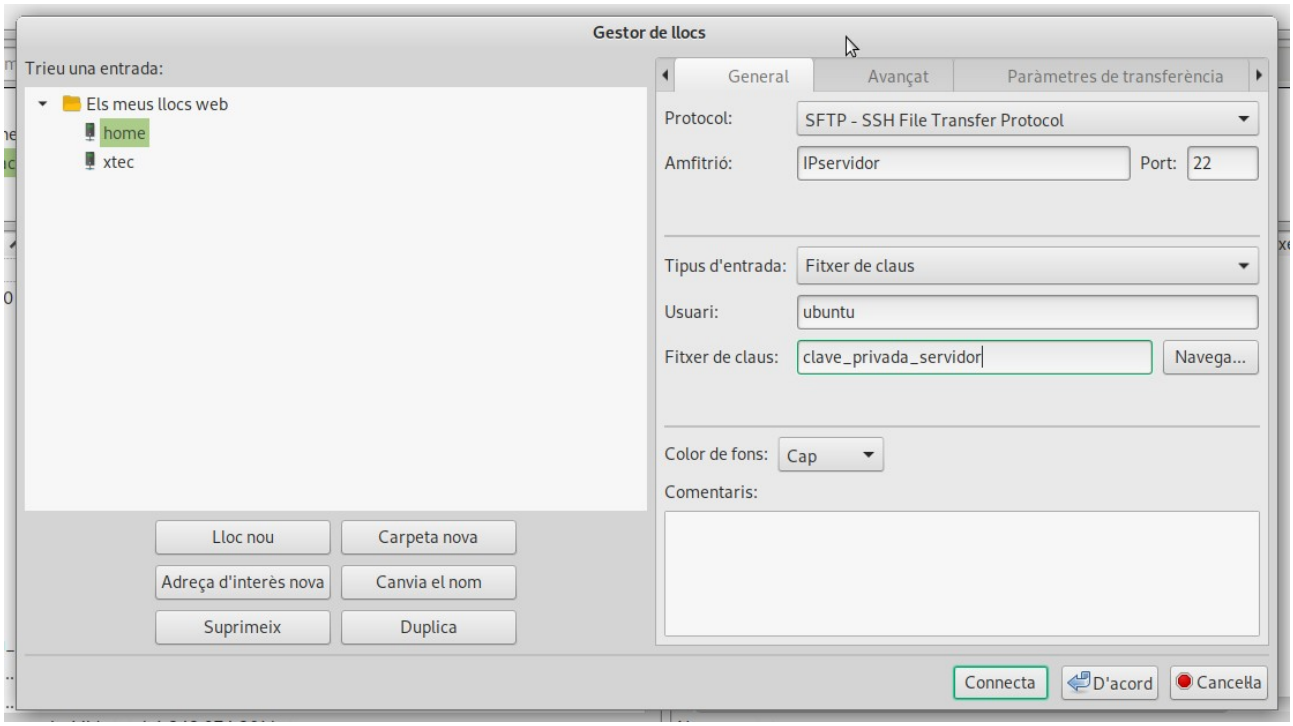
3. Para acceder remotamente, desde línea de comandos se hace de la misma manera que para acceder por SSH, pero el cliente será `sftp`:

   `$ sftp ubuntu@IPservidor -i clave_privada_servidor`

   `sftp> mkdir prova`

```
sftp> exit
```

4. Para acceder con “Filezilla” la configuración sería:



He visto algunas personas que se han encontrado con problemas para acceder a Oracle Cloud por SFTP des del cliente Filezilla. Yo solucioné dicho problema añadiendo la siguiente línea al fichero de configuración del servidor:

```
ubuntu@IPservidor:~$ sudo nano /etc/ssh/sshd_config
```

```
PubkeyAcceptedAlgorithms +ssh-rsa
```

```
ubuntu@IPservidor:~$ sudo systemctl restart sshd
```

5. Puedes comprobar que sólo con instalarlo ya funciona pero que el usuario que accede no está limitado a su carpeta inicial, sino que puede salir de ella para navegar por todo el sistema. Sería conveniente “enjaularlo” en su carpeta. O quizás queramos usuarios de SFTP que no tengan accesso a la línea de comandos. Para ello tenemos que hacer dos cosas:

   a) Crea los usuarios como has hecho en el punto 1 del anterior ejercicio (FTP), pero para que funcione el enjaulado en SFTP esta vez *root* debe ser propietario de las carpetas raíz de los usuarios, que tendrán permisos 755:

   ```
   # chown root /var/ftp/usuario
   # chmod 755 /var/ftp/usuario
   ```

   Pero si el propietario es *root* y los permisos son 755, entonces los usuarios no podrán subir ficheros a menos que les creemos una carpeta dentro con permisos de escritura. Por ejemplo:

   ```
   # mkdir /var/ftp/usuario/subidas/
   # chmod 777 /var/ftp/usuario/subidas
   ```

   b) Editamos el fichero de configuración del servidor SSH/SFTP:

   ```
   # nano /etc/ssh/sshd_config
   ```

```
#Subsystem   sftp   /usr/lib/openssh/sftp-server
Subsystem   sftp   internal-sftp
Match Group ftp_users
  ChrootDirectory %h
  ForceCommand internal-sftp
  X11Forwarding no
  AllowTCPForwarding no
```

```
# systemctl restart sshd
```

Práctica con Linux: servidor NextCloud para ficheros en la nube

- https://www.howtoforge.com/how-to-install-nextcloud-on-debian-11/
- https://computingforgeeks.com/how-to-install-and-configure-nextcloud-on-debian/

Práctica con Linux: NAS

Instalar y crear usuarios con carpetas compartidas mediante FTP, NFS o SMB utilizando una distribución especializada en NAS como *FreeNAS* o *OpenMediaVault*. Dichas distribuciones se administran mediante interfaz web.

FreeNas:

- https://www.youtube.com/watch?v=2lW5Uz8k4u4
- https://www.youtube.com/watch?v=QgTBUQ6C2ZY

OpenMediaVault:

- https://www.youtube.com/watch?v=X6y85dEDYoE
- https://www.youtube.com/watch?v=M_oxzpvMPTE

Referencias

- http://en.wikipedia.org/wiki/File_Transfer_Protocol
- http://en.wikipedia.org/wiki/Network_File_System_(protocol)
- http://en.wikipedia.org/wiki/Server_Message_Block
- http://en.wikipedia.org/wiki/Samba_(software)