

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 “loguearse” 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 raíz 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?

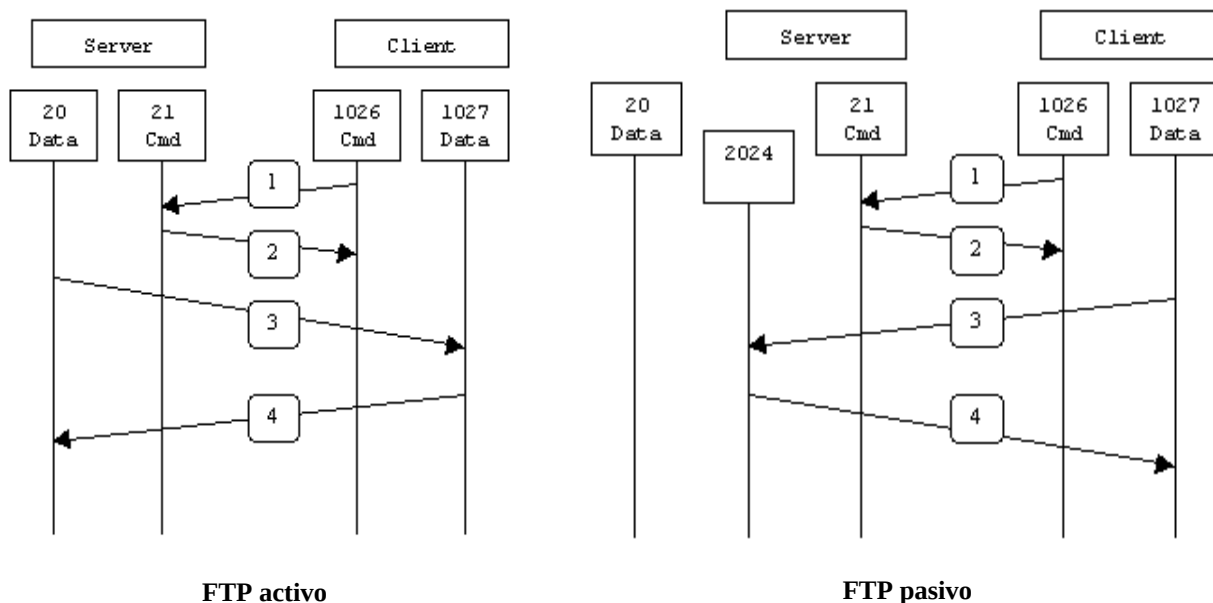
- **open**: inicia una conexión.
- **close**: cierra una conexión.
- **quit**: finaliza el cliente.
- **dir**: lista el directorio en el servidor.
- **cd** y **lcd**: cambia de directorio en el servidor y en el cliente, respectivamente.
- **put** y **mput**: transfiere al servidor uno o múltiples archivos 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.



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/3213-instalar-configurar-servidor-ftp-windows-server/>

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
```

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

```
RequireValidShell off
```

```
DefaultRoot ~
```

```
# 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
  AllowOverride 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
```

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

local_enable=YES
local_write_enable=YES
local_umask=022
check_shell=NO
chroot_local_user=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. En el caso de querer instalar un servidor de STFP:

```
# apt install openssh-sftp-server
```

La configuración del servidor se guardará en el archivo */etc/ssh/sshd_config*. Puedes ver opciones con:

```
$ man ssh_config
```

Si realizamos cambios en la configuración, debemos reiniciar el servidor para que se apliquen:


```
# systemctl restart sshd
```

5. Comprueba que los servicios están escuchando en sus puertos correspondientes (ftp 21, sftp 22):

```
# ss -tlna
```

Y prueba el servicio navegando por: ftp://IP_servidor/

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



```
C:\Documents and Settings\Administrador>ftp 192.168.100.2
Conectado a 192.168.100.2.
220 ProFTPD 1.3.5b Server <Debian> [::ffff:192.168.100.2]
Usuario <192.168.100.2:(none)>: ana
331 Contrase[â]a necesaria para ana
Contraseña:
230 Usuario ana conectado
ftp> dir
200 Comando PORT exitoso
150 Abriendo ASCII modo conexi[â]n de datos para file list
-rwxr-xr-x  1 ana      ftp_users      48 Jan 10 08:56 index.html
226 Transferencia completada
ftp: 68 bytes recibidos en 0,00 segundos 68000,00 a KB/s.
ftp> get index.html
200 Comando PORT exitoso
150 Opening ASCII mode data connection for index.html (48 bytes)
226 Transferencia completada
ftp: 49 bytes recibidos en 0,05 segundos 0,98 a KB/s.
ftp> put index.html
200 Comando PORT exitoso
150 Abriendo ASCII modo conexi[â]n de datos para index.html
226 Transferencia completada
ftp: 57 bytes enviados en 0,00 segundos 57000,00 a KB/s.
```

index.html - Bloc de notas

Archivo Edición Formato Ver Ayuda

```
<html><body><p>Mi primera web por ftp</p></body></html>
```

6. Si quisieramos activar certificados para *proFTPD* o *vsFTPD*, consulta estos tutoriales:

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

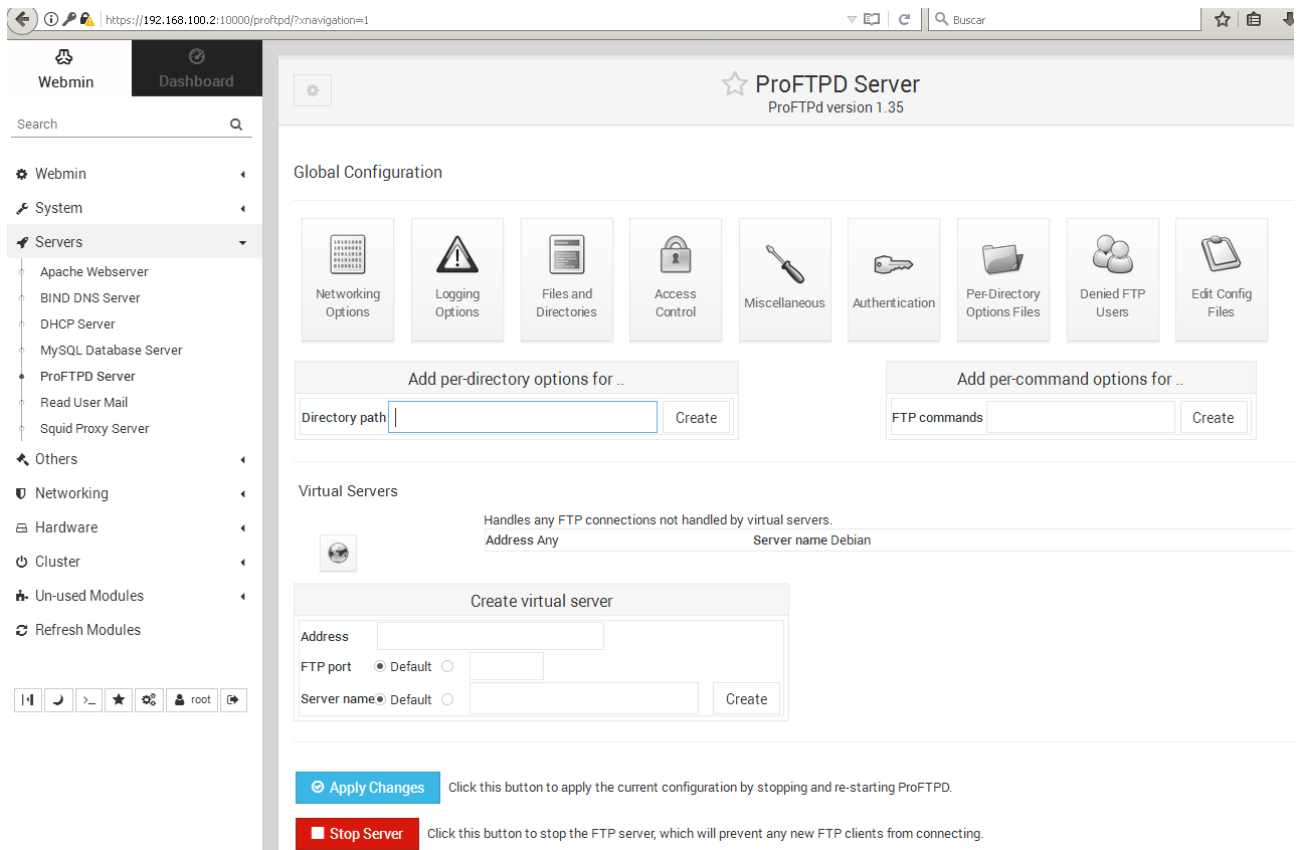
<https://www.howtoforge.com/tutorial/ubuntu-vsftpd/>

7. 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

1. Primero instalaremos el servidor 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 la misma manera de autenticación 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:

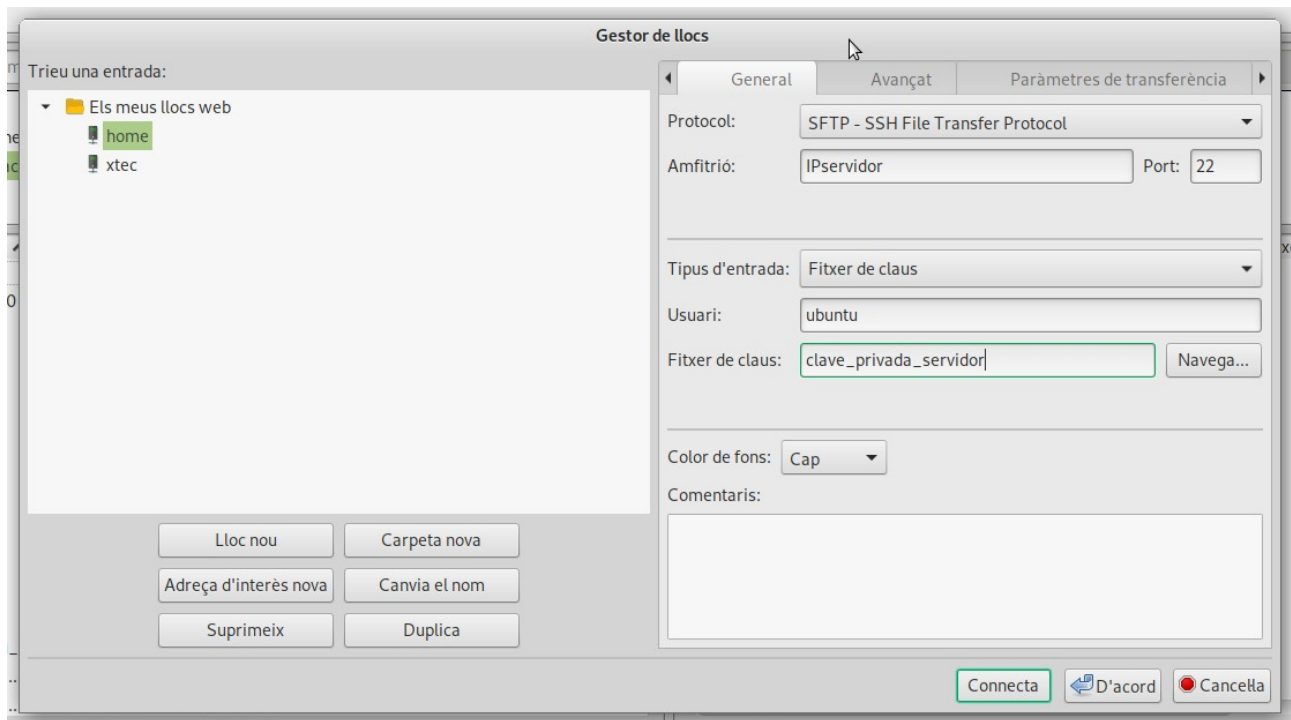
```
$ sftp ubuntu@IPservidor -i clave_privada_servidor
```

```
sftp> mkdir prova
```



```
sftp> exit
```

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



5. Puedes comprobar que inicialmente 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: <https://www.techrepublic.com/article/how-to-use-sftp-with-a-chroot-jail/>

Práctica adicional con Linux: TFTP para arranque por red (obsoleto)

Instalaremos un servidor de PXE de arranque por red, que permitirá la instalación o ejecución de sistemas operativos por red mediante un menú. Para ello necesitaremos instalar y configurar los siguientes servicios: DHCP, TFTP y NFS. El servicio DHCP/BOOTP no interferirá con los otros servidores DHCP ya instalados en la red.

El proceso de arranque en red es el siguiente:

- Los clientes reciben por DHCP los parámetros de red, la IP del servidor BOOTP que sirve la imagen de arranque en red, y el nombre de la imagen de red.
- Los clientes piden la imagen de arranque red a dicho servidor mediante TFTP, la cargan y la ejecutan.
- Al ejecutar la imagen de arranque en red, ésta carga y ejecuta un núcleo del sistema operativo, y monta el sistema de ficheros mediante NFS. Esto último no sería necesario si lo que arrancamos por red es una imagen de instalación, o un LiveCD, o un sistema operativo

que está en el disco duro.

De momento, como sistemas operativos para cargar, tan sólo he escogido uno: la ISO de la Xarxa de Suport de la Reutilització para la catalogación de equipos, que se encuentra en la web:

<https://github.com/eReuse/workbench/releases/latest>

1. Primero instalaremos el servidor TFTP:

```
# apt install tftpd-hpa
```

La configuración del servidor se guardará en el archivo `/etc/default/tftpd-hpa`. Editaremos el archivo de configuración para indicarle la ubicación del servicio de archivos que publicará el servidor:

```
# nano /etc/default/tftpd-hpa
```

```
TFTP_DIRECTORY="/var/lib/tftpboot"
```

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

```
# systemctl restart tftpd-hpa
```

Podemos comprobar que el servicio está funcionando:

```
# ss -upna | grep tftp
```

2. A continuación instalaremos el servidor de DHCP. Para ello el servidor DHCP deberá tener IP fija. Suponiendo que su tarjeta de red sea `eth0`:

```
# ip addr
```

```
# nano /etc/network/interfaces
```

```
auto eth0
iface eth0 inet static
    address          192.168.100.2
    netmask           255.255.255.0
    gateway           192.168.100.1
    dns-nameservers   8.8.8.8
```

y a continuación reinicia la red y prueba que ésta funciona:

```
# systemctl restart networking
```

```
# ping www.google.es
```

Hecho esto, instalamos y configuramos el servidor DHCP:

```
# apt install isc-dhcp-server
```

```
# man dhcpd.conf
```

```
# nano /etc/dhcp/dhcpd.conf
```

```
ddns-update-style interim;
ignore client-updates;

# Opciones generales para TFTP
allow booting;
allow bootp;

# ámbito de mi red
subnet 192.168.100.0 netmask 255.255.255.0 {
    next-server 192.168.15.2;
```

```
filename "pxelinux.0";
range dynamic-bootp 192.168.100.100 192.168.100.200;
option domain-name-servers 8.8.8.8;
option routers 192.168.100.1;
option broadcast-address 192.168.100.255;
}
```

y a continuación reiniciad el servicio:

```
# systemctl restart isc-dhcp-server
```

si el servicio parece que no funciona, probad si ha arrancado con:

```
# ss -upna
```

```
# tail /var/log/syslog
```

3. Ahora toca instalar el servidor de NFS, que sirve para compartir recursos o carpetas en red:

```
# apt install nfs-kernel-server
```

Para probar que funciona podemos compartir nuestra carpeta principal:

```
# nano /etc/exports
```

```
/home/usuario 192.168.100.0/255.255.255.0(no_root_squash,no_subtree_check,rw)
```

y a continuación reinicia el servicio:

```
# systemctl restart nfs-kernel-server
```

```
# showmount -e 192.168.100.2
```

4. Con estos tres servicios instalados, ahora vamos a descargar la imagen PXE de arranque por red, y dejarla en la carpeta raíz del servidor TFTP:

```
# cd /var/lib/tftpboot/
```

```
# wget http://ftp.es.debian.org/debian/dists/buster/main/installer-i386/current/images/netboot/netboot.tar.gz
```

```
# tar xzvf netboot.tar.gz
```

De dicha imagen nos interesan dos archivos:

- *pxelinux.cfg/default* contiene las opciones de arranque disponibles. Cada sistema operativo que desee arrancar debe tener una entrada en este fichero.

- *pxelinux.cfg/graphics.conf* contiene las opciones de apariencia del menú de arranque.

5. Por último, vamos a instalar en el servidor varias ISOS de arranque de sistemas operativos, y a hacerlas disponibles por red. Utilizaremos los siguientes directorios:

- */var/lib/tftpboot/iso* contendrá las imágenes ISO de instalación.

- */var/lib/tftpboot/mnt* contendrá los puntos de montaje de la imagen de instalación publicados por NFS.

- */var/lib/tftpboot/ks* contendrá los archivos “[kickstart](#)” publicados por NFS.

(a) Vamos a crear dichos directorios:

```
# mkdir /var/lib/tftpboot/iso
```

```
# mkdir /var/lib/tftpboot/mnt
# mkdir /var/lib/tftpboot/ks
```

(b) Vamos a colocar la imagen con la que queremos arrancar y a crear su punto de montaje:

```
# cd /var/lib/tftpboot/iso
# wget https://github.com/eReuse/workbench/releases/download/10.0b8/workbench-live-10.0b8.iso
# mv workbench-live-10.0b8.iso ereuse.iso
# mkdir /var/lib/tftpboot/mnt/ereuse
```

(c) Ahora, para que los ficheros de dicha ISO (`/var/lib/tftpboot/iso/ereuse`) se vean en el punto de montaje (`/var/lib/tftpboot/mnt/ereuse/`), en lugar de descomprimirlos, vamos a montarlos mediante la opción “loop”. El punto de montaje lo añadiremos a `/etc/fstab` para que esté disponible aunque se reinicie el sistema.

```
# nano /etc/fstab
```

```
...
/var/lib/tftpboot/iso/ereuse.iso /var/lib/tftpboot/mnt/ereuse iso9660 user,ro,loop 0 0
...
```

y a continuación se monta para probar que funciona:

```
# mount -a
# ls -l /var/lib/tftpboot/mnt/ereuse/
```

(d) Tenemos que exportar el punto de montaje y los archivos kickstart:

```
# nano /etc/exports
```

```
/var/lib/tftpboot/ks *(no_root_squash,no_subtree_check,ro)
/var/lib/tftpboot/mnt/ereuse *(no_root_squash,no_subtree_check,ro)
```

y a continuación reinicia el servicio:

```
# systemctl restart nfs-kernel-server
# showmount -e 192.168.100.2
```

(e) Hace falta [crear un archivo kickstart](#) para indicar como arrancar la ISO. Esta es la parte más liosa. La ISO es una Ubuntu modificada. En una Ubuntu podemos crear un archivo kickstart instalando el paquete `system-config-kickstart` y después ejecutando el comando `system-config-kickstart`. Tras responder una serie de cuestiones sobre cómo queremos el arranque, generará un archivo que deberemos copiar a nuestro servidor. Ya existen ejemplos de dichos archivos en el directorio `isolinux` de la ISO de XSR, como por ejemplo `txt.cfg`. Si nos atrevemos a crearlo a mano, el archivo podría ser algo tan sencillo como esto:

```
# nano /var/lib/tftpboot/ks/ereuse-ks.cfg
```

```
install
nfs --server 192.168.100.2 --dir /var/lib/tftpboot/mnt/ereuse
```

(f) Por último, debemos crear la entrada del menú para el arranque en red. En el fichero `/var/lib/tftpboot/pxelinux.cfg/default` ya existen un par de entradas, una para *Elastix* y otra para

Ubuntu, que podemos aprovechar o borrar.

```
# nano /var/lib/tftpboot/pxelinux.cfg/default
```

```
LABEL ERUSE
MENU LABEL ^EREUSE
kernel mnt/ereuse/casper/vmlinuz
append file=mnt/ereuse/preseed/ubuntu.seed initrd=mnt/ereuse/casper/initrd.lz boot=casper netboot=nfs
ip=dhcp nfsroot=192.168.100.2:/var/lib/tftpboot/mnt/ereuse quiet splash
```

(g) Ya podemos probarlo configurando la BIOS de un cliente para que arranque por red.

Práctica con Linux: servidor NextCloud para ficheros en la nube

- <https://www.howtoforge.com/tutorial/how-to-install-nextcloud-on-debian-10/>
- <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/Network_File_System_(protocol))
- http://en.wikipedia.org/wiki/Server_Message_Block
- [http://en.wikipedia.org/wiki/Samba_\(software\)](http://en.wikipedia.org/wiki/Samba_(software))

