Documentación del Servidor FTP

1. Instalación del servidor (vsftpd en Linux)

¿Qué es vsftpd?

Para poder empezar a usar FTP en mi servidor Linux, necesité instalar un programa que actúe como servidor. Elegí `vsftpd` (Very Secure FTP Daemon).

Sistema operativo

He instalado vsftpd en Ubuntu 24.04 LTS.

Proceso de instalación

La instalación fue bastante directa. Primero actualicé los repositorios:

```
sudo apt update
```

Luego instalé vsftpd:

```
sudo apt install vsftpd -y
```

Verificación y habilitación

Una vez instalado, verifiqué que el servicio se iniciará correctamente:

```
sudo systemctl start vsftpd
sudo systemctl status vsftpd
```

El servicio estaba activo y funcionando.

Para que arranque automáticamente al reiniciar el sistema:

```
sudo systemctl enable vsftpd
```

Verificación del puerto

Comprobé que vsftpd estaba escuchando en el puerto 21 (puerto estándar de FTP):

```
sudo ss -tulpn | grep vsftpd
```

El resultado mostró que estaba escuchando en `0.0.0.0:21`, lo que significa que acepta conexiones desde cualquier interfaz de red.

2. Configuración básica

Archivo de configuración

La configuración de vsftpd se realiza principalmente editando el archivo `/etc/vsftpd.conf`.

Antes de modificarlo, hice una copia de seguridad por si acaso:

```
sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.backup
```

Edición del archivo

Abrí el archivo con nano:

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

Configuraciones que apliqué

Directiva	Descripción	Valor configurado
`listen`	Hace que vsftpd escuche en IPv4	`YES`
`listen_ipv6`	Escucha en IPv6	`NO`
`anonymous_enable`	Permite o deniega el acceso anónimo	`NO` (más seguro)
`local_enable`	Permite a los usuarios locales iniciar sesión	`YES`
`write_enable`	Permite comandos de escritura (subir archivos, crear directorios)	`YES`
`chroot_local_user`	Encierra a los usuarios locales en su directorio home (MUY IMPORTANTE para seguridad)	`YES`
`allow_writeable_chroot`	Permite chroot aunque el directorio sea escribible	`YES`

Configuración de modo pasivo:

```
pasv_enable=YES
pasv_min_port=40000
pasv_max_port=40100
```

Límites de conexión:

```
max_clients=10
max_per_ip=2
```

Mensaje de bienvenida:

```
ftpd_banner=Bienvenido al servidor FTP de Aroa
```

Permisos de archivos:

```
file_open_mode=0666
local_umask=022
```

Los archivos subidos tienen permisos 644 `(rw-r--r--)` por defecto.

Aplicar cambios

Después de cada modificación, reinicié el servicio:

```
sudo systemctl restart vsftpd
```

3. Usuarios y permisos

Creación de grupo FTP

Los usuarios que acceden al servidor FTP son usuarios del sistema Linux. Primero creé un grupo específico para ellos:

```
sudo groupadd ftpusers
```

Creación de usuarios

Creé dos usuarios de ejemplo para las pruebas:

Usuario 1:

```
sudo useradd -m -g ftpusers -s /bin/bash aroa1
sudo passwd aroa1
```

Usuario 2:

```
sudo useradd -m -g ftpusers -s /bin/bash invitado1
sudo passwd invitado1
```

Estructura de directorios (chroot)

Como tengo la directiva `chroot_local_user=YES`, el usuario solo puede moverse dentro de su directorio `/home/aroa1`.

Para que vsftpd funcione correctamente con chroot, el directorio home no puede tener permisos de escritura. Por eso creé esta estructura:

→ Directorio raíz (solo lectura):

```
sudo mkdir -p /home/aroa1/ftp
sudo chown nobody:nogroup /home/aroa1/ftp
sudo chmod a-w /home/aroa1/ftp
```

Este directorio es de solo lectura por requisitos de seguridad de vsftpd.

→ Subdirectorio con permisos de escritura:

```
sudo mkdir /home/aroa1/ftp/files
sudo chown aroa1:ftpusers /home/aroa1/ftp/files
sudo chmod 775 /home/aroa1/ftp/files
```

En este subdirectorio el usuario SÍ puede subir, modificar y borrar archivos.

Repetí el mismo proceso para `invitado1`.

Permisos explicados

Los permisos `775 (rwxrwxr-x)` significan:

Quién	Permiso	Número	¿Qué puede hacer?
Propietario (aroa1)	`rwx`	7	Leer, escribir y ejecutar
Grupo (ftpusers)	`rwx`	7	Leer, escribir y ejecutar
Otros	`r-x`	5	Solo leer y ejecutar

Lista de usuarios permitidos

Creé una lista de usuarios que pueden usar FTP:

```
sudo nano /etc/vsftpd.user_list
```

Contenido:

```
aroa1
invitado1
```

Y en el archivo de configuración añadí:

```
userlist_enable=YES
userlist_file=/etc/vsftpd.user_list
userlist_deny=NO
```

Solo los usuarios en esta lista pueden conectarse por FTP.

Acceso anónimo (configurado temporalmente para pruebas)

Para probar el acceso anónimo, lo configuré temporalmente:

Directorio para anónimos:

```
sudo mkdir -p /srv/ftp/anonymous
sudo chown nobody:nogroup /srv/ftp/anonymous
sudo chmod 555 /srv/ftp/anonymous
```

Configuración en vsftpd.conf:

```
anonymous_enable=YES
anon_root=/srv/ftp/anonymous
no_anon_password=YES
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
```

Los usuarios anónimos pueden ver y descargar archivos, pero no pueden subir, crear carpetas ni modificar nada.

4. Seguridad (FTPS)

FTP por sí solo no es seguro, ya que envía contraseñas y datos en texto plano. Para añadir seguridad, usé FTPS (FTP Secure), que utiliza SSL/TLS para cifrar la conexión.

Generación del certificado SSL/TLS

Generé un certificado autofirmado válido por 365 días:

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

Durante la generación me pidió estos datos:

- `Country Name: ES`
- `State or Province Name: Castilla-La Mancha`
- `Locality Name: Alcázar de San Juan`
- `Organization Name: IES Juan Bosco`
- `Organizational Unit Name: DAW`
- `Common Name: localhost`
- `Email Address: aroa@localhost`

Configuración de vsftpd para usar SSL/TLS

Añadí estas líneas en `/etc/vsftpd.conf`:

`ssl_enable=YES` Activa SSL/TLS
`rsa_cert_file=/etc/ssl/certs/vsftpd.crt`
`rsa_private_key_file=/etc/ssl/private/vsftpd.key`
`allow_anon_ssl=NO`
`force_local_data_ssl=YES` Obliga a usar cifrado para la transferencia de datos
`force_local_logins_ssl=YES` Obliga a usar cifrado para el login
`ssl_tlsv1=YES` Usa TLS 1.x (versiones seguras)
`ssl_sslv2=NO` Deshabilita versiones antiguas e inseguras
`ssl_sslv3=NO` Solo permite algoritmos de cifrado fuertes
`ssl_ciphers=HIGH`
`require_ssl_reuse=NO`

Verificación del cifrado obligatorio

Para comprobar que el cifrado era obligatorio, intenté conectarme con el cliente básico `ftp` (que no soporta SSL):

```
ftp localhost
```

El servidor rechazó la conexión con el mensaje:

```
530 Non-anonymous sessions must use encryption.
Login failed.
```

Esto confirma que es obligatorio usar cifrado.

Conexión con FTPS

Para conectarme con cifrado, usé FileZilla configurado con:

- Cifrado: "Requerir FTP explícito sobre TLS"

La conexión se estableció correctamente mostrando:

- "Inicializando TLS..."
- "Conexión TLS establecida"

Todo el tráfico (credenciales y archivos) viaja cifrado.

5. Modos activo y pasivo

El FTP tiene dos modos de conexión para la transferencia de datos:

Diferencias entre modos

Modo	¿Quién inicia la conexión de datos?	Descripción
Modo Activo	El servidor se conecta al cliente	El cliente envía al servidor la IP y puerto al que el servidor debe conectarse. Esto suele ser BLOQUEADO por los firewalls del cliente.
Modo Pasivo	El cliente se conecta al servidor	El cliente le pide al servidor un puerto. El servidor le indica el puerto y el cliente inicia la conexión. Es el modo que mejor funciona con NAT y firewalls.

Recomendación: El modo pasivo funciona mejor casi siempre porque es más fácil atravesar firewalls y routers modernos.

Configuración del modo pasivo en vsftpd

En `/etc/vsftpd.conf` configuré:

```
pasv_enable=YES
pasv_min_port=40000
pasv_max_port=40100
```

El servidor usa puertos entre 40000 y 40100 para las conexiones de datos en modo pasivo.

Configuración del firewall

Instalé y configuré ufw para abrir los puertos necesarios:

```
sudo apt install ufw -y
sudo ufw enable
sudo ufw allow 21/tcp
sudo ufw allow 40000:40100/tcp
```

Explicación:

- Puerto 21: Para el canal de control
- Rango 40000-40100: Para el canal de datos en modo pasivo

Pruebas realizadas

Con modo pasivo:

Me conecté con FileZilla. El servidor respondió:

```
227 Entering Passive Mode
```

Funcionó perfectamente usando un puerto del rango configurado.

Con modo activo:

Dentro del cliente FTP ejecuté el comando `passive` para desactivar el modo pasivo. En localhost funcionó, pero en un escenario real con firewall fallaría porque el firewall del cliente bloquearía la conexión entrante del servidor.

¿Por qué el modo pasivo es mejor?

- El cliente inicia las dos conexiones (salientes desde su punto de vista)
- Los firewalls permiten conexiones salientes por defecto
- Compatible con NAT sin configuración adicional en el cliente
- No requiere abrir puertos en el firewall del cliente

6. Clientes utilizados

He probado diferentes tipos de clientes FTP para conectarme al servidor

Clientes de línea de comandos

ftp (cliente básico):

```
ftp localhost
```

- Muy simple, viene incluido en el sistema
- No soporta FTPS
- Útil para pruebas rápidas
- Comandos básicos: ls, cd, get, put, quit

lftp (cliente avanzado):

```
lftp -u aroa1 localhost
```

- Más potente que ftp
- Soporta FTPS
- Permite hacer scripts
- Comandos adicionales como mirror

Para que funcione con FTPS autofirmado:

```
lftp -u aroa1 -e "set ssl:verify-certificate no; ls; quit" localhost
```

curl:

```
curl -u aroa1:contraseña ftp://localhost/files/
curl -u aroa1:contraseña -T archivo.txt ftp://localhost/files/
```

- Útil para automatización
- Soporta FTPS
- Ideal para scripts

Cliente gráfico: FileZilla

FileZilla Client es el que más he usado. Es multiplataforma, gratuito y muy fácil de usar.
Configuración de conexión guardada:

Campo FileZilla	Información
Servidor	localhost
Nombre de usuario	aroa1
Contraseña	(contraseña de aroa1)
Puerto	21 (estándar para FTP/FTPS)
Cifrado	Requerir FTP explícito sobre TLS

Funcionalidades que usé:

- Vista de dos paneles (local y remoto)
- Arrastrar y soltar archivos
- Transferencias simultáneas
- Gestión de conexiones guardadas
- Soporte completo para FTPS

Conexión con certificado autofirmado:

Al conectar por primera vez, FileZilla mostró un aviso de "Certificado desconocido". Comprobé que los datos coincidían (localhost, IES Juan Bosco, DAW, ES) y acepté el certificado marcando "Confiar siempre en este certificado".

Navegador web y gestor de archivos

Navegadores modernos:

Los navegadores (Firefox, Chrome) ya no soportan FTP. Al intentar `ftp://localhost`, Firefox muestra un mensaje pidiendo elegir una aplicación externa.

Gestor de archivos Nautilus:

Como alternativa, el gestor de archivos del sistema tiene soporte FTP básico. Presioné Ctrl+L y escribí:

```
ftp://localhost
```

Me pidió usuario y contraseña, y pude navegar y descargar archivos, pero con funcionalidad muy limitada comparado con FileZilla.

Comparación de clientes

Cliente	Ventajas	Desventajas
ftp	Simple, siempre disponible	No soporta FTPS, muy básico
lftp	Potente, scripting, FTPS	Línea de comandos
curl	Automatización, FTPS	Solo operaciones simples
FileZilla	Interfaz completa, FTPS, fácil	Requiere instalación
Navegador	No requiere nada extra	Ya no funciona
Nautilus	Integrado en el sistema	Muy limitado

7. Integración web

Objetivo

Un uso muy habitual del FTP es para subir contenido a una web. Por eso vinculé el servidor FTP con Apache para poder subir archivos por FTP y que sean accesibles vía HTTP.

Verificación de Apache

Comprobé que Apache estaba instalado y funcionando:

```
sudo systemctl status apache2
```

El DocumentRoot de Apache está en `/var/www/html.`

Usuario para publicación web

Creé un usuario específico para subir contenido web:

```
sudo useradd -m -g www-data -s /bin/bash webmaster
sudo passwd webmaster
```

Lo añadí al grupo `www-data` (el grupo de Apache) para que los archivos subidos sean accesibles por el servidor web.

Directorio vinculado

Creé un directorio dentro del DocumentRoot de Apache:

```
sudo mkdir -p /var/www/html/ftp
sudo chown www-data:www-data /var/www/html/ftp
sudo chmod 755 /var/www/html/ftp
```

Permisos 755 permiten que Apache pueda leer y servir los archivos.

Configuración de vsftpd para webmaster

Creé una configuración específica para que el directorio raíz de webmaster sea el directorio de Apache:

```
sudo mkdir -p /etc/vsftpd/user_conf
sudo nano /etc/vsftpd/user_conf/webmaster
```

Contenido:

```
local_root=/var/www/html/ftp
```

Esto hace que cuando webmaster se conecte por FTP, su directorio raíz sea `/var/www/html/ftp.`

También añadí webmaster a la lista de usuarios permitidos:

```
sudo nano /etc/vsftpd.user_list
```

Añadí la línea:

```
webmaster
```

Y en `/etc/vsftpd.conf`:

```
user_config_dir=/etc/vsftpd/user_conf
```

Flujo de publicación web

1. Creación del contenido → Creo un archivo HTML en mi ordenador local
2. Conexión FTP → Me conecto con FileZilla como usuario webmaster
3. Subida del archivo → Transfiero el archivo HTML al servidor por FTP
4. Almacenamiento → El archivo se guarda en `/var/www/html/ftp/`
5. Servido por Apache → Apache sirve el archivo automáticamente
6. Acceso HTTP → Accedo desde el navegador a http://localhost/ftp/archivo.html

Prueba realizada

Creé un archivo HTML de prueba con contenido formateado y lo subí por FTP. Luego accedí desde el navegador a:

```
http://localhost/ftp/prueba_web.html
```

La página se cargó correctamente mostrando el contenido HTML, demostrando que la integración FTP-HTTP funciona perfectamente.

Ventajas de este método

- Fácil de usar para desarrolladores que no tienen conocimientos de SSH
- Interfaz gráfica (FileZilla) para gestión de archivos
- Separación de responsabilidades (FTP para subida, HTTP para servir)
- Los archivos quedan disponibles inmediatamente tras subirlos

Precauciones con permisos

Tuve que ajustar los permisos correctamente porque el usuario FTP necesita permisos de escritura, pero los permisos del servidor web también deben estar correctos. Un error de permisos puede causar problemas de seguridad o de funcionamiento de la web.

8. Recomendaciones de administración

La administración del FTP es clave para mantener la seguridad y el rendimiento.

Seguridad y contraseñas

- Usar contraseñas fuertes
- Obligatorio usar FTPS (nunca FTP sin cifrar en producción)
- Deshabilitar acceso anónimo en producción
- Mantener chroot activado
- Nunca usar root para conexiones FTP (principio del mínimo privilegio)

Límites de conexión

Usar directivas como `max_clients` y `max_per_ip` en `vsftpd.conf` para limitar el número de conexiones y evitar ataques de denegación de servicio:

```
max_clients=50
max_per_ip=3
idle_session_timeout=600
data_connection_timeout=300
```

Esto protege contra sobrecarga y ataques DoS.

Auditoría y logs

Habilitar logs detallados:

```
xferlog_enable=YES
xferlog_file=/var/log/vsftpd.log
log_ftp_protocol=YES
vsftpd_log_file=/var/log/vsftpd-detailed.log
```

Revisión regular:

Revisar regularmente los logs de vsftpd (suelen estar en `/var/log/vsftpd.log`) para detectar intentos de acceso no autorizados.

Rotación de logs:

Configurar rotación automática:

- Rotar diariamente
- Mantener 30 días de histórico
- Comprimir logs antiguos

Copias de seguridad

¿Qué hacer backup?

- Archivos de configuración (`/etc/vsftpd.conf`)
- Configuraciones de usuarios (`/etc/vsftpd/user_conf/`)
- Datos de usuarios
- Certificados SSL
- Logs

Frecuencia:

- Backup completo semanal
- Backups incrementales diarios
- Retención de 30 días

Regla 3-2-1:

- 3 copias de los datos
- En 2 tipos de medios
- 1 copia fuera del servidor

Verificación:

Probar restauración mensualmente. Un backup que no funciona es como no tenerlo.

Firewall

Asegurarse de que el firewall permite los puertos necesarios:

```
sudo ufw allow 22/tcp      # SSH para administración
sudo ufw allow 21/tcp      # FTP control
sudo ufw allow 40000:40100/tcp  # FTP datos en modo pasivo
sudo ufw enable
```

Protección contra ataques de fuerza bruta

Instalar y configurar fail2ban para bloquear automáticamente IPs con múltiples intentos fallidos:

```
sudo apt install fail2ban -y
```

Configuración básica:

- 3 fallos de login en 10 minutos → Bloqueo de 1 hora

NAT y conectividad externa

Si el servidor está detrás de un router (NAT):

1. Configurar port forwarding en el router:

- Puerto 21 → IP del servidor FTP
- Rango 40000-40100 → IP del servidor FTP

2. Indicar la IP pública en vsftpd:

```
pasv_address=IP_PUBLICA_DEL_SERVIDOR
```

Esto ayuda a que el cliente sepa a dónde conectarse para las transferencias de datos.

Actualizaciones

Mantener tanto el servidor vsftpd como el cliente FileZilla actualizados para tener los últimos parches de seguridad:

```
sudo apt update
sudo apt upgrade vsftpd
```