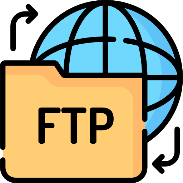
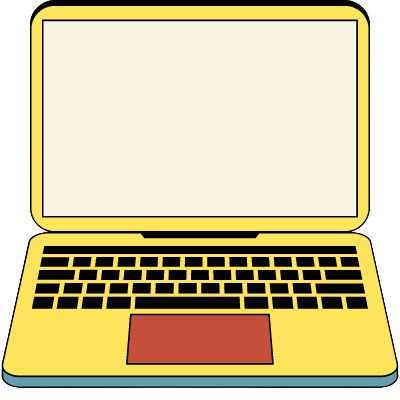
**Christian Millán Soria**

Instalación de Servidor FTP





- Curso: 2023-2024

- Clase: 2º DAW Tarde

- Profesor: David Hormigo Ramírez

- Módulo: Despliegue de Aplicaciones Web

- Fecha: 31/10/2023

**1. Instalación del servidor** 3

**2. Usuario** 6

**2.1. Enjaulamiento** 6

**2.2. Puertos y método de conexión** 7

**2.2.1. Método de Conexión Activa** 7

**2.2.2. Método de Conexión Pasiva** 7

**2.2.3. Puerto 21 (Control)** 8

**2.2.4. Puerto 20 (Datos, en modo activo)** 8

**2.3. Masquerade Address** 8

**2.4. Limit** 9

**3. Reglas del firewall** 9

**4. Cliente local** 11

**4.1. Configuración del sitio** 12

**4.1.1. Protocolo** 12

**4.1.2. Host** 12

**4.1.3. Cifrado** 13

**4.1.4. Método de acceso** 13

**4.1.5. Usuario** 13

**4.1.6. Transferencia** 13

**4.2. Comprobación de los métodos de conexión** 15

**4.2.1. Activo** 15

**4.2.2. Pasivo** 15

**4.3. Comprobación de validez de usuario** 16

**5. Configuraciones adicionales** 18

**5.1. Nombre del servidor** 18

**5.2. Transfer timeout** 18

**5.3. Número máximo de instancias** 18

**5.4. Registro de actividades** 19

**5.5. Permitir sobrescribir** 19

**5.6. Denegación de filtro** 19

**5.7. Usuarios bloqueados** 19

**5.8. Lista de usuarios permitidos** 20

En esta práctica se explica el proceso de instalación de un [**servidor FTP🔗**](https://en.wikipedia.org/wiki/File_Transfer_Protocol#:~:text=The%20File%20Transfer%20Protocol%20(FTP)%20is%20a%20standard%20communication%20protocol%20used%20for%20the%20transfer%20of%20computer%20files%20from%20a%20server%20to%20a%20client%20on%20a%20computer%20network.%20FTP%20is%20built%20on%20a%20client%E2%80%93server%20model%20architecture%20using%20separate%20control%20and%20data%20connections%20between%20the%20client%20and%20the%20server.%5B1%5D).

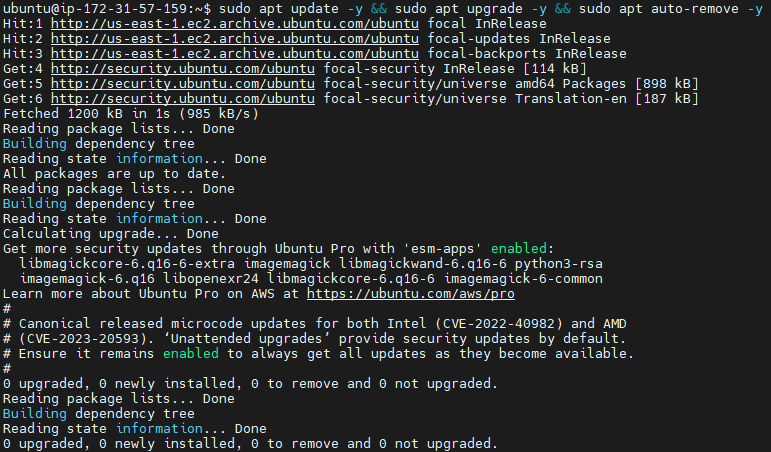
FTP es un protocolo utilizado para transferir todo tipo de archivos entre equipos conectados a una red.

Es un protocolo en desuso, debido a que ciertos aspectos de su seguridad y eficiencia han quedado obsoletos en comparación con protocolos más modernos. Sin embargo, aún se utiliza en algunos entornos y para ciertas aplicaciones específicas.

A continuación, se explica el proceso de instalación de un servidor FTP en un sistema Linux (instancia en AWS).

**1. Instalación del servidor**

Seguimos con las buenas prácticas para cada una de las tareas que vamos realizando: actualizamos el servidor de nuevo con el comando “**sudo apt update -y && sudo apt upgrade -y && sudo apt auto-remove -y**”.

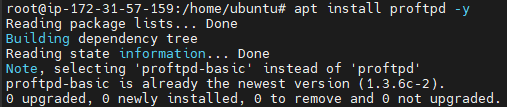


Debido a que hay determinados paquetes que se instalan con permisos u owners diferentes dependiendo del método de instalación utilizado, para la instalación y configuración de este servidor utilizaremos el usuario [“**root**”🔗](https://www.ssh.com/academy/pam/root-user-account#:~:text=Root%20is%20the%20superuser%20account,account%2C%20regardless%20of%20the%20name.) para llevar a cabo todos estos pasos y que no ocurra ningún conflicto.

Para iniciar sesión como root usaremos el comando [“**sudo su**”🔗](https://explainshell.com/explain?cmd=sudo+su), el cual nos permitirá ejecutar las siguientes acciones necesarias en la práctica sin la necesidad de utilizar “sudo” para cada comando de importancia superior a la normal.



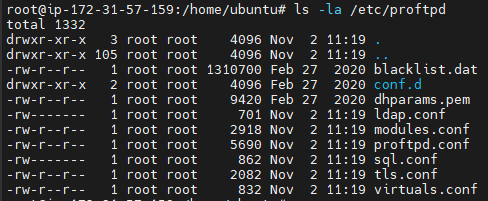
Acto seguido, ejecutamos el comando de instalación de nuestro nuevo servidor, el cual es “**apt install proftpd -y**”.



Podemos comprobar que el servidor está corriendo sin problemas con el comando “**systemctl status proftpd**”.



También podemos verificar que el árbol de directorios del servidor se ha creado correctamente en la ruta por defecto para ellos: **ls -la /etc/proftpd**.



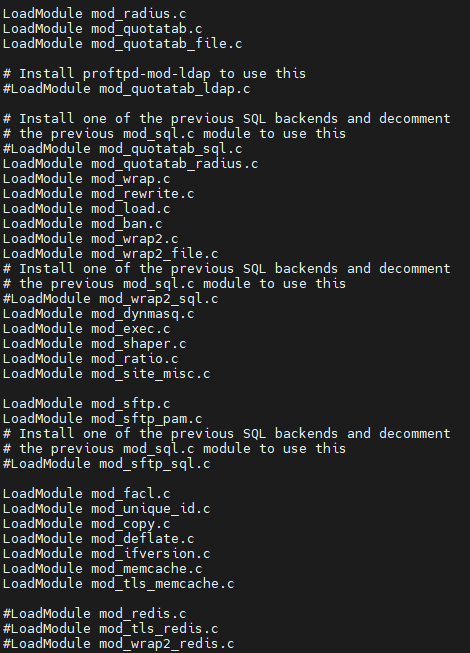
Vemos como el dueño de todos los archivos y directorios es **root**.

Hemos instalado un [**servidor** **ProFTPD🔗**](http://www.proftpd.org/docs/howto/Limit.html), el cuál es una de las variadas opciones disponibles, entre las que destaca por ser una solución confiable y ampliamente utilizada para la transferencia de archivos a través de FTP.

Este servidor es [**modular**🔗](http://www.proftpd.org/docs/modules/index.html) (igual que Apache), lo que quiere decir que su funcionalidad puede ser alterada/modificada mediante la adición o eliminación de módulos según las necesidades del mismo.

Para ver una lista de los módulos habilitados en el servidor, al contrario que en Apache (donde contábamos con el comando **a2** para habilitar o deshabilitar módulos), debemos editar a mano el archivo dedicado para dichos elementos, donde se descomentarán las líneas dedicadas a los módulos que queremos habilitar, y se comentarán las líneas dedicadas a los módulos que no queremos tener activos.

Con el comando “**cat /etc/proftpd/modules.conf**” vemos el contenido de dicho archivo.



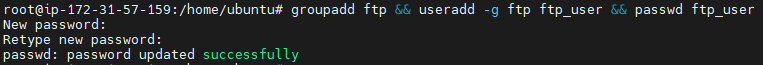
**2. Usuario**

La segunda parte de la práctica se basa en la creación del usuario “**ftp\_user**” y añadirlo al grupo “**ftp**”. Es probable que este grupo no exista, por lo que utilizaremos el siguiente comando para crearlo y añadir al usuario al grupo:

**groupadd ftp && useradd -g ftp ftp\_user && passwd ftp\_user**

[**Este comando🔗**](https://explainshell.com/explain?cmd=groupadd+ftp+%26%26+useradd+-g+ftp+ftp_user+%26%26+passwd+ftp_user)crea el nuevo grupo llamado “ftp” y añade el nuevo usuario al mismo.

Al final se pide un cambio de contraseña para el nuevo usuario, la cual será necesaria para más adelante.



Con el comando “**groups ftp\_user**” podemos comprobar que el usuario se ha añadido al grupo correcto.



Como podemos comprobar con el comando “**ls -la /home/ftp\_user**”, este directorio no existe, pero lo necesitamos para el correcto funcionamiento de las conexiones y de las restricciones que aplicaremos más adelante.

Para solucionar esto, ejecutaremos el siguiente comando: “**sudo mkdir /home/ftp\_user**”, el cual se encarga de crear dicho directorio, seguido del comando “**sudo chown ftp\_user:ftp /home/ftp\_user**” para otorgar a este usuario los permisos suficientes para poder trabajar con su directorio personal.

**2.1. Enjaulamiento**

En prácticas anteriores hemos utilizado WinSCP, un cliente SCP que proporciona un entorno gráfico mediante el cual se pueden transferir archivos de una máquina local a una remota utilizando el protocolo SCP.

El entorno gráfico mencionado permitía una completa visualización del árbol de directorios de la máquina remota, algo que debemos evitar en esta práctica.

Mediante el método conocido como “**enjaulamiento**”, haremos que el usuario con el que nos conectemos a la máquina remota mediante la utilización del protocolo FTP solo pueda navegar en su árbol de directorios personal, es decir, su home.

Así como el archivo de configuración global de Apache era el “apache2.conf”, en ProFTP el archivo es el llamado “**proftpd.conf**”:

**sudo nano /etc/proftpd/proftpd.conf**

\* A veces es necesario ejecutar sudo incluso siendo root.

Una vez dentro del editor para este archivo, nos dirigimos a la línea 36, donde se encuentra la directiva [“**DefaultRoot**”🔗](http://www.proftpd.org/docs/howto/Chroot.html), la cual está comentada (**con un “#” al comienzo de la línea**). Descomentamos la línea para aplicar la directiva eliminando el “#” del comienzo de la misma.

Además, al final de dicha línea agregaremos un espacio y el nombre del usuario que hemos creado.



**2.2. Puertos y método de conexión**

En FTP existen dos métodos posibles para poder llevar a cabo una comunicación con el servidor: [**Activo** y **Pasivo**🔗](https://www.geeksforgeeks.org/difference-between-active-and-passive-ftp/).

**2.2.1. Método de Conexión Activa**

En el método activo de FTP, el servidor utiliza el puerto 21 para la comunicación de control. Cuando se necesita transferir datos, el servidor inicia una conexión de datos saliente desde su puerto 20 hacia un puerto efímero (generalmente en el rango de puertos no privilegiados) en el cliente.

El cliente notifica al servidor sobre su dirección IP y el puerto en el que está escuchando a través del comando [**PORT**🔗](https://www.techtarget.com/searchnetworking/tip/Understanding-the-FTP-PORT-command#:~:text=In%20summary%2C%20the%20PORT%20command,a%20connection%20to%20that%20port.). Esto permite al servidor enviar los datos al cliente a través de la conexión de datos establecida.

**2.2.2. Método de Conexión Pasiva**

En el método pasivo de FTP, el servidor aún utiliza el puerto 21 para la comunicación de control. Sin embargo, cuando se necesita transferir datos, el cliente envía un comando [**PASV**🔗](https://www.serv-u.com/resources/tutorial/pasv-response-epsv-port-pbsz-rein-ftp-command#:~:text=Other%20commands-,PASV%20FTP%20command,-This%20command%20tells) al servidor. El servidor responde con una dirección IP y un puerto efímero en el que el cliente debe conectarse para la transferencia de datos.

El cliente establece una conexión de datos entrante con el servidor en el puerto especificado por el servidor.

**2.2.3. Puerto 21 (Control)**

El puerto 21 se utiliza para la comunicación de control en FTP. Aquí, el cliente y el servidor intercambian comandos y respuestas para la gestión de la sesión FTP, como iniciar sesión, cambiar directorios y solicitar operaciones.

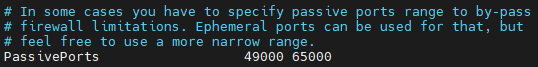
**2.2.4. Puerto 20 (Datos, en modo activo)**

En una conexión FTP activa, el puerto 20 se utiliza para la transferencia de datos. El servidor inicia la conexión de datos hacia un puerto efímero en el cliente para enviar los archivos.

Si bajamos un poco en el archivo de configuración nos encontramos con la directiva “**Port**”, la cual establece el puerto de control para las conexiones que vayamos a realizar.

Lo dejamos en el puerto 21.

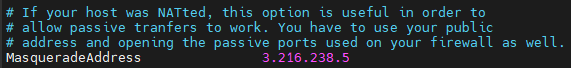
Si seguimos bajando un poco más vemos la directiva “**PassivePorts**”, donde deberemos establecer un rango de puertos que el servidor FTP utilizará para poder llevar a cabo las acciones necesarias para ofrecer el servicio.



**2.3. Masquerade Address**

Para ser capaces de utilizar el método de conexión pasiva debemos establecer la dirección IP pública de nuestro servidor para que pueda ser localizable.

En la línea número 53 del archivo de configuración, descomentamos la directiva “**MasqueradeAddress**” y especificamos la dirección IP pública de nuestro servidor.



Adicionalmente, en la línea número 70 del archivo, cambiaremos el usuario que aparece por el nombre del usuario que hemos creado y el grupo de la línea 71 por el nuevo de esta práctica.



**2.4. Limit**

La documentación de la práctica indica que el **único usuario que debe tener acceso** al servidor es el nuevo usuario creado, “**ftp\_user**”.

Esto se consigue con la directiva [“**Limit**”**🔗**](http://www.proftpd.org/docs/howto/Limit.html).

Nos dirigimos al final del archivo para no tocar ningún ajuste que no debemos. Hacemos clic en la tecla **Enter** dos veces para separarnos del resto del contenido de configuración.

Aquí deberemos ingresar el siguiente contenido:

**<Limit LOGIN>**

**AllowUser ftp\_user**

**DenyAll**

**</Limit>**

Esto establece que el único usuario que tiene permiso para realizar conexiones con el servidor FTP es el usuario que hemos creado en esta misma práctica.

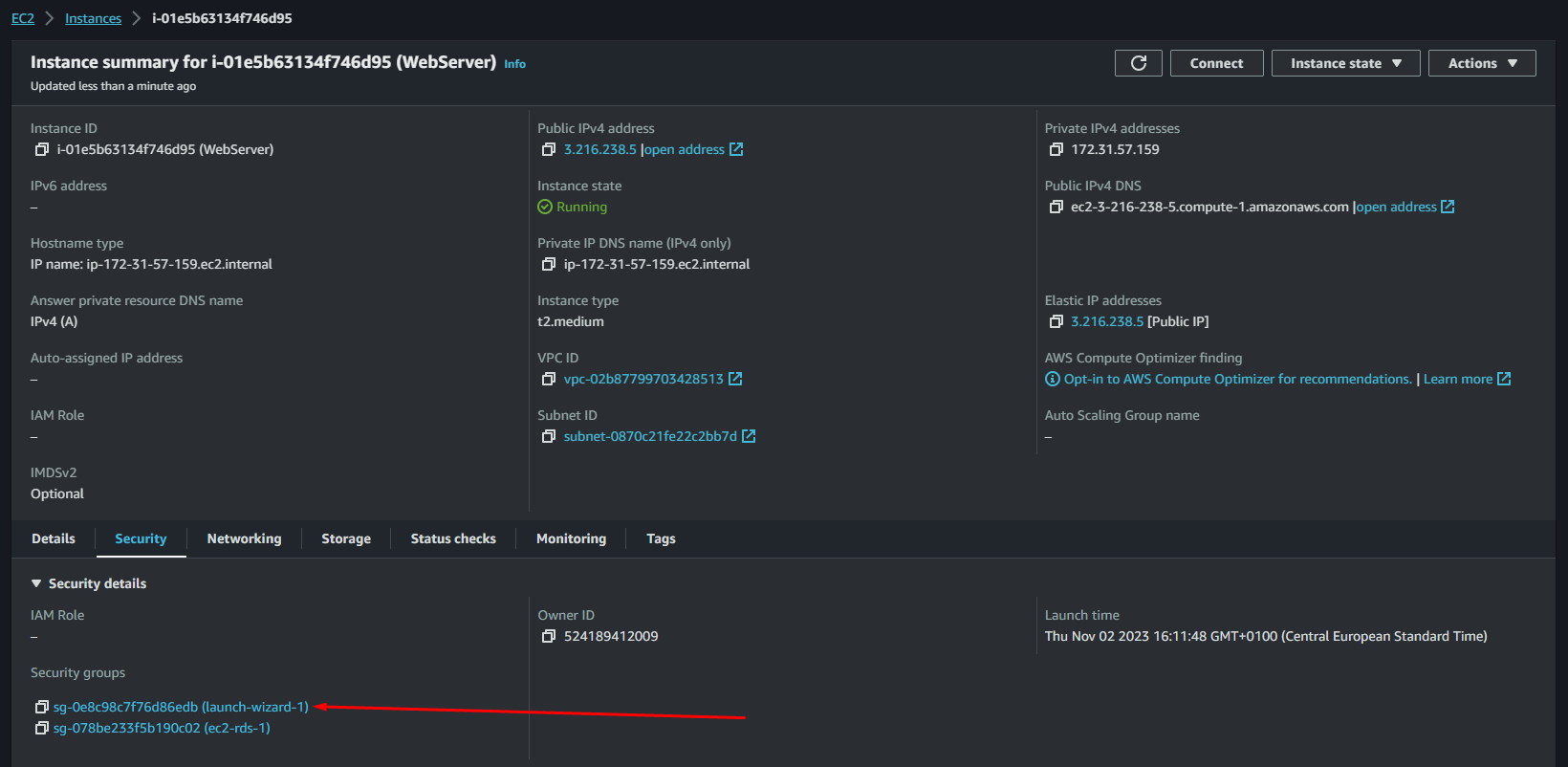
Guardamos los cambios y salimos del archivo. Reiniciamos el servidor con el comando “**sudo systemctl restart proftpd**” para aplicarlos.

**3. Reglas del firewall**

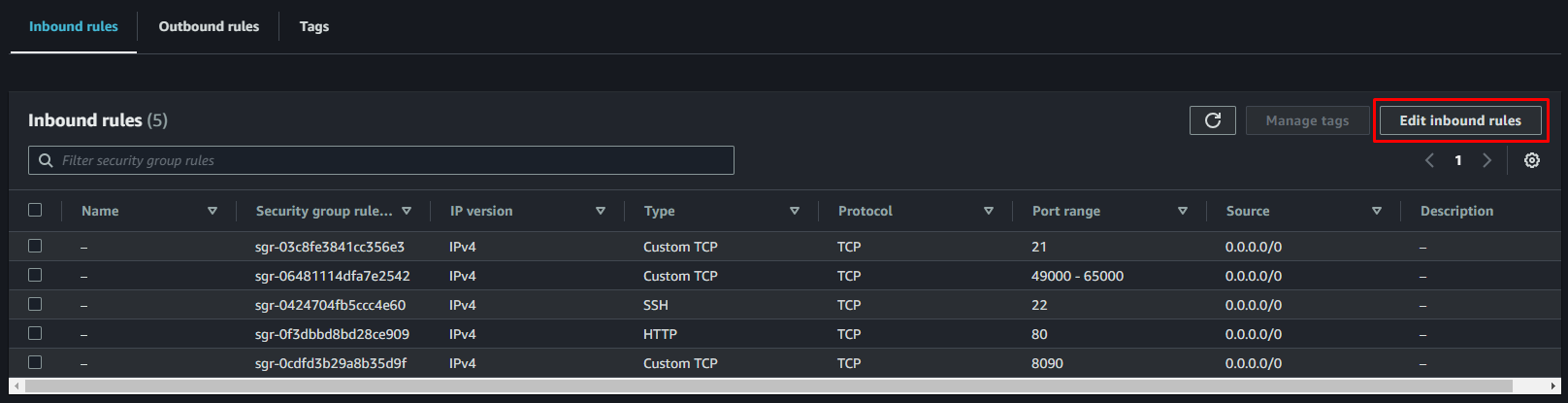
Dado que vamos a utilizar varios puertos de nuestra máquina de AWS, vamos a necesitar habilitar unas reglas de entrada para que la conexión se pueda realizar mediante el uso de unos puertos determinados, los cuales ya hemos establecido en el archivo de configuración.

Nos dirigimos al resumen de nuestra instancia de AWS.

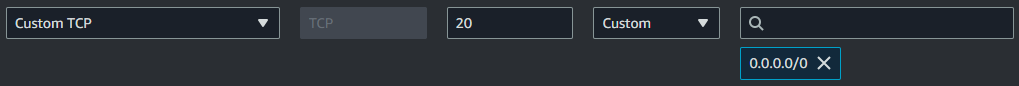
Abrimos la pestaña de seguridad y nos encontramos con que aparecen dos grupos de seguridad listados en nuestra instancia. Esto se debe a que cuando creamos la RDS en la [**anterior práctica**🔗](https://github.com/cmilsor245/daweb/blob/main/04-finalizar_pila_lamp/Finalizar%20instalaci%C3%B3n%20-%20Pila%20LAMP.pdf) este grupo de seguridad se enlazó a nuestra instancia, pero el que debemos seleccionar es el “**launch-wizard**” (o el grupo de seguridad con el que cuente nuestra instancia).

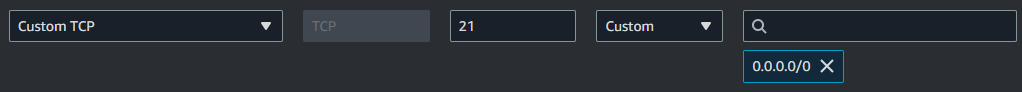


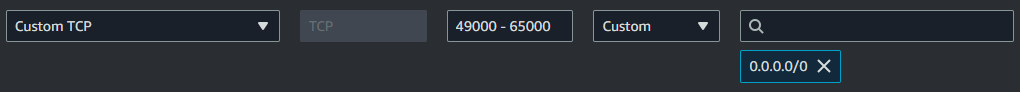
Una vez aquí, hacemos clic en el botón “**Edit inbound rules**” para agregar las nuestras.



Dentro del menú de configuración de las reglas de entrada de nuestra máquina, agregaremos **tres nuevas reglas**, una para el puerto 20, otra para el 21 y otra para el rango de puertos del método de conexión pasivo.







Guardamos los cambios.



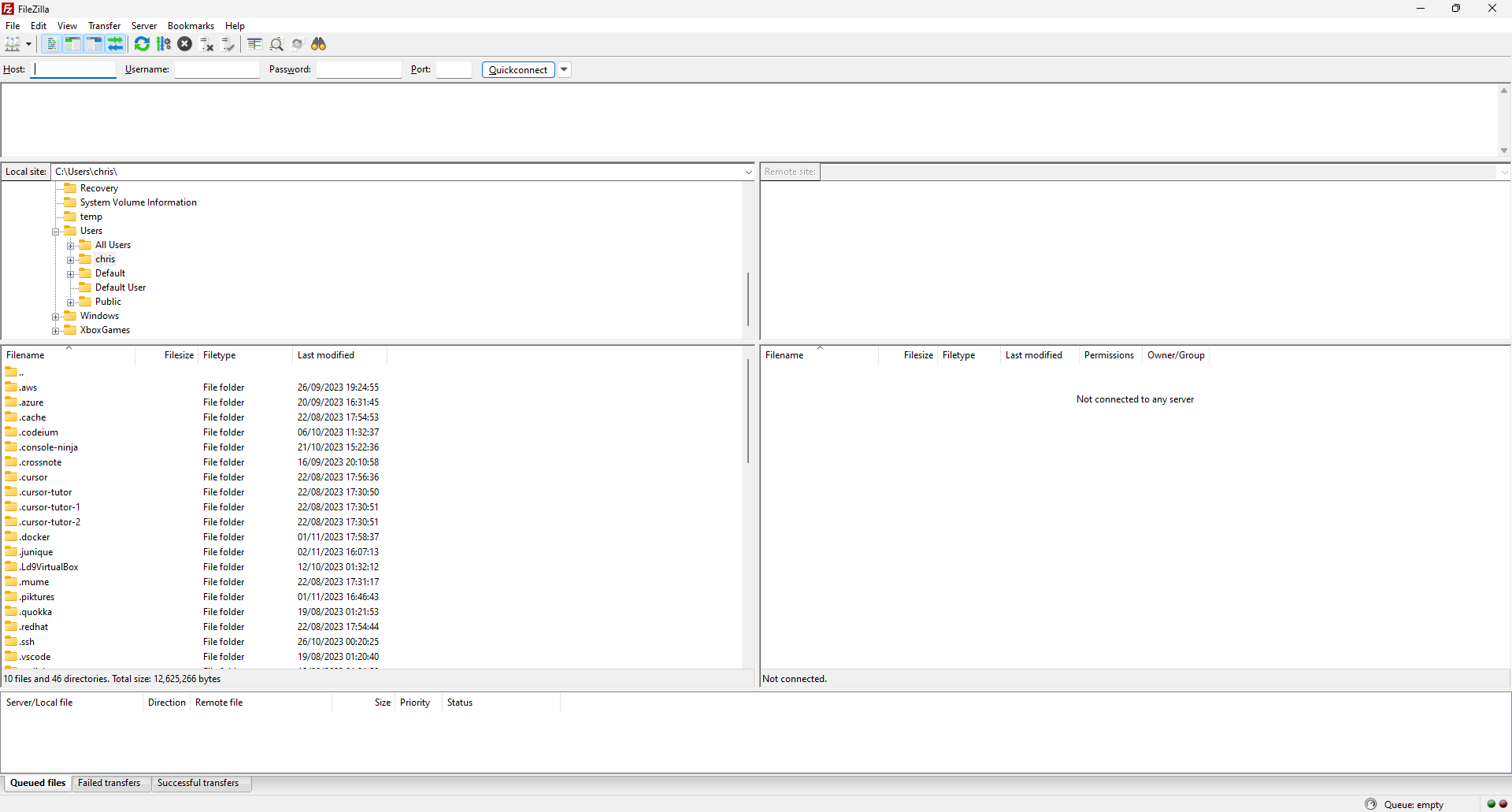
**4. Cliente local**

El siguiente paso es instalar un cliente con el que realizar la conexión hacia el servidor FTP desde nuestra máquina local.

Para esta práctica utilizaremos el llamado [“**FileZilla**”🔗](https://filezilla-project.org/), un cliente bastante utilizado para este tipo de conexiones.

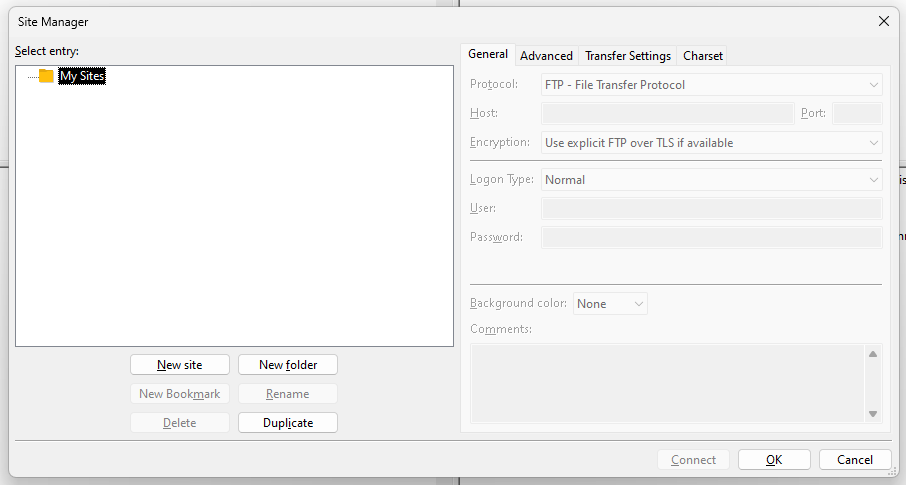
\* Se presupone que FileZilla ya está instalado.

Al abrir el programa se muestra una interfaz como la siguiente:



Es una interfaz muy similar a la de WinSCP, como se ha mencionado anteriormente.

Hacemos clic en el botón “**File**”, en la esquina superior izquierda, y elegimos la opción “**Site Manager…**” en el desplegable que se nos muestra.



En este cuadro podremos crear la nueva conexión.

Hacemos clic en el botón “**New site**” de abajo a la izquierda.

Establecemos un nombre para el nuevo sitio.



Pasamos al menú lateral derecho, donde configuraremos el sitio.

**4.1. Configuración del sitio**

Pasaremos por cada uno de los apartados de la configuración que precisan de un cambio de su forma predeterminada.

**4.1.1. Protocolo**

Seleccionamos el protocolo que vamos a utilizar para conectarnos al servidor, el cual es “**FTP - File Transfer Protocol**”.



**4.1.2. Host**

Ingresamos el host de nuestro servidor, el cual corresponde a la dirección IP pública de nuestra máquina/instancia en AWS.

Dejamos el puerto para la conexión vacío, así el programa utilizará el predeterminado para la misma.



**4.1.3. Cifrado**

Debido a que nuestro servidor no cuenta con [**TLS🔗**](https://en.wikipedia.org/wiki/Transport_Layer_Security), no podrá establecer la conexión utilizando este método, pero dejaremos la opción de encriptación en “**Use explicit FTP over TLS if available**” seleccionada igualmente, ya que se realizará una comprobación y si el servidor no cuenta con TLS se comenzará una conexión sin este método.



**4.1.4. Método de acceso**

Elegimos que se nos solicite una contraseña a la hora de realizar la conexión.



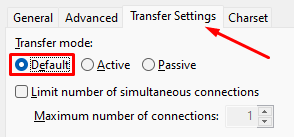
**4.1.5. Usuario**

Entraremos con el usuario “**ftp\_user**”, creado anteriormente en esta práctica.

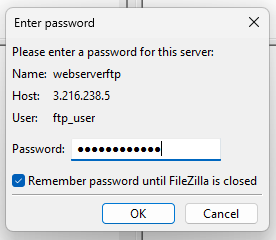


**4.1.6. Transferencia**

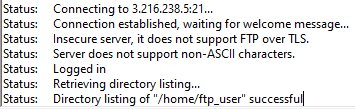
Nos dirigimos a la pestaña de configuración del método de transferencia, donde seleccionaremos el método “**Default**” para que el programa escoja de momento.



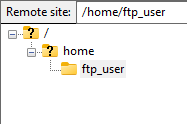
Una vez hagamos clic en el botón “**Connect**”, se nos pedirá la contraseña que establecimos para el usuario con el que estamos realizando la conexión.



Se muestran los siguientes logs en el cuadro superior del programa:



Además, podemos ver cómo el árbol de directorios al que el usuario es capaz de acceder está limitado a su directorio personal, como configuramos anteriormente.

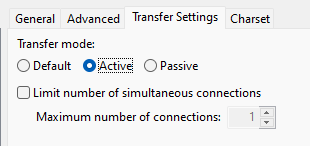


**4.2. Comprobación de los métodos de conexión**

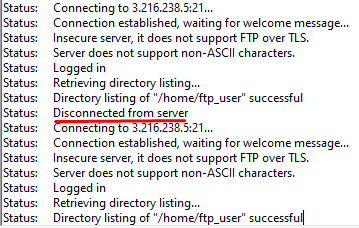
A modo de prueba, vamos a realizar la misma conexión utilizando los métodos activo y pasivo para comprobar que hemos configurado correctamente el servidor.

**4.2.1. Activo**

Nos dirigimos de nuevo a la pestaña File y al Gestor de sitios. Entramos en la pestaña de métodos de transferencia del sitio que hemos creado y seleccionamos el método “**Active**”.



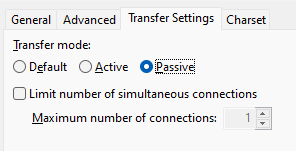
Hacemos clic en el botón Connect y comprobamos que, efectivamente, la conexión se realiza sin problemas.



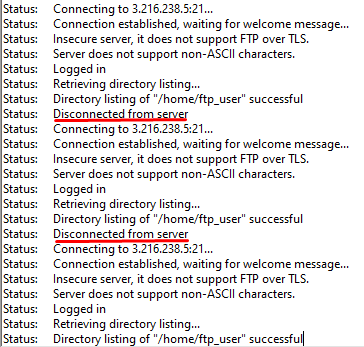
\* En la captura se ve cómo a partir de la línea roja se realiza una nueva conexión.

**4.2.2. Pasivo**

Realizamos los mismos pasos para seleccionar el método de conexión “**Passive**”.



De nuevo, se puede ver cómo se han realizado tres conexiones distintas con, realizadas con distintos métodos de transferencia.



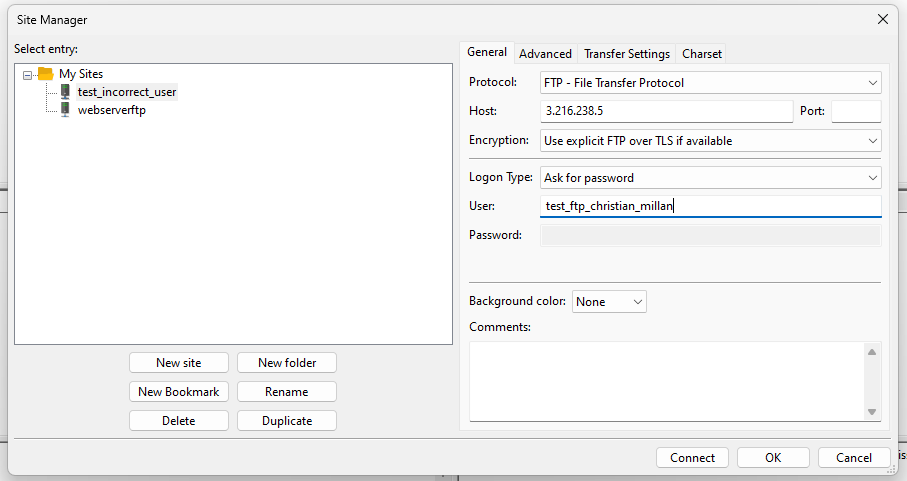
**4.3. Comprobación de validez de usuario**

A modo de prueba, vamos a crear un usuario de prueba en la instancia de AWS y vamos a intentar realizar una conexión con él hacia el servidor FTP.

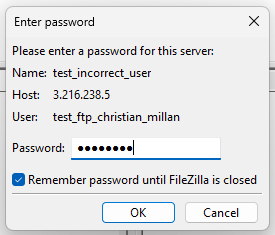
Ejecutamos el comando [“**useradd -m test\_ftp\_christian\_millan && passwd test\_ftp\_christian\_millan**”🔗](https://explainshell.com/explain?cmd=useradd+-m+test_ftp_christian_millan+%26%26+passwd+test_ftp_christian_millan).



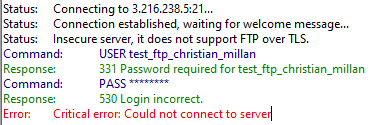
Ahora nos dirigimos de nuevo al cliente FTP, donde crearemos un nuevo sitio llamado “**test\_incorrect\_user**”, con una configuración igual al anterior, pero estableciendo como usuario el recién creado.



Introducimos la contraseña del mismo:



Y podemos ver cómo en el recuadro de logs aparece un mensaje de error indicando que **la conexión no se puedo realizar exitosamente**:



**5. Configuraciones adicionales**

Este último apartado se basa en mostrar algunas **configuraciones adicionales que se consideren oportunas** para el servidor ProFTP.

**5.1. Nombre del servidor**

En la línea 15, podemos encontrar una directiva que establece el nombre del servidor a “**Debian**” de forma nativa. Cambiamos esto para establecer un nombre personalizado para nuestro nuevo servidor.



**5.2. Transfer timeout**

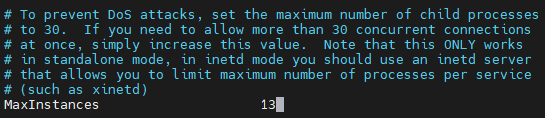
Por defecto, en la línea 25, el tiempo de espera del servidor mientras que no se realiza ninguna acción es de **600 segundos**, podemos cambiar esto a **1200 segundos** (por ejemplo), lo que hará que, cuando no se esté haciendo nada en el servidor, la conexión dure un tiempo máximo de **1200 segundos antes de desconectarse automáticamente**.



**5.3. Número máximo de instancias**

Podemos cambiar el valor establecido para el número total de instancias/conexiones simultáneas que pueden existir hacia nuestro servidor FTP en la línea 67.

Vamos a cambiar de un número máximo de 30 instancias a 13, por ejemplo.



**5.4. Registro de actividades**

En la línea 91 nos encontramos con una directiva que habilita el registro de logs de actividades realizadas en el servidor, lo que puede ser muy útil para llevar cuenta de actividades maliciosas o posibles errores.

Si se comenta esta línea se deshabilita esta característica.



**5.5. Permitir sobrescribir**

Esta directiva situada en la línea 77 permite/prohíbe a los usuarios del servidor el hehco de poder sobrescribir archivos de configuración del mismo.



**5.6. Denegación de filtro**

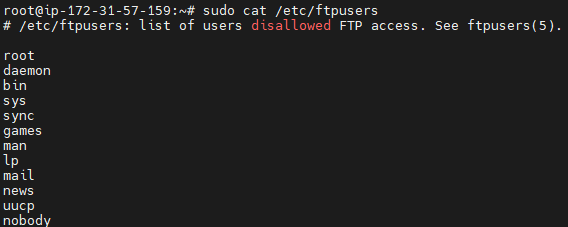
Esta opción permite denegar el acceso a directorios o archivos basados en expresiones regulares. Podemos utilizar esto para configurar reglas de acceso específicas.



**5.7. Usuarios bloqueados**

En ProFTPD, la lista de usuarios bloqueados para la conexión por FTP se encuentra en el archivo **/etc/ftpusers**. Los usuarios listados en este archivo no pueden iniciar sesión a través de FTP.

Con el comando “**sudo cat /etc/ftpupsers**” vemos la lista de estos usuarios.

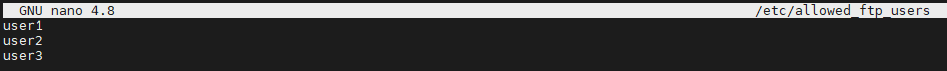


**5.8. Lista de usuarios permitidos**

Por último, vamos a ver cómo crear una lista de usuarios permitidos, para excluir a los usuarios que no se encuentren en ella.

Primero ejecutamos el comando “**nano /etc/allowed\_ftp\_users**” para crear nuestro archivo.

Acto seguido, introducimos la lista de usuarios que queremos permitir, línea a línea.



Guardamos y salimos del archivo.

Ahora nos volvemos a dirigir al archivo de configuración del servidor FTP con el comando “**sudo nano /etc/proftpd/proftpd.conf**”. Nos vamos al final del archivo y hacemos clic en la tecla **Enter** dos veces.

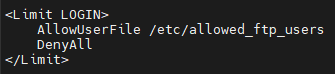
Introducimos el siguiente contenido:

**<Limit LOGIN>**

**AllowUserFile /etc/allowed\_ftp\_users**

**DenyAll**

**</Limit>**



Con esta directiva **Limit** podemos establecer que únicamente la lista de usuarios perteneciente al archivo creado recientemente es válida para realizar conexiones con nuestro servidor FTP.

\* He eliminado esta directiva para no entorpecer las conexiones con el usuario **ftp\_user**.