Prácticas Acceso Remoto

Índice

1. Configurar SSH en una máquina Ubuntu

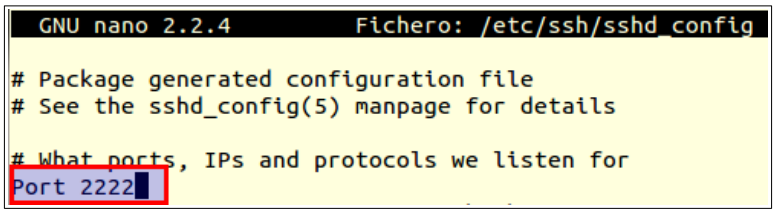
*Configurar SSH en una máquina Linux (**Ubuntu 3**) para que:*

- *El puerto de conexión sea **2222,** diferente al de defecto.*
- *El tiempo de conexión sea como máximo de **15 segundos**. Poco, pero suficiente para conectar teniendo en cuenta el tráfico de la red.*
- *No se permitirá la conexión del usuario **root**.*
- *Se permitirá la conexión del usuario "**usuario1"** y "**administrador**" desde cualquier equipo, pero el usuario **"seguridad"** sólo desde un cliente concreto.*
- *Permitir una sola posibilidad de escribir de forma correcta la contraseña.*

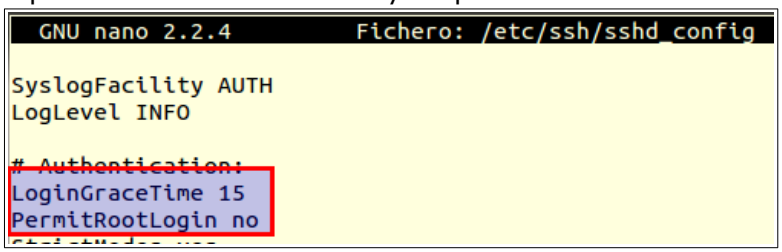
Para la instalación, deberemos ejecutar el comando ***apt-get install openssh-server***, con la precaución de ejecutarlo con los permisos de root.

Tras la instalación procedemos a cambiar la configuración de defecto del fichero */etc/ssh/sshd_config:*

- El puerto por defecto al valor 2222:



Configuramos el tiempo dedicado a la conexión y no permitimos la conexión con el usuario root:



- Permitimos la conexión con el usuario **seguridad** sólo desde un equipo concreto (no permite nombre de dominio) y con el **usuario1** y **administrador** desde cualquier equipo:

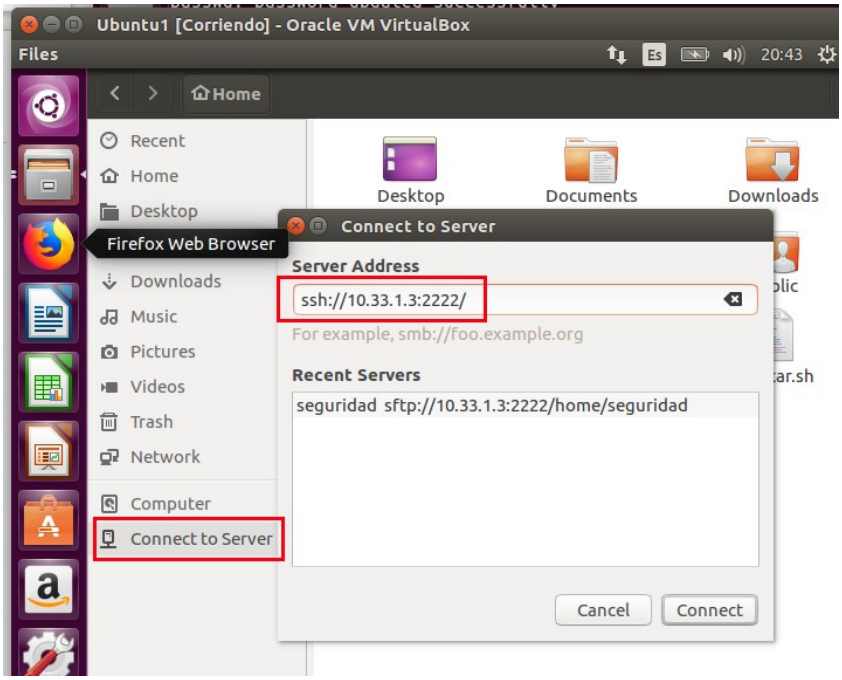
```
AllowUsers seguridad@10.33.1.1 usuario1 administrador
```

- Añadimos la opción de sólo un intento para escribir de forma correcta la contraseña:
MaxAuthTries 1

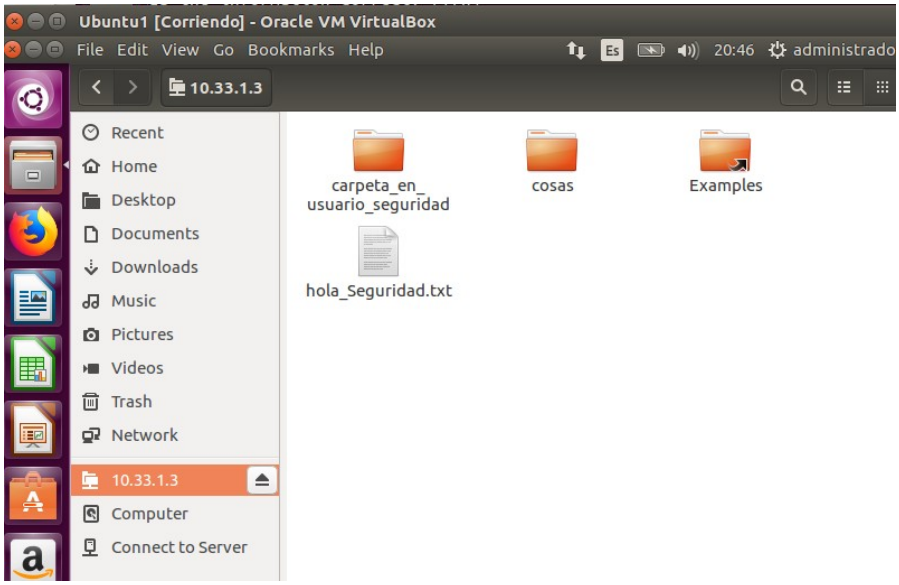
Tras guardar los cambios y reiniciar el servicio (**service ssh reload**) comprobamos su funcionamiento desde el cliente en modo línea de comandos. Usamos como cliente "Debian":

```
root@debian100:~# ssh -p 2222 administrador@10.33.1.3
The authenticity of host '[10.33.1.3]:2222 ([10.33.1.3]:2222)' can't be established.
ECDSA key fingerprint is SHA256:NhOk8A4n4dwJcuNOpZcN/PjAGHMGoY4BnRcP5uwzqMw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.33.1.3]:2222' (ECDSA) to the list of known hosts.
administrador@10.33.1.3's password:
```

Desde Ubuntu1, también podemos acceder utilizando el programa gráfico de conexión incluido por defecto en Ubuntu:



(si el puerto fuera el 22, defecto, no sería necesario ponerlo)
Y si todo funciona correctamente permitirá la conexión mediante el explorador de archivos nautilus:


Ubuntu1 [Corriendo] - Oracle VM VirtualBox
File Edit View Go Bookmarks Help
20:46
administrado
10.33.1.3
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
10.33.1.3
Computer
Connect to Server
carpeta_en_
usuario_seguridad
cosas
Examples
hola_Seguridad.txt

2. SSH Linux transferencia SCP SFTP

Dado el servidor SSH configurado en la práctica anterior:

*A) Realizar una copia de un fichero local al servidor remoto mediante **scp**.*

*B) Realizar una copia de un fichero remoto al servidor local mediante **scp**.*

C) Probar el funcionamiento de la conexión SFTP gráfica desde un cliente Linux transfiriendo ficheros entre el usuario local y el remoto.

Solución: Situados en la máquina cliente (ubuntu1)

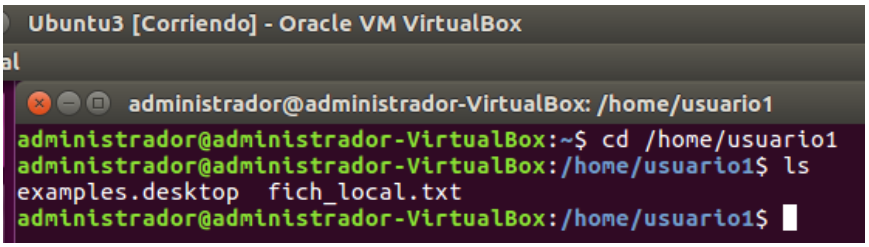
A) Copiamos el fichero local ***fich_local.txt*** (que se encuentra en la máquina cliente "Ubuntu") en el ***home*** del usuario remoto "usuario1" de Ubuntu 3:

```
administrador@administrador-VirtualBox:~$ scp -P 2222 ./fich_local.txt usuario1@10.33.1.3:./
The authenticity of host '[10.33.1.3]:2222 ([10.33.1.3]:2222)' can't be established.
ECDSA key fingerprint is SHA256:NhOk8A4n4dwJcuNOpZcN/PjAGHMGoY4BnRcP5uwzqMw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.33.1.3]:2222' (ECDSA) to the list of known hosts.
usuario1@10.33.1.3's password:
fich_local.txt                                100%   11     0.0KB/s   00:00
```

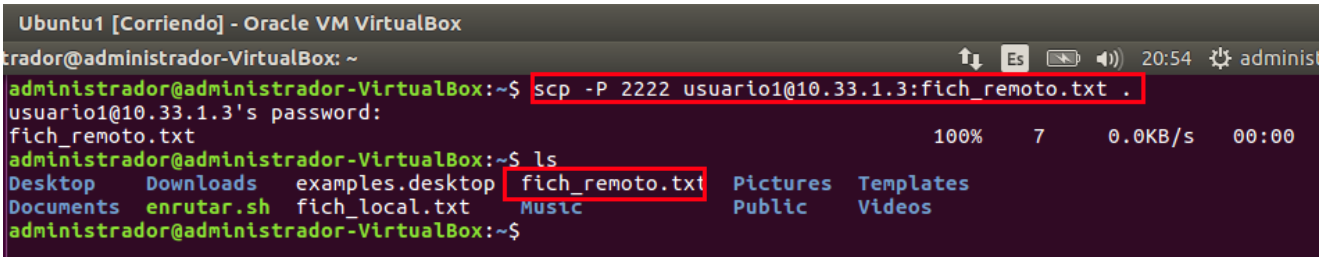
** Ejecutar el comando desde la máquina "Ubuntu 1"

** El parámetro "**-P 2222**" es necesario puesto que el puerto no es el por defecto. El parámetro se sitúa al principio de la línea, tras "scp"

Ahora comprobamos que el fichero se ha copiado en el servidor "Ubuntu 3"

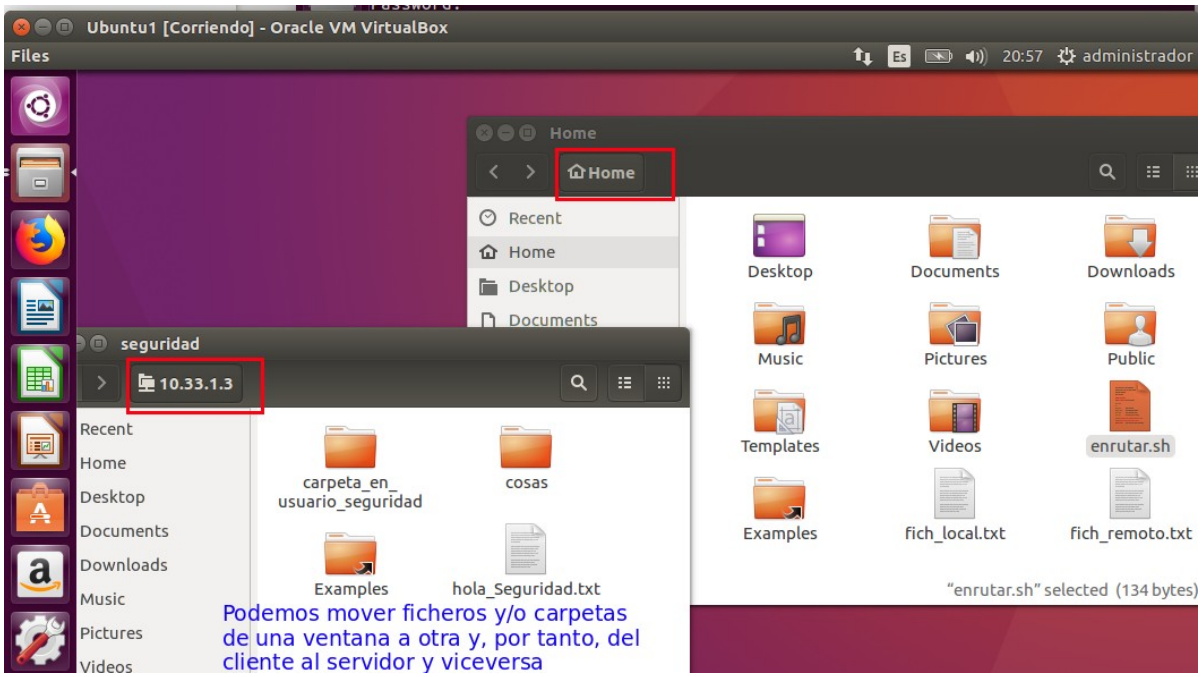


B) Ahora copiamos el fichero remoto ***fichero_remoto.txt*** de Ubuntu3 desde el home del usuario remoto hasta el home local (ubuntu1):



** Ejecutar el comando también desde "Ubuntu 1"

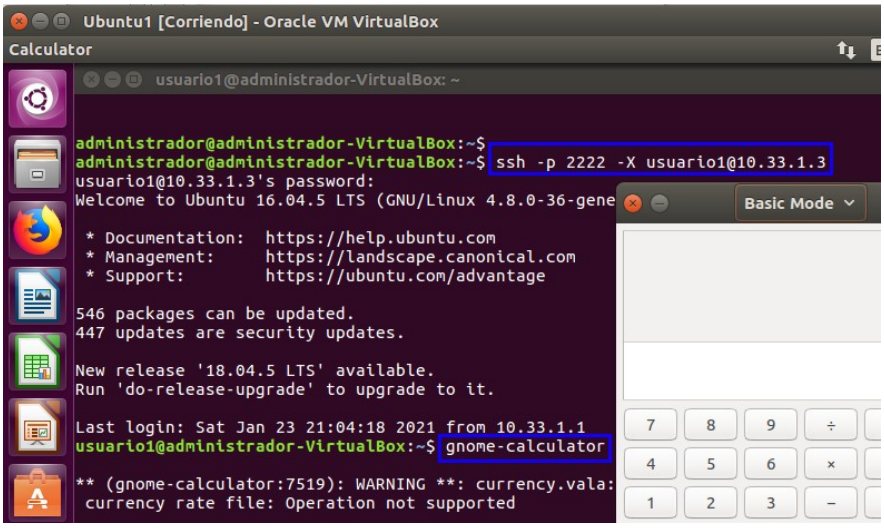
c)



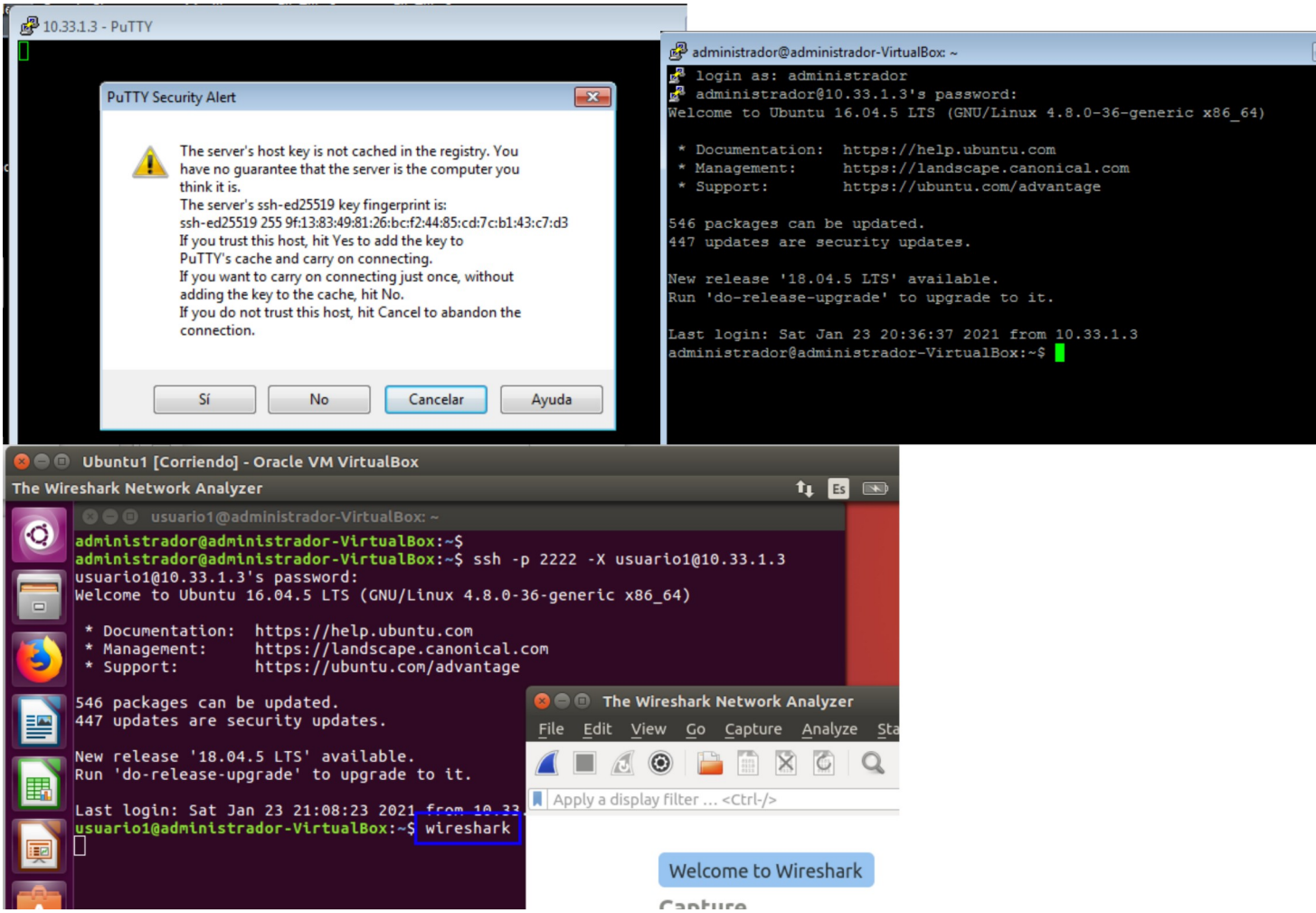
Podemos apreciar que aunque cerremos el explorador de archivos nautilus queda abierta la conexión SFTP:

3. SSH Linux: sesión gráfica

Conectarse al servidor SSH desde un cliente Linux (Ubuntu 1), mediante una sesión gráfica para abrir la calculadora, el navegador y el explorador de archivos.



Al estar en una sesión gráfica podemos ejecutar cualquier programa gráfico del cliente, por ejemplo, la calculadora (comando "**gcalctool**" o "**gnome-calculator**"), un navegador (comando "**firefox**"). Para asegurarse que el programa que se abre es el del servidor remoto podríamos, por ejemplo, añadir marcadores en el firefox del servidor y comprobar que se ven al abrirlo desde el cliente. También podemos ejecutar software de Ubuntu3, por ejemplo, el wireshark:



4. Configurar SSH Linux cliente Windows

Dado el servidor SSH Linux configurado en una práctica anterior, acceder a él desde línea de comando con un cliente Windows y probar su funcionamiento.

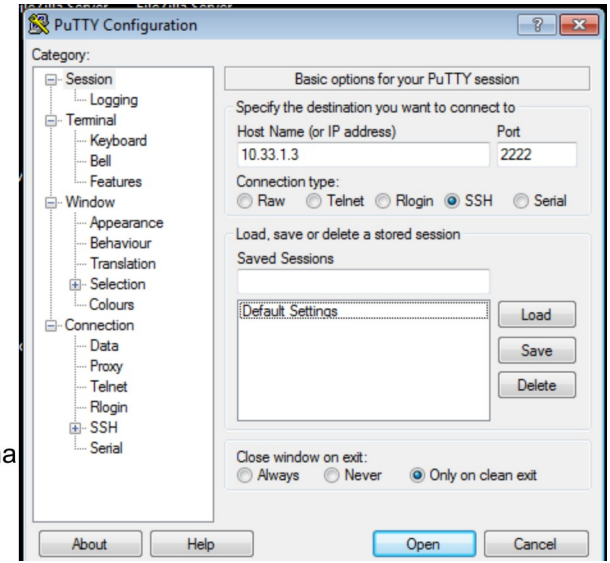
a) Linea de comandos:

Usar el comando "ssh usuario1@10.33.1.2 -p 2222"

- Es necesario tener instalado un cliente SSH. Si usamos la máquina en la que está instalado "openssh" funciona.
- La opción "-p" es necesaria si el puerto no es 22.
- Si superamos el tiempo de conexión (15 segundos) se cerrará la conexión
- Probar la conexión con los usuarios "seguridad" y "usuario1" desde Windows1 y Windows2
- http://sysadm.mielnet.pl/no-kex-alg/

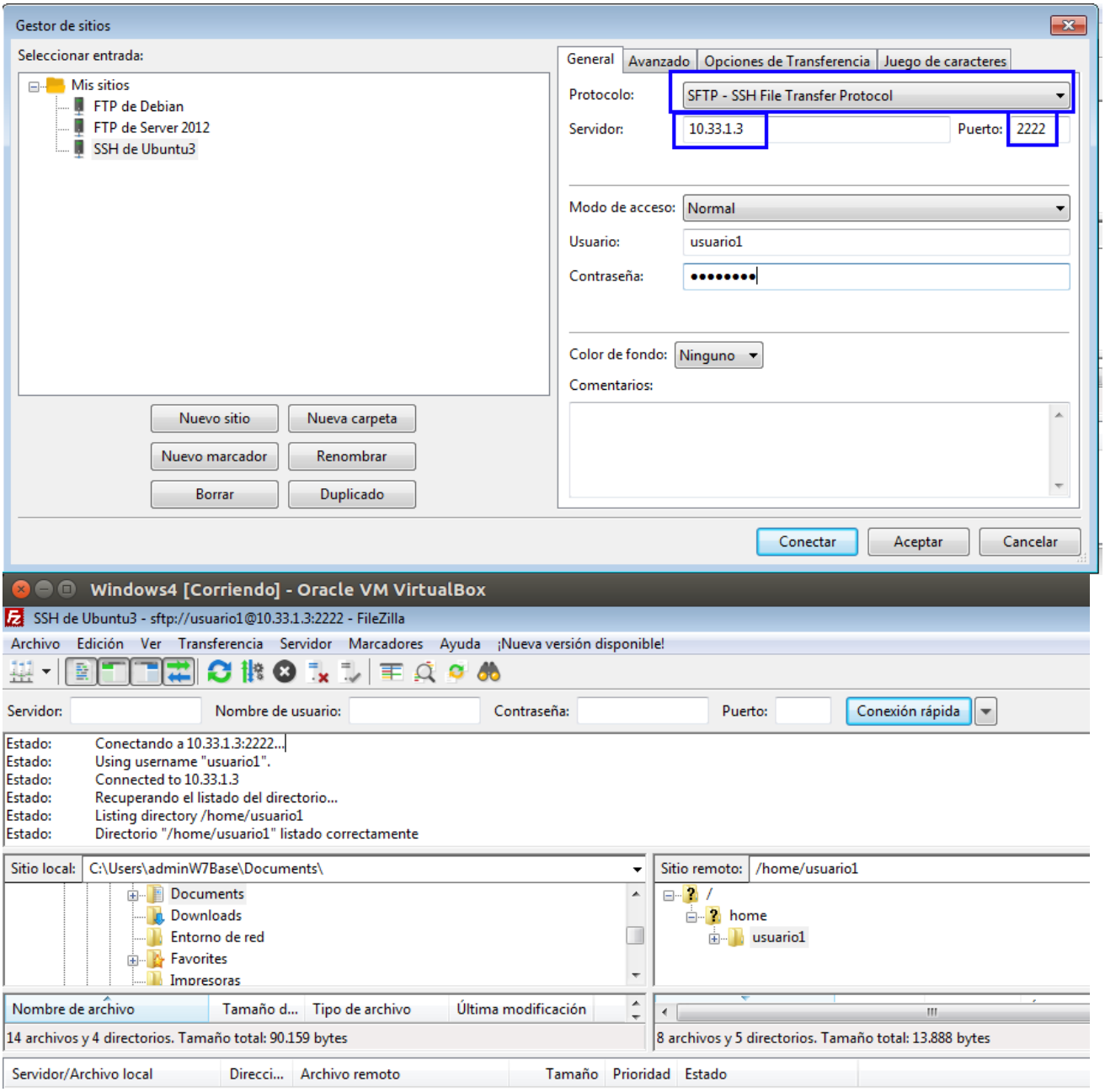
b) Conexión usando "putty"

Nos descargamos el programa putty de la página oficial. Es un cliente Telnet y SSH:



5. SSH Linux transferencia de ficheros desde Windows

Para transferir ficheros de la máquina Linux a Windows y viceversa usando el servicio SSH podemos usar "FilezillaClient" con la siguiente configuración (usando el procotolo **SFTP**):

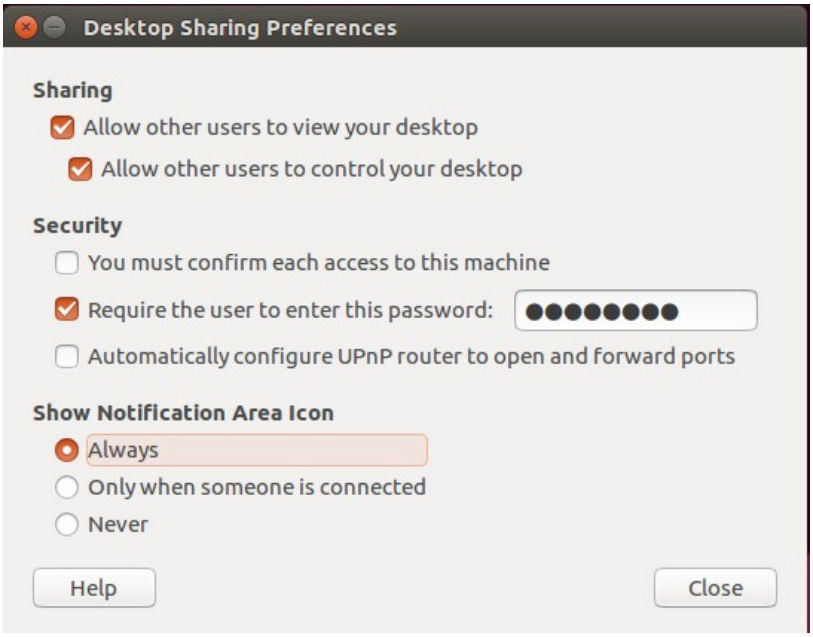


6. Escritorio remoto: VNC Linux cliente Linux

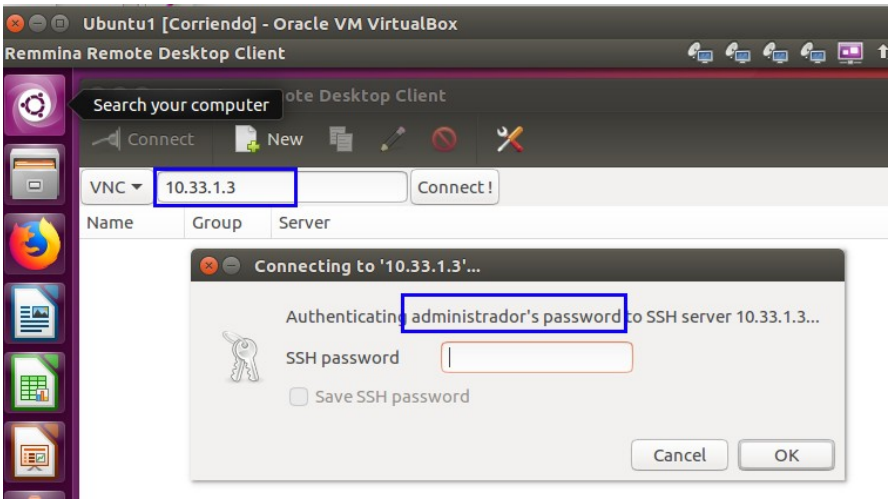
Configurar el control por VNC a un servidor Linux con las siguientes consideraciones:
-Permitiendo que otros usuarios puedan controlar mi equipo accediendo mediante un usuario del sistema al que deseo conectarme.
- Indicar una contraseña específica (distinta a la contraseña local del usuario con el que me conecto).
- No debe existir la necesidad de confirmar de forma local el acceso.
- Además queremos que nos muestre un icono cuando hay un usuario conectado.

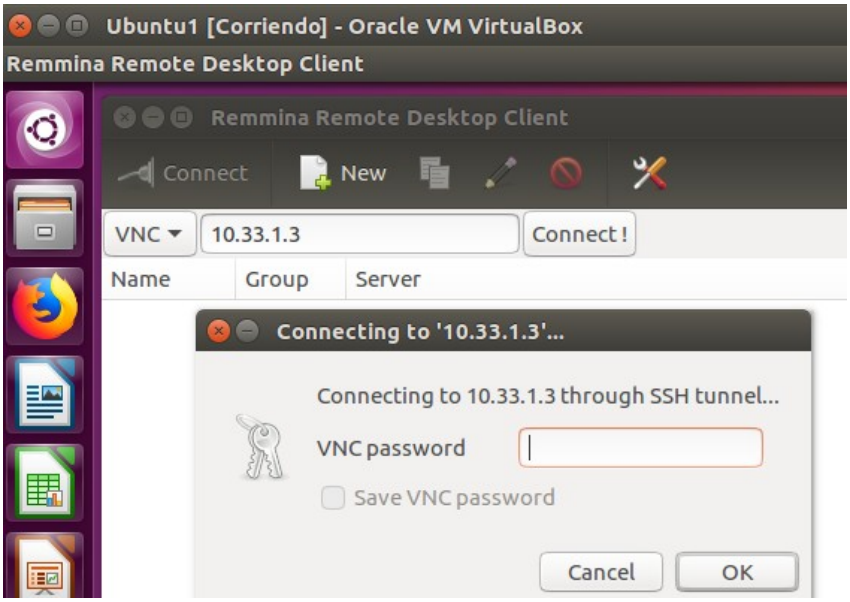
Acceder a él mediante un cliente Linux y probar su funcionamiento.

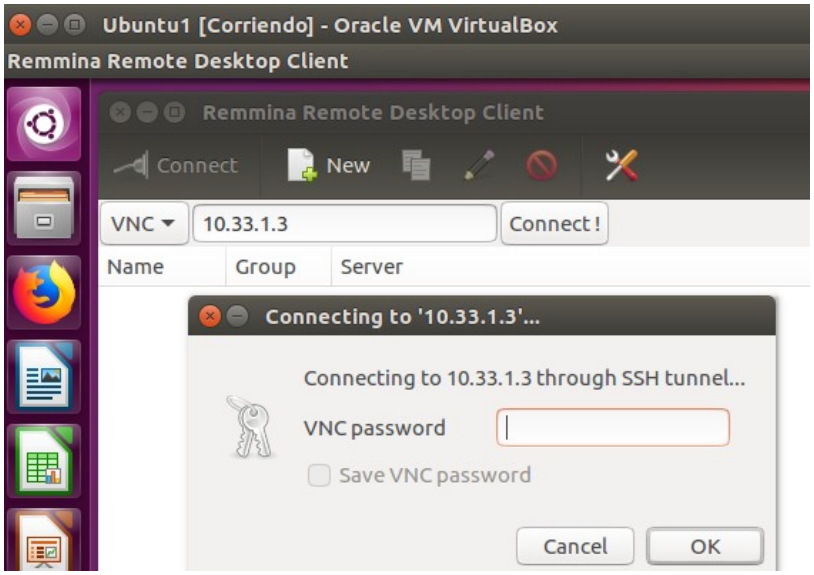
- Vamos a compartir el escritorio de **Ubuntu3** y accedemos a él en remoto desde **Ubuntu1:**
  En Ubuntu3: **Sistema / Preferencias / Compartición de escritorio** (buscar **DESKTOP SHARING)**

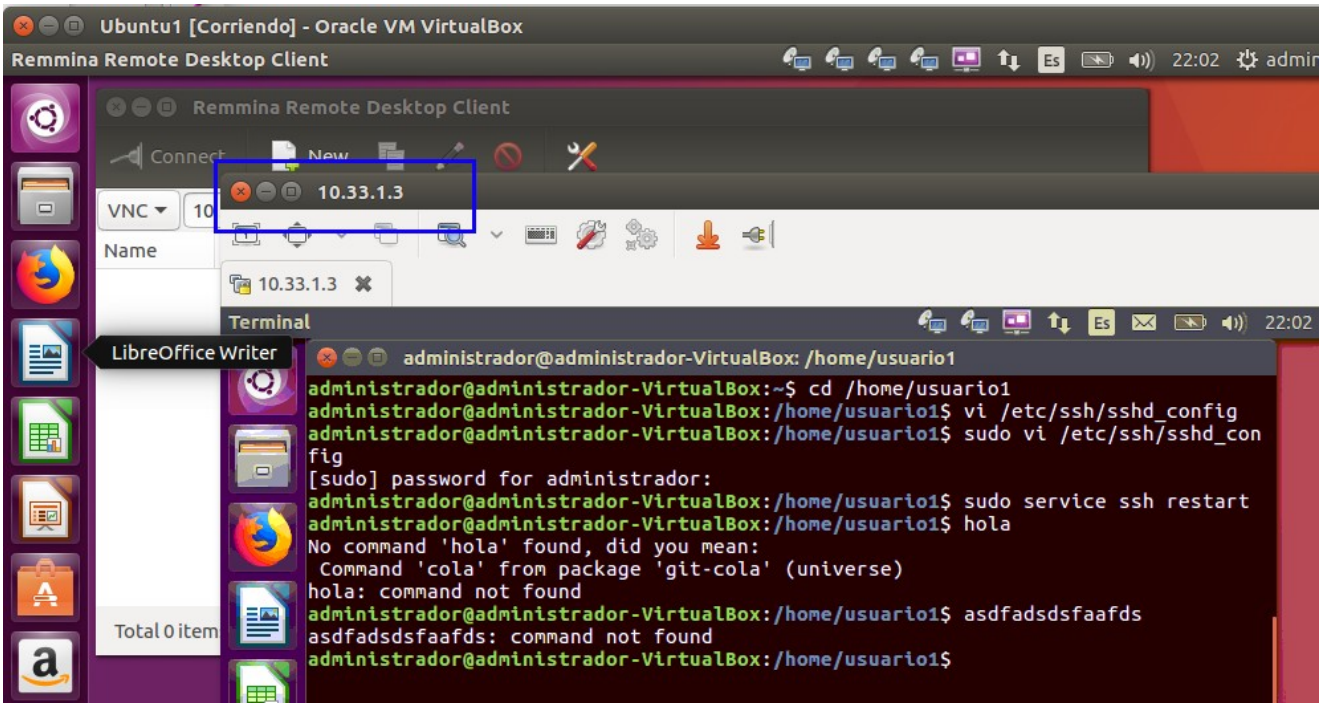


En el cliente remoto desde **Ubuntu3**: *Aplicaciones / Internet / Cliente de escritorio remoto Remmina*




Ubuntu1 [Corriendo] - Oracle VM VirtualBox
Remmina Remote Desktop Client
Remmina Remote Desktop Client
Connect
New
VNC
10.33.1.3
Connect !
Name
Group
Server
Connecting to '10.33.1.3'...
Connecting to 10.33.1.3 through SSH tunnel...
VNC password
Save VNC password
Cancel
OK





7. Configurar conexiones SSH sin password (usando claves privadas/públicas)

Ahora vamos a ver como conectarse a un PC remotamente por **SSH** tan solo introduciendo el password una vez durante la configuración; luego, aunque reiniciemos ambos ordenadores, no se nos volverá a pedir el password del servidor SSH.

Usaremos dos máquinas Ubuntu de la misma red interna: Una tendrá el servidor SSH (Ubuntu3) y otra será el cliente que se conecta a ella (Ubuntu1).

Volver a poner puerto 22 a servidor SSH y aumentar el valor del parámetro MaxAuthTries

1. En **Ubuntu 1 (cliente SSH)** escribimos lo siguiente:

ssh-keygen -b 4096 -t rsa

Esto generará un par de claves. La clave privada no puede salir de esta máquina, pero la clave pública se enviará a los equipos con los que esta máquina se pueda comunicar. En este caso, queremos enviar la clave pública al servidor SSH.

Al escribir el comando aparecerá:

```
administrador@administrador-VirtualBox:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/administrador/.ssh/id_rsa):
```

Aceptamos la ruta en la que se guardará la clave y introducimos una frase dos veces (será la contraseña de la clave). Al finalizar obtenemos este mensaje:

```
administrador@administrador-VirtualBox:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/administrador/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/administrador/.ssh/id_rsa.
Your public key has been saved in /home/administrador/.ssh/id_rsa.pub.
The key fingerprint is:
b4:b0:48:93:05:e7:dd:22:b2:be:09:da:50:b0:76:b6 administrador@administrador
-VirtualBox
The key's randomart image is:
+--[ RSA 4096]----+
|     ..o         |
|      = . .      |
|.    = + + .     |
| o . = = o       |
|...oo . S        |
|..o..            |
|. .E.            |
| + . o           |
|. . o            |
+-----------------+
administrador@administrador-VirtualBox:~$
```

Ya tenemos las claves. Ahora falta dar la pública a quien queramos, en este caso al servidor SSH.

2. También en **Ubuntu 1 (cliente SSH),** escribimos lo siguiente para enviar la clave pública al servidor:

- **ssh-copy-id administrador@10.33.1.3**

Esto lo que hace es simplemente darle la llave pública a **Ubuntu3**, o sea, **Ubuntu3** ya tiene la llave pública de **Ubuntu1**.

Es importante no equivocarse con el usuario. Primero, si el usuario "**administrador**" no existe en Ubuntu3 nos dará error. Pero además es importante tener claro qué usuario usaremos para esto, ya que el usuario con el que configuramos el acceso **sin password** será el mismo con que deberemos acceder en el futuro. Para los demás usuarios se pedirá password a no ser que hagamos para ellos lo mismo.

3. Para probar, nos conectamos desde **Ubuntu1 a Ubuntu 3** por SSH:

- **ssh administrador@10.33.1.3**

Si es la primera vez que accedemos en la sesión, comprobamos que nos pide la contraseña de la clave privada, ya que ahora es la que se va a usar para el acceso. OJO: *No está pidiendo la contraseña del usuario de Ubuntu3, si no la contraseña de la clave privada generada al principio de la práctica.*

*** Si da el error "agent admitted failure to sign", lo arreglamos indicando al agente ssh del cliente que use esa clave para conectarse con el servidor usando el comando:**

ssh-add

Opcionalmente, se le puede dar al comando la ruta del fichero de clave privada (necesario si ese fichero no tiene el nombre por defecto)

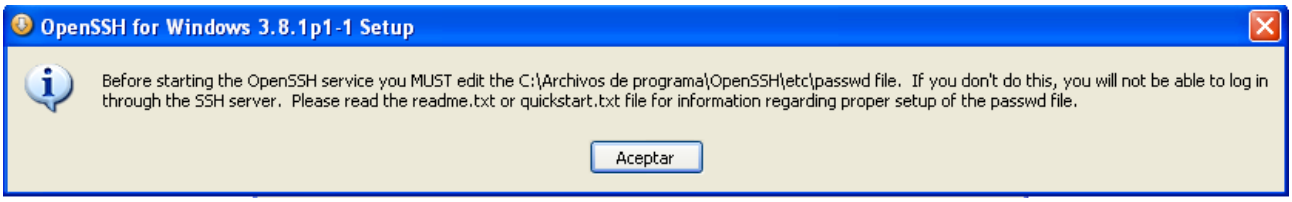
Si queremos acceder a otro ordenador sin introducir tampoco password simplemente le damos nuestra llave pública y listo.

Y si queremos acceder también sin contraseña desde otro PC al servidor SSH (ubuntu3), en el nuevo PC tendremos que generar las claves y enviarle la pública al servidor.

8. Instalar OpenSSH en Windows (Windows 4)

El protocolo SSH tiene el mismo objetivo que el telnet (abrir consolas de máquinas remotas en otros equipos para poder ejecutar comandos), pero cifra la información, lo que lo hace más seguro.
Se descarga OpenSSH de la página http://sshwindows.sourceforge.net - downloads – Binary Installer Release (no escoger el zip de arriba, sino el 3.8p... de la lista. Al pinchar sale un zip.)

➢Se comienza a instalar. En un momento dado aparece este mensaje:



➢Como nos indican en el fichero de ayuda c:\Archivos de programa\OpenSSH\ docs\quickstart.txt, cuando se acaba de instalar, abrir una pantalla de MS Dos y ejecutar desde c:\Archivos de programa\OpenSSH\bin

mkgroup -l >>..\etc\group Genera grupo local
mkpasswd -l >>..\etc\passwd Genera usuarios locales con acceso al servidor
*chown adminW7Base **
*chmod 700 **

** Para que funcionen las dos últimas sentencias se deben pegar en la carpeta "bin" las dll's que se adjuntan en el blog.

**OJO: La segunda sentencia genera en el fichero "passwd" los nombres de usuarios que hay en la máquina windows. El usuario que se vaya a usar para la conexión vía ssh debe tener contraseña.

➢Editamos (edit) el fichero passwd y editamos la línea de nuestro usuario (el que vamos a usar para la conexión) modificando la parte /home/Administrador por /cygdrive/c/ruta_carpeta_por_defecto (se refiere a la carpeta en la que nos conectaremos por defecto cuando hagamos el ssh, c:/ruta_carpeta_por_defecto).

- Si posteriormente añadimos un usuario a Windows y queremos conectarnos con ese usuario en el SSH, lo añadimos al fichero "passwd" con la sentencia

mkpasswd -l -u usuario_nuevo >> ..\etc\passwd

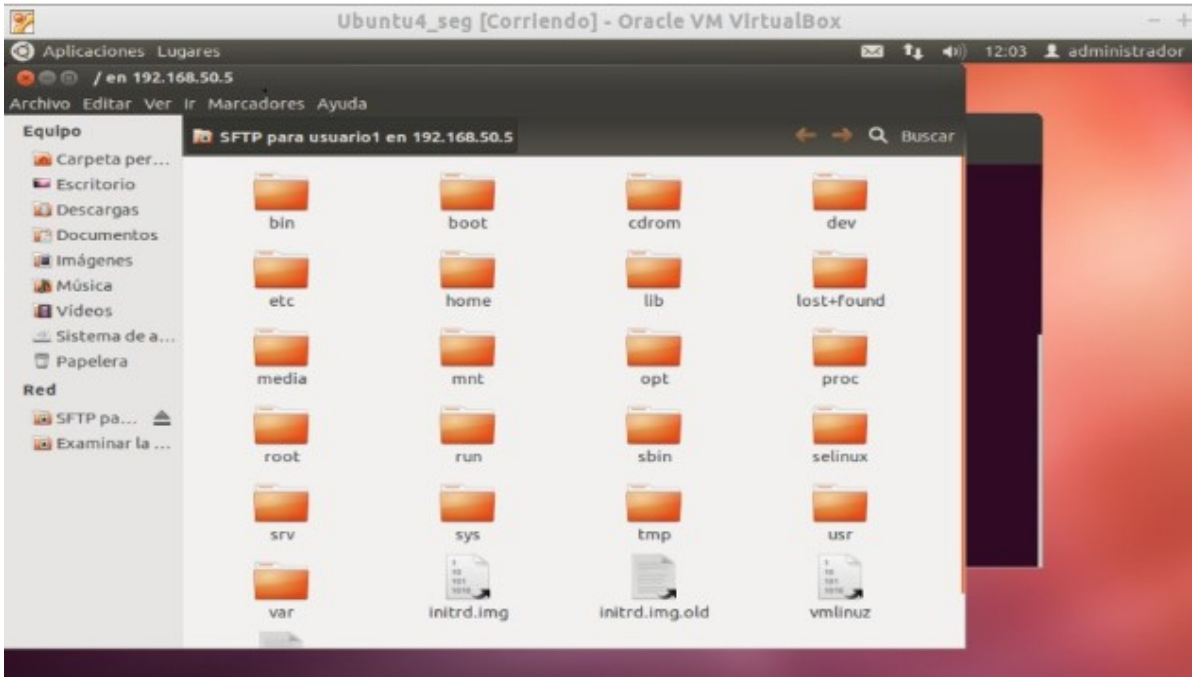
- Además, si cambiamos la contraseña de alguno de los usuarios que usamos para el SSH, debemos generar de nuevo la línea de ese usuario en el fichero "passwd", con la siguiente sentencia (previamente se debe borrar del fichero la línea del usuario):
mkpasswd -l -u usuario_modificado >> ..\etc\passwd

➢Arrancar el servicio desde la pantalla de DOS (*en modo administrador*): ***net start opensshd***

➢Conectarse al equipo desde sí mismo
ssh nombre_usuario@192.168.50.1

➢Ahora vamos a probar desde otro equipo de la red, por ejemplo, Ubuntu1.

SSH también permite conectarse al explorador gráfico de archivos de la máquina remota para hacer transferencia de ficheros. Para probarlo, desde "Ubuntu4" abrimos Nautilus y pulsamos la opción "Conectar con el servidor". En la ventana que aparece, introducimos en la casilla "servidor" el comando
ssh://10.33.1.4 .Y si todo funciona correctamente permitirá la conexión mediante el explorador de archivos nautilus, desde el que podemos copiar archivos de la máquina remota a la local o viceversa.



Si el cliente fuera otra máquina Windows nos conectaríamos con putty

9. Instalar FreeSSH en Windows (Windows 4)

Previamente, parar el OpenSSH de la práctica anterior.
Seguir los pasos de la guía:

https://www.redeszone.net/windows/freesshd-para-windows-instalacion-y-manual-de-configuracion-de-freesshd-para-windows-servidor-ssh-y-sftp/