2021

Práctica Acceso Remoto

FRANCISCO JAVIER LÓPEZ CALDERÓN

Prácticas Acceso Remoto

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.

```
administrador@Ubuntu3:~$ sudo apt-get install openssh-server
sudo: unable to resolve host Ubuntu3
[sudo] password for administrador:
Reading package lists... Done
```

Tras la instalación procedemos a cambiar la configuración de defecto del fichero */etc/ssh/sshd_config:*
El puerto por defecto al valor 2222:

```
# What ports, IPs and protocols we listen for
Port 2222
# Use these options to restrict which interfac
#ListenAddress ::
```

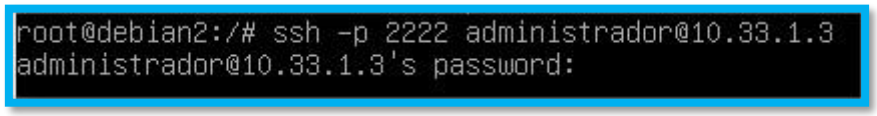
Configuramos el tiempo dedicado a la conexión y no permitimos la conexión con el usuario root:

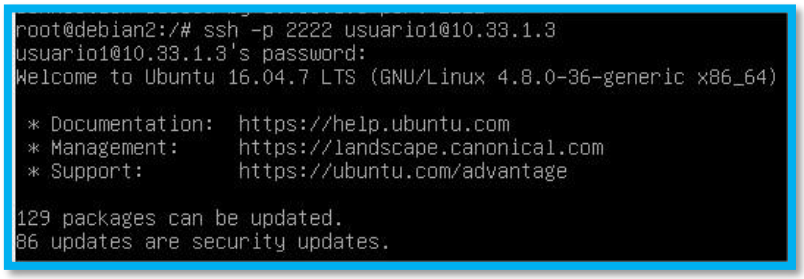
```
# Authentication:
LoginGraceTime 15
PermitRootLogin no
StrictModes yes
```

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

```
AllowUsers seguridad@10.33.1.3 usuario1 adminitrador
MaxAuthTries 1
```

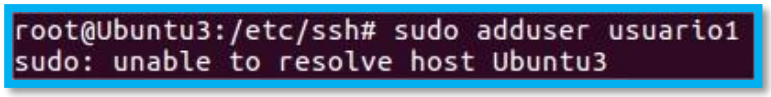
Tras guardar los cambios y reiniciar el servicio (**service ssh reload**) comprobamos su funcionamiento desde el cliente en modo línea de comandos. Usamos como cliente "Debian":

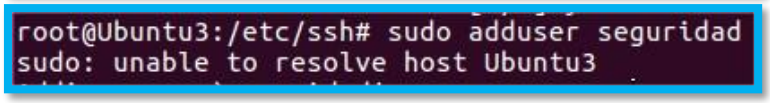




Desde Ubuntu1, también podemos acceder utilizando el programa gráfico de conexión incluido por defecto en Ubuntu:

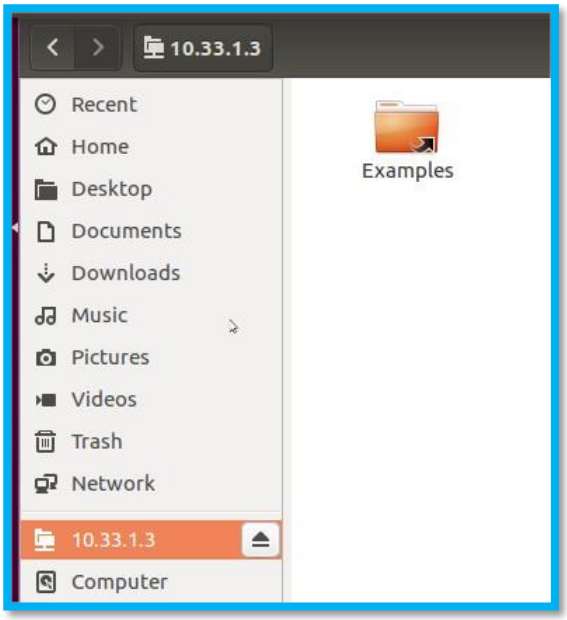
Crearemos el usuario "usuario1" y "seguridad" importante recordar sus contraseñas





Introducimos la dirección siguiente

Hemos accedido de forma gráfica al servidor ssh.



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:

Creamos el archivo testeo.txt en Ubuntu1 (cliente)

```
administrador@Ubuntu1:~$ cat testeo.txt
hola friends 1
```

```
administrador@Ubuntu1:~$ scp -P 2222 ./testeo.txt usuario1@10.33.1.3:./
usuario1@10.33.1.3's password:
testeo.txt                                    100%   16     0.0KB/s   00:00
```

Desde Ubuntu1 enviamos el archivo testeo a Ubuntu3

En la home del usuario1 (home/usuario1)

```
administrador@Ubuntu3:/home/usuario1$ ls
examples.desktop  prueba.txt  testeo.txt
```

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

Crearemos en Ubuntu3 el documento prueba.txt en el home de usuario1

```
administrador@Ubuntu3:/home/usuario1$ ls
examples.desktop  prueba.txt
administrador@Ubuntu3:/home/usuario1$
```

Desde Ubuntu1 traemos el archivo desde Ubuntu3 llamado prueba.txt

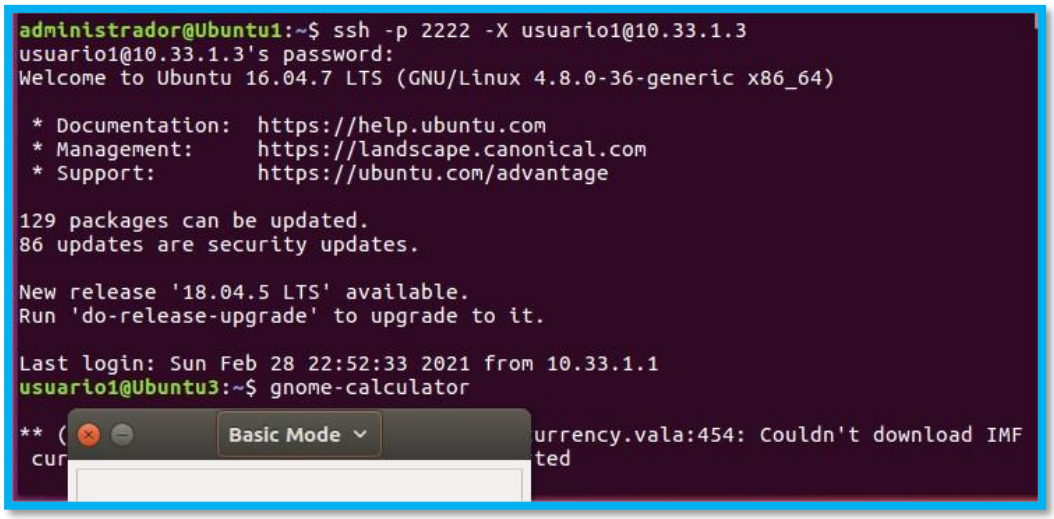
```
administrador@Ubuntu1:~$ scp -P 2222 usuario1@10.33.1.3:prueba.txt .
usuario1@10.33.1.3's password:
prueba.txt                                    100%   14     0.0KB/s   00:00
```

Hacemos un ls y comprobamos que el archivo prueba.txt se encuentra aquí

```
administrador@Ubuntu1:~$ ls
Desktop     Downloads     examples.desktop  Pictures
Documents   enrutador.sh  Music             prueba.txt
```

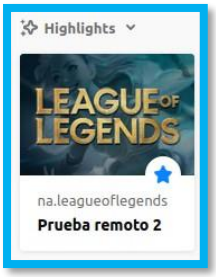
3. SSH Linux: sesión gráfica

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:

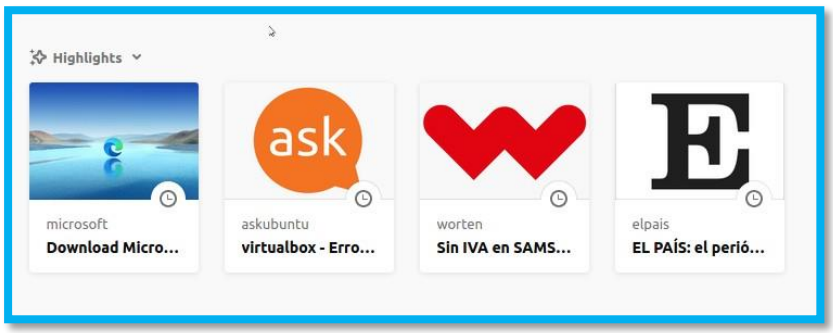
Desde Ubuntu1 accedemos al servidor ssh



Primero añadiremos el marcador en Ubuntu1 utilizando Firefox en el servidor remoto, en este caso, la página "na.leagueoflegends"

Ahora, accederemos a Firefox original de Ubuntu1

Comprobamos que no posee los marcadores que habíamos utilizado previamente.



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.

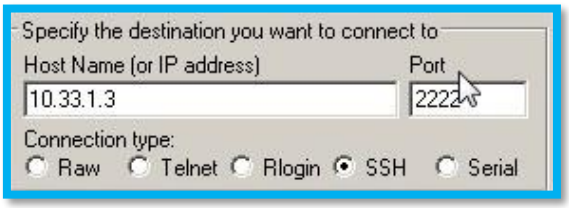
En este caso utilizaremos putty

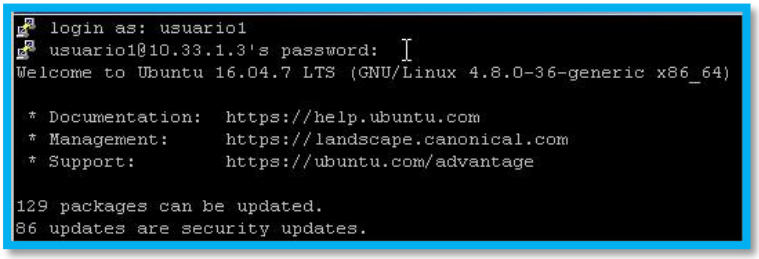
a) Linea de comandos:

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

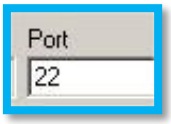
- Es necesario tener instalado un cliente SSH. Si usamos la máquina en la que está instalado "openssh" funciona.

En login as introducimos el usuario "usuario1" e introducimos su password





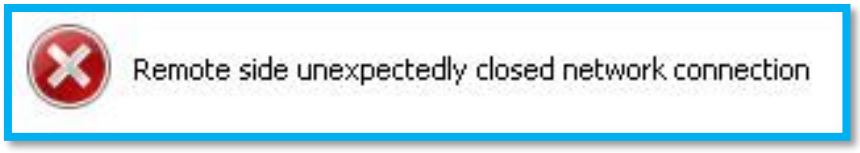
- La opción "-p" es necesaria si el puerto no es 22.



Putty nos permite introducir el puerto deseado, en este caso, usamos 2222

- Si superamos el tiempo de conexión (15 segundos) se cerrará la conexión

Accedemos al servidor ssh y esperamos 15 segundos para que nos expulse el servidor



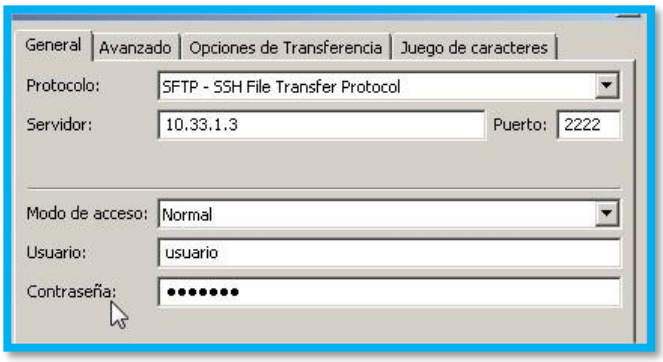
5. SSH Linux transferencia de ficheros desde Windows

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**):

Se procederá a acceder al servidor ssh desde el cliente filezilla.

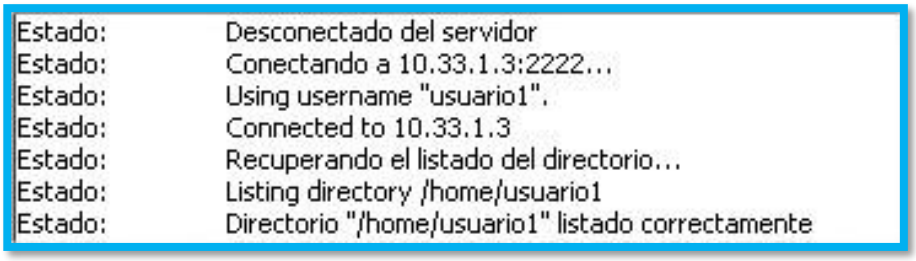
Primero, accedemos al gestor de sitios

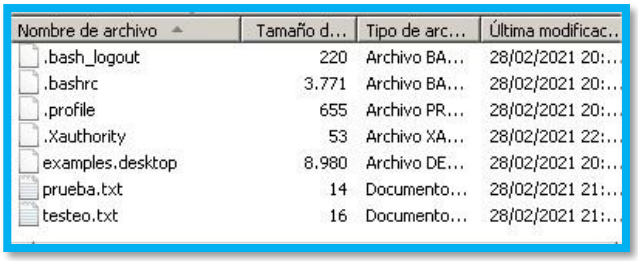
En el Gestor, nombramos el sitio y rellenamos los datos correspondientes



Una vez terminado, pulsamos "Conectar"

Comprobamos el estado de conexión, en este caso, sin problemas.





Nombre de archivo	Tamaño d...	Tipo de arc...	Última modificac...
.bash_logout	220	Archivo BA...	28/02/2021 20:...
.bashrc	3.771	Archivo BA...	28/02/2021 20:...
.profile	655	Archivo PR...	28/02/2021 20:...
.Xauthority	53	Archivo XA...	28/02/2021 22:...
examples.desktop	8.980	Archivo DE...	28/02/2021 20:...
prueba.txt	14	Documento...	28/02/2021 21:...
testeo.txt	16	Documento...	28/02/2021 21:...

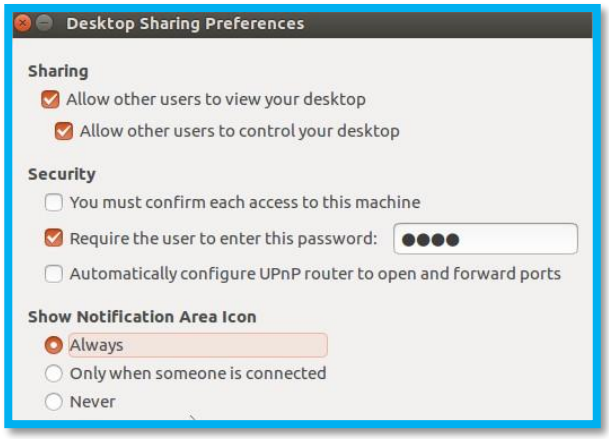
Podemos ver que es el servidor correcto, ya que poseen, los archivos "prueba.txt" y "testeo.txt"

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)**

Desde Ubuntu 3 buscamos el "Desktop Sharing"

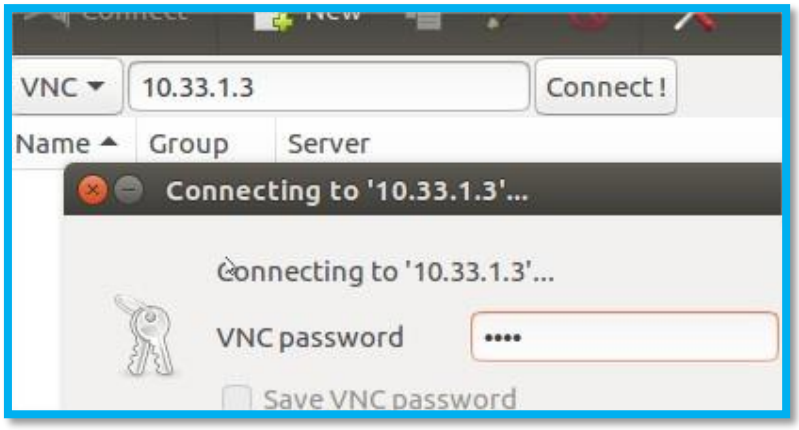


Realizamos los pasos e introducimos la contraseña "1234"

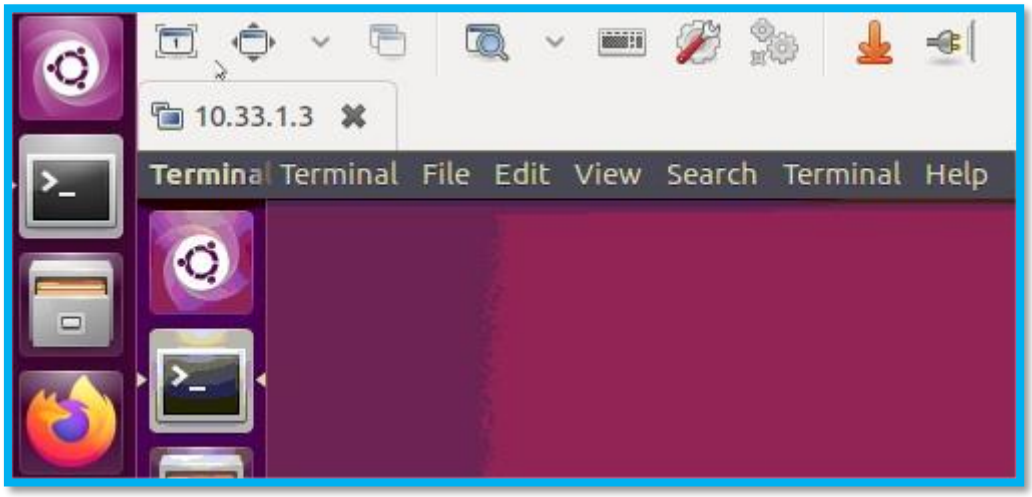
En el cliente remoto desde **Ubuntu1**: *Aplicaciones / Internet / Cliente de escritorio remoto Remmina*

Accedemos a la aplicación "Remmina Remote Desktop Cliente"

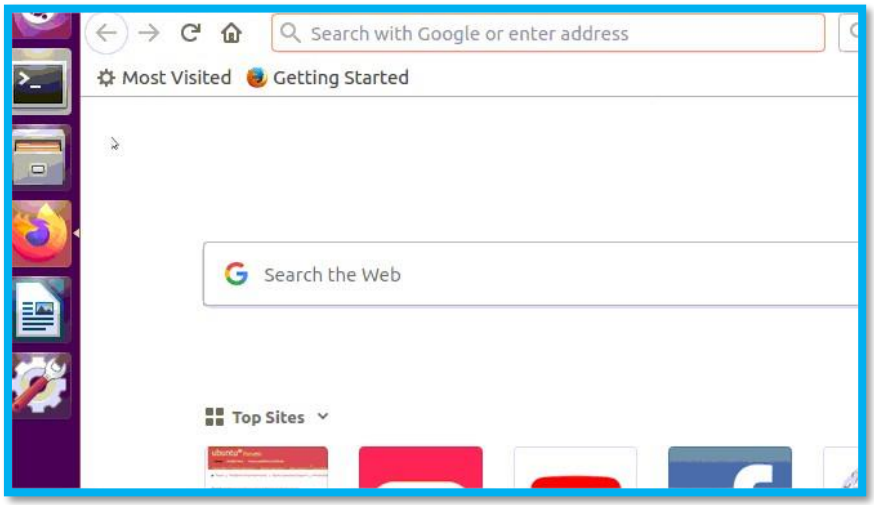
Introducimos la IP del servidor ssh y utilizamos la contraseña anteriormente creada (1234)



Acto seguido, aparecerá el escritorio remoto de Ubuntu 3



Podemos cargar aplicaciones sin problemas como Firefox

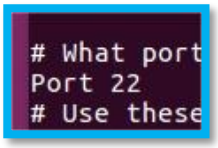


7. Configurar conexiones SSH sin password (usando claves privadas/públicas)

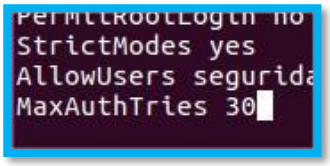
Ahora vamos a ver como conectarse a un PC remotamente por **SSH** tan solo introduciendo el password una vez durante la configuración; luego, aunque reiniciemos ambos ordenadores, no se nos volverá a pedir el password del servidor SSH.

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



Modificamos el puerto a 22



Y reiniciaremos el servicio ssh

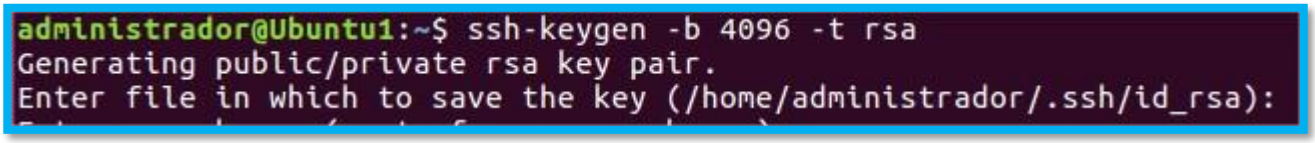
1. En **Ubuntu 1 (cliente SSH)** escribimos lo siguiente:

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.

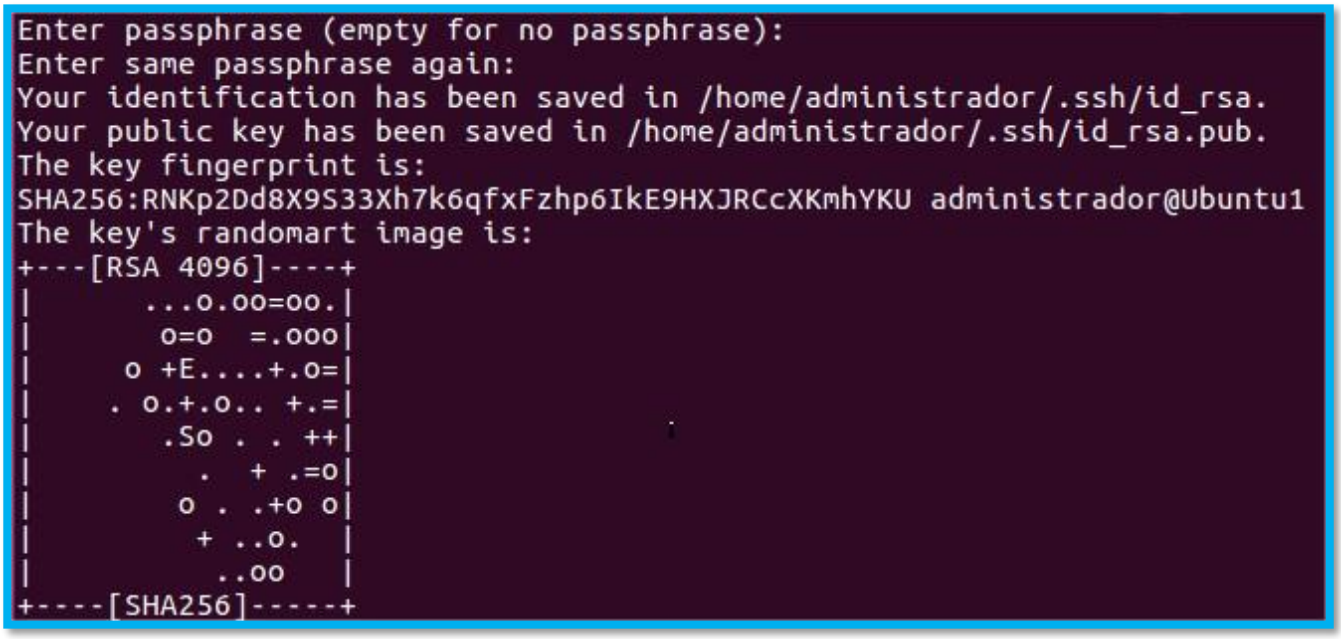
Al escribir el comando aparecerá:

Desde Ubuntu1 realizamos los pasos



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:

Utilizaremos la contraseña 1234qwerty



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.

Enviamos la clave pública a Ubuntu3

```
administrador@Ubuntu1:~$ ssh-copy-id administrador@10.33.1.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/administrad
or/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
administrador@10.33.1.3's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'administrador@10.33.1.3'"
and check to make sure that only the key(s) you wanted were added.
```

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.*

Introducimos la contraseña anteriormente creada (1234qwerty)

```
administrador@Ubuntu1:~$ ssh administrador@10.33.1.3
Enter passphrase for key '/home/administrador/.ssh/id_rsa':
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.8.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

129 packages can be updated.
86 updates are security updates.
```

*** 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.

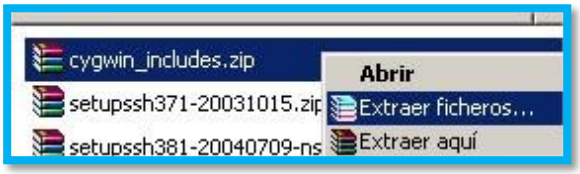
Y si queremos acceder también sin contraseña desde otro PC al servidor SSH (ubuntu3), en el nuevo PC tendremos que generar las claves y enviarle la pública al servidor.

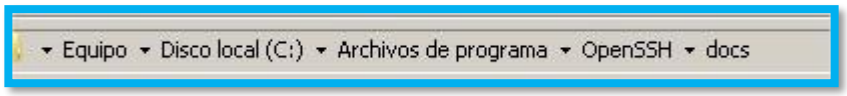
8. Instalar OpenSSH en Windows (Windows 4)

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,

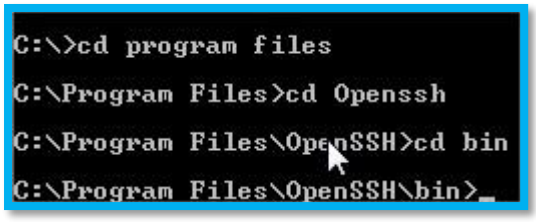
Extraemos los ficheros .dll necesarios en la carpeta bin localizada en c/program files/openssh/bin





cuando se acaba de instalar, abrir una pantalla de MS Dos y ejecutar desde c:\Archivos de programa\OpenSSH\bin

Accedemos a la ruta



mkgroup -l >>..\etc\group Genera grupo local

Generamos el grupo local

mkpasswd -l >>..\etc\passwd Genera usuarios locales con acceso al servidor

Generamos usuarios locales

*chown adminW7Base **

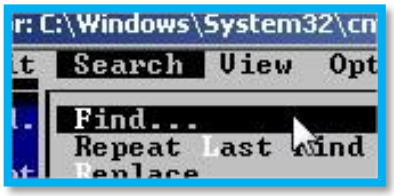
*chmod 700 **

** 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).

Editamos el fichero mkpasswd.c



Seleccionamos "Find..." para localizar el sitio y escribimos /home/

Modificamos la ruta

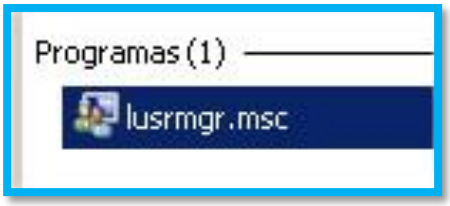
- Si posteriormente añadimos un usuario a Windows y queremos conectarnos con ese usuario en el SSH, lo añadimos al fichero "passwd" con la sentencia *mkpasswd -l -u usuario_nuevo >> ..\etc\passwd*

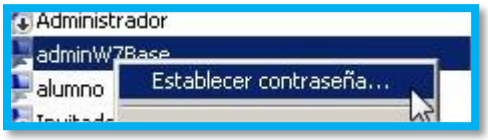
- Además, si cambiamos la contraseña de alguno de los usuarios que usamos para el SSH, debemos generar de nuevo la línea de ese usuario en el fichero "passwd", con la siguiente sentencia (previamente se debe borrar del fichero la línea del usuario): *mkpasswd -l -u usuario_modificado >> ..\etc\passwd*

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

Modificaremos el usuario adminW7Base para que tenga una contraseña

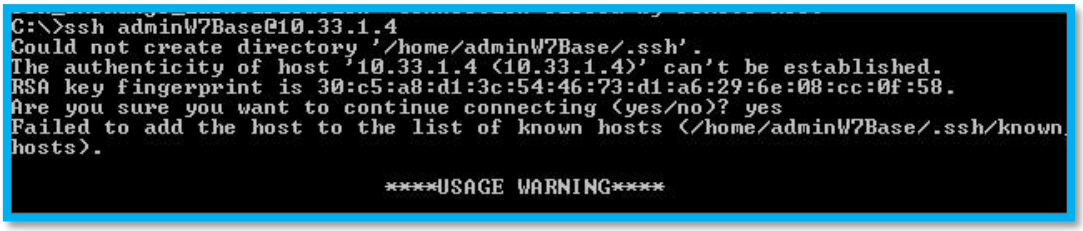
Accederemos a lusrmgr.msc



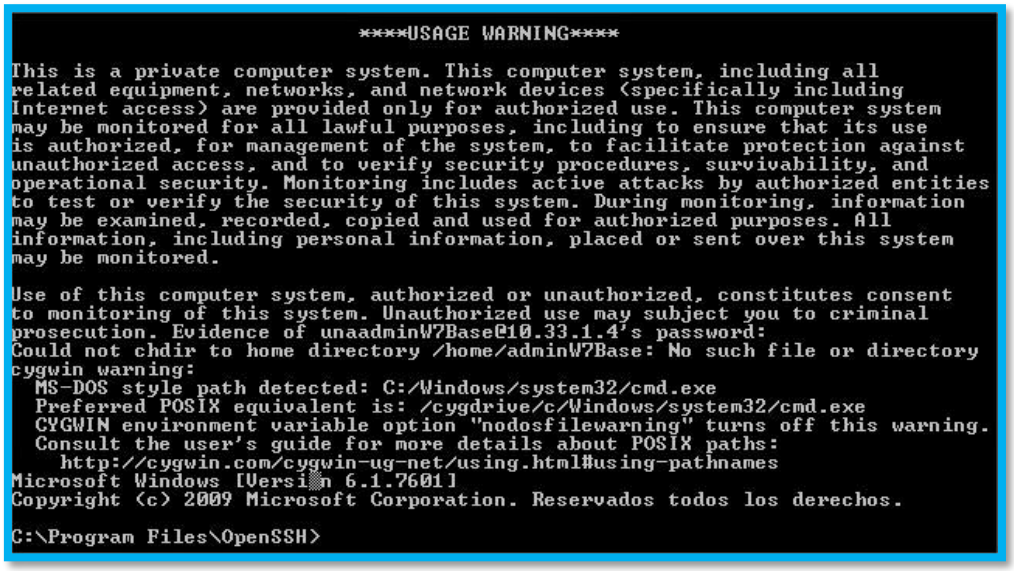


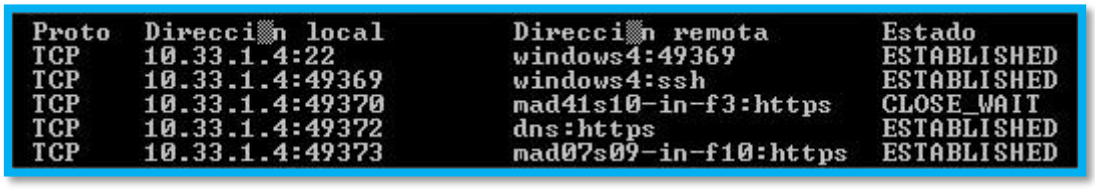
Seleccionamos adminW7Base con clic derecho y hacemos clic en restablecer contraseña (1234qwerty)

➢Conectarse al equipo desde sí mismo **ssh**
nombre_usuario_@192.168.50.1
En este caso utilizaremos el nombre del equipo y su dirección IP (ssh adminW7Base@10.33.1.4)



Comprobamos que funciona con éxito

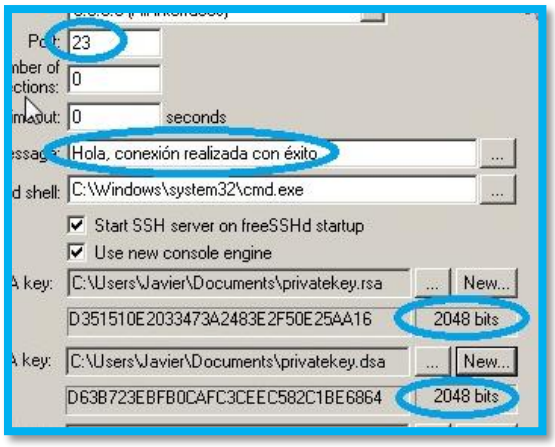




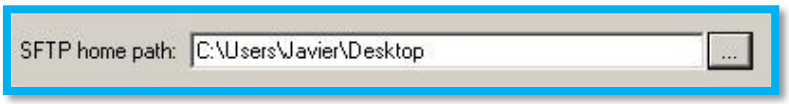
9. Instalar FreeSSH en Windows (Windows 4)

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

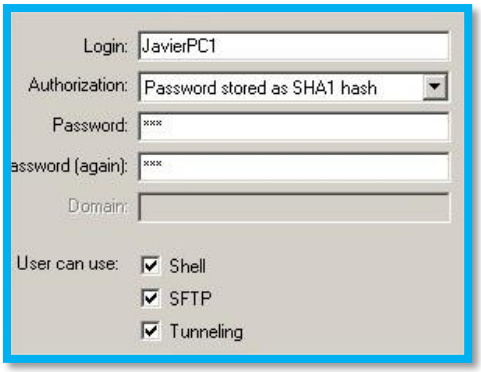
Se utilizará el puerto 23, ya que el puerto 22 nos da un error, tendrá un mensaje de bienvenida y se utilizará el cifrado más seguro (2048b)



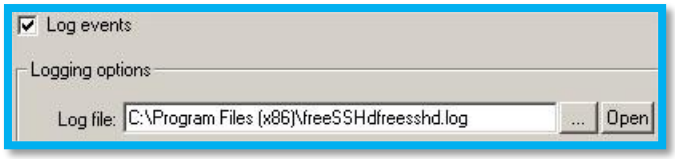
Se utilizará como ruta para el cliente, el escritorio



Crearemos el usuario "*JavierPC1*" con contraseña "123"



Activaremos el log para almacenar información sobre el SSH



Finalmente, activaremos el servidor SSH:

- **Instalamos Putty en w10-2 y lo configuramos.**

Teniendo el servidor SSH activo en la máquina (Windows 7-1) accederemos a la máquina cliente, (Windows 7 – 2)

En host name y port, colocaremos la dirección IP del servidor SSH y su puerto.

Una vez introduzcamos la información y le demos a "*ok*", nos aparecerá un aviso de seguridad, hacemos click en "*sí*"

A pesar de seguir los pasos nos aparece un error que al parecer no tiene solución en Windows 7, en el caso de utilizar un servidor Linux podría accederse sin problema.
Dicho error sucede al introducir el usuario, este nos muestra "*acceso denegado*" no tiene solución al menos, en Windows 7. (Se ha probado distintas soluciones, como desactivar "attempt GSAPI" que a varios usuarios ha podido solucionar o crear otros usuarios, sin solución)

Sin embargo, podemos comprobar que "funciona" ya que el servidor SSH es capaz de detectar el usuario, aunque este no se conecte.