Universidad Francisco de Vitoria
Centro de Documentación Europea
UFV Madrid

Universidad Francisco de Vitoria
Centro de Documentación Europea
UFV Madrid

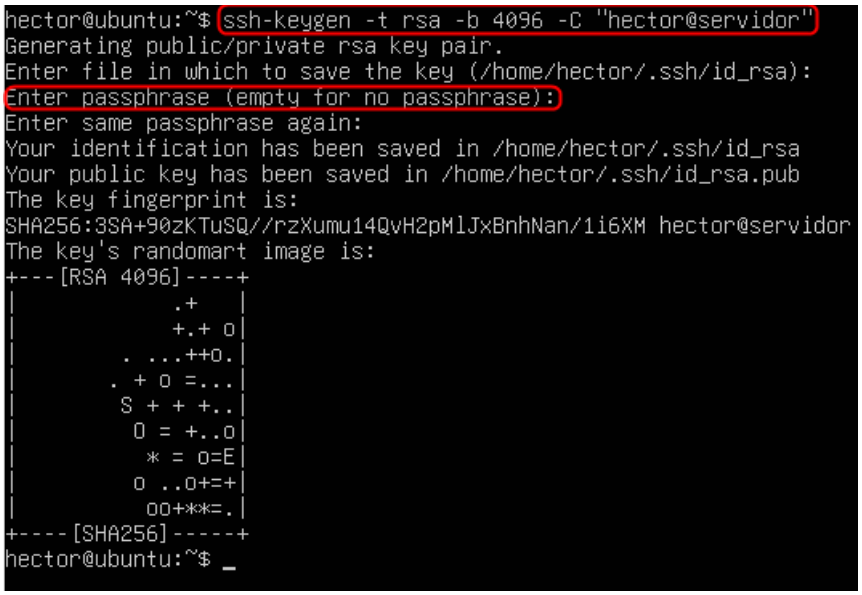
Héctor Ramírez López

Índice

Contenido

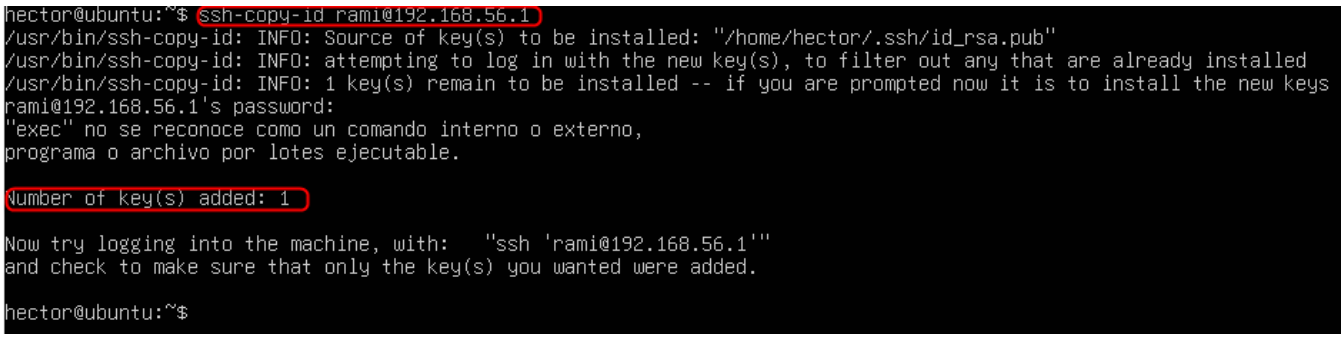
1. Generación de claves

Generamos la clave con la herramienta ssh-keygen, crearemos un par de claves RSA de 4096 bits protegida por una contraseña. Existen varios motivos para usar claves de mas de 2048 bits: tenemos mayor seguridad frente ataques modernos, longevidad en la protección y cumple los estándares de seguridad.



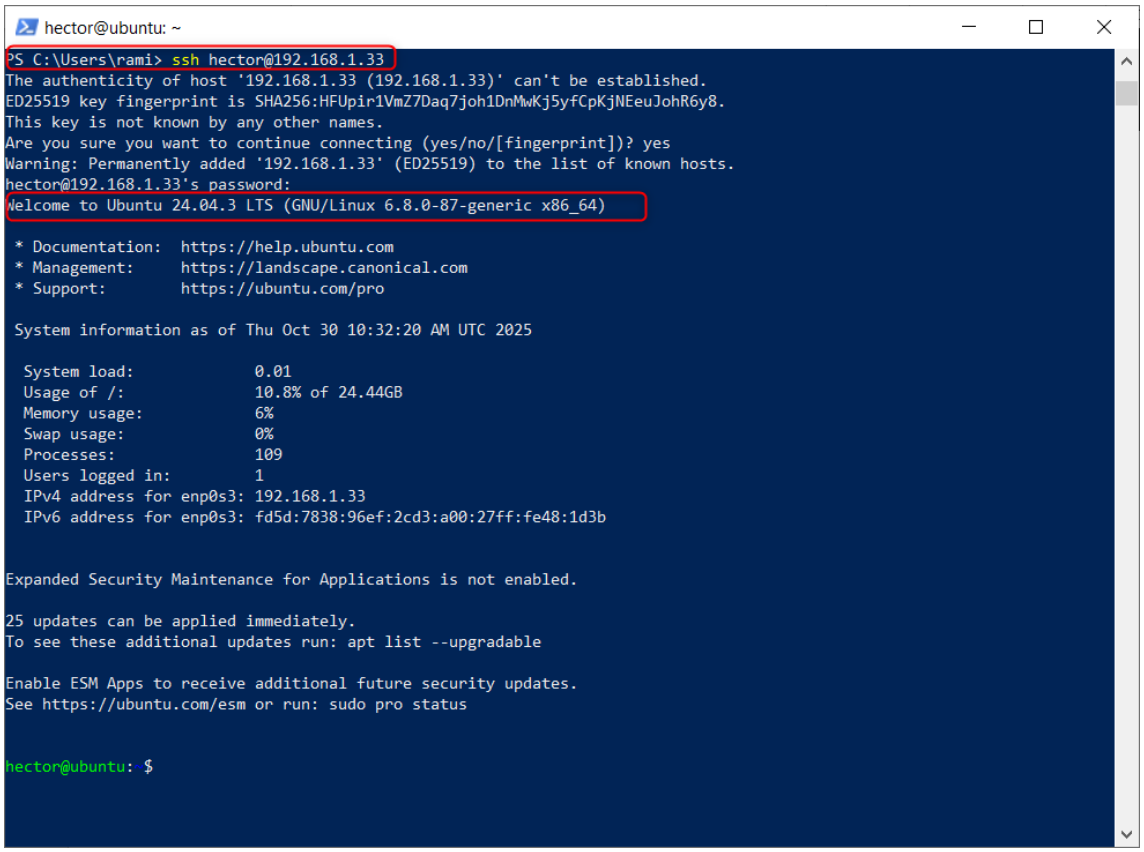
2. Transferencia de la clave pública:

Como podemos ver en esta imagen realizamos la transferencia de la clave publica al equipo cliente con el comando ssh-copy-id rami@192.168.56.1.



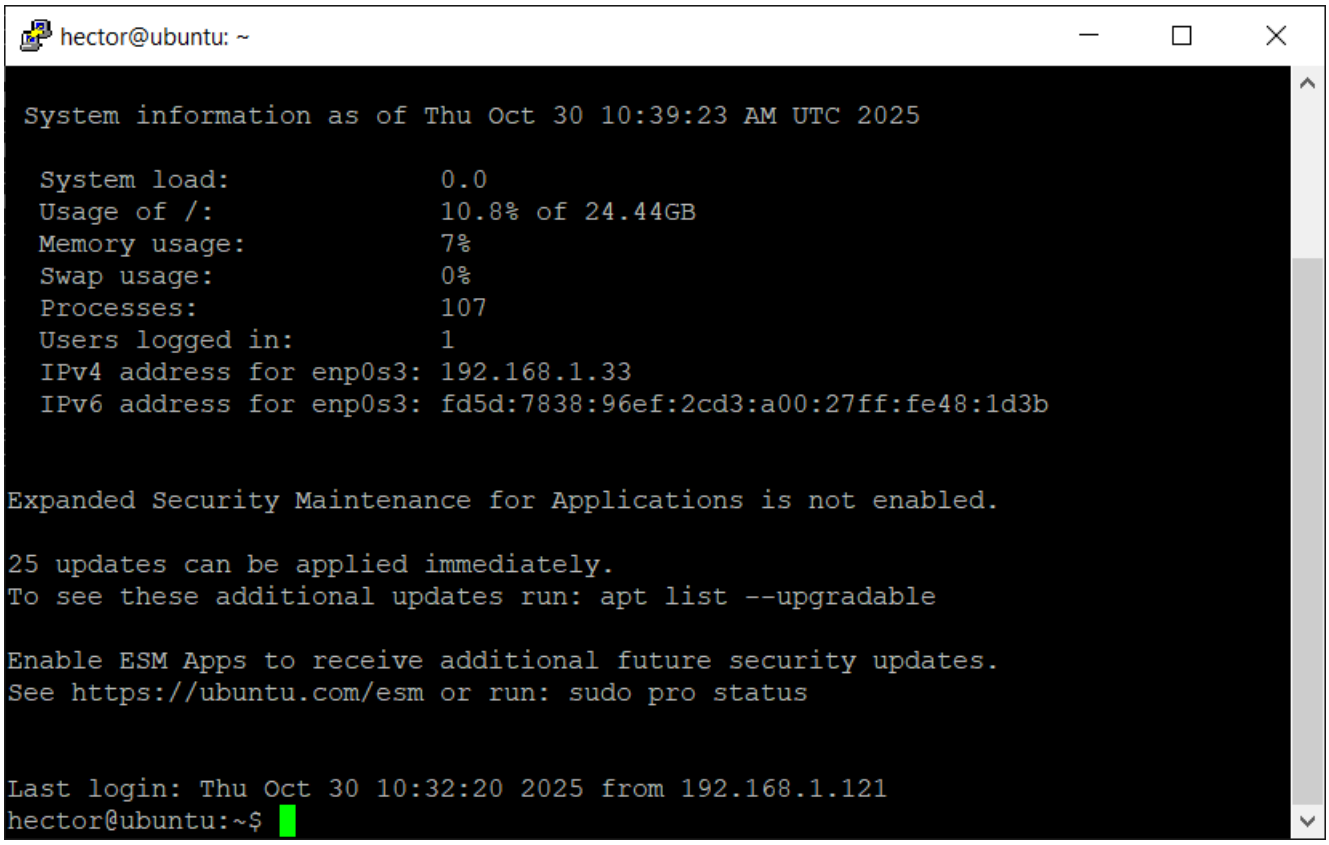
Ahora verificamos la conexión de Linux a Windows con el comando ssh rami@ipdewindows le damos a intro y como podemos ver se establece la conexión ssh.

Ahora probaremos la conexión desde nuestro Windows a nuestro servidor Ubuntu

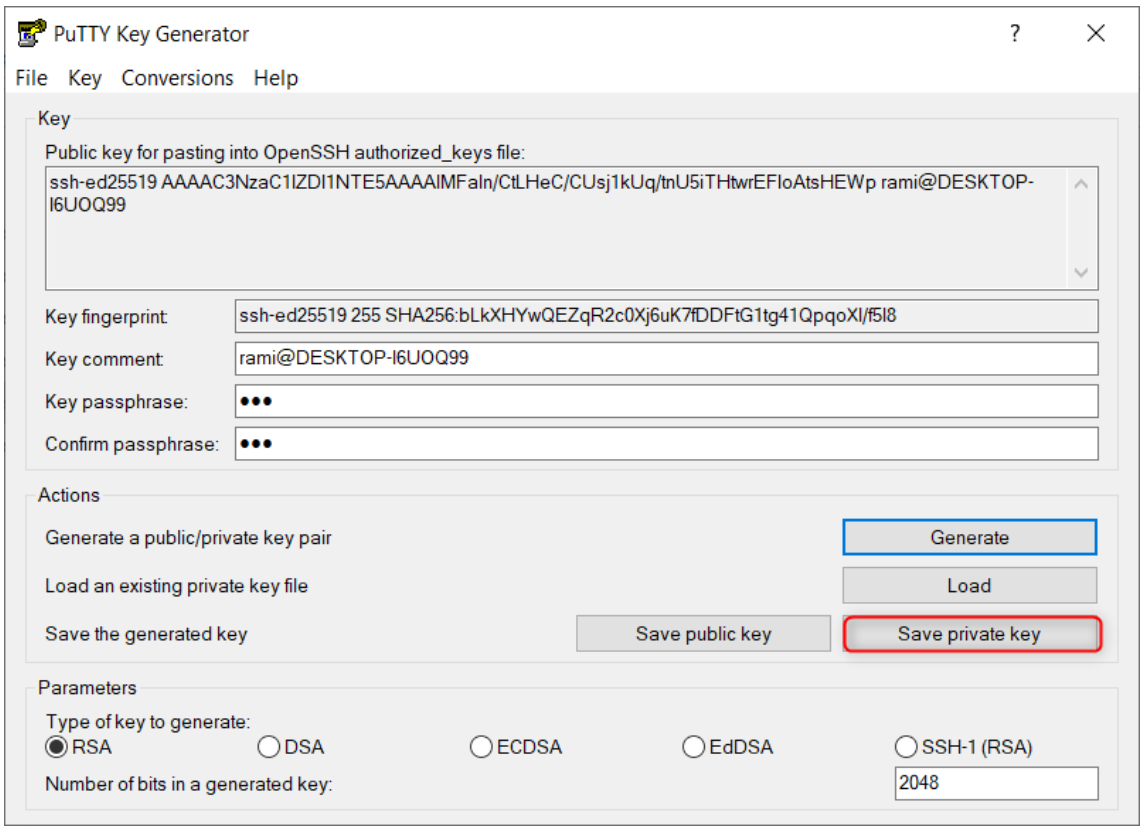


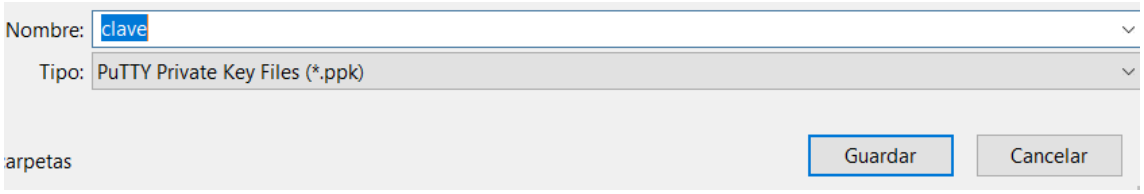
3. Integración con Putty (Windows):

Probamos a realizar la conexión con putty y nos conectamos al servidor

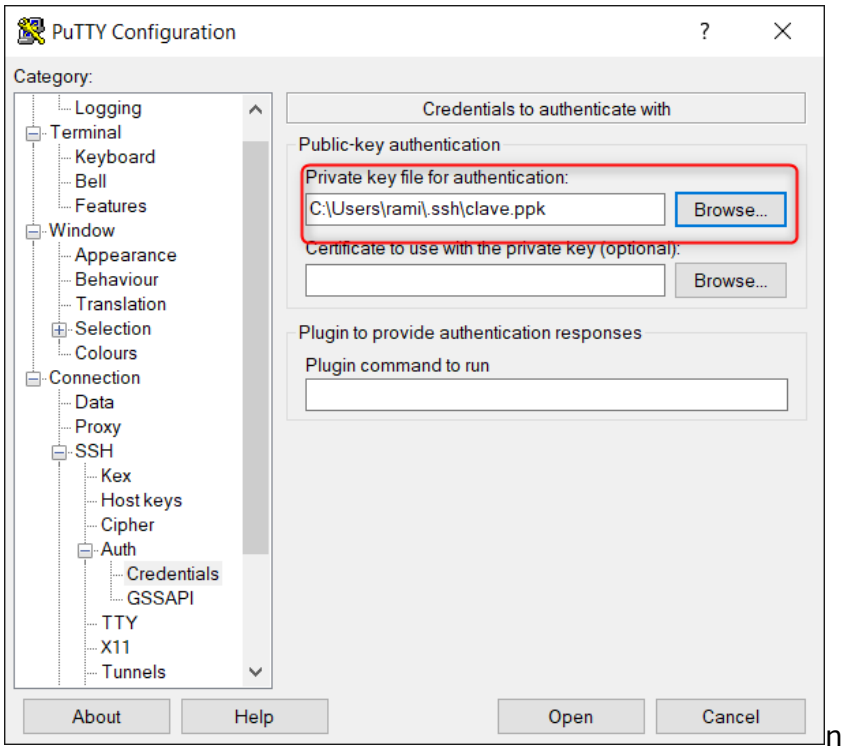


Ahora importamos la clave privada en puttigen y guardamos la clave y se nos guardara en formato.pkk

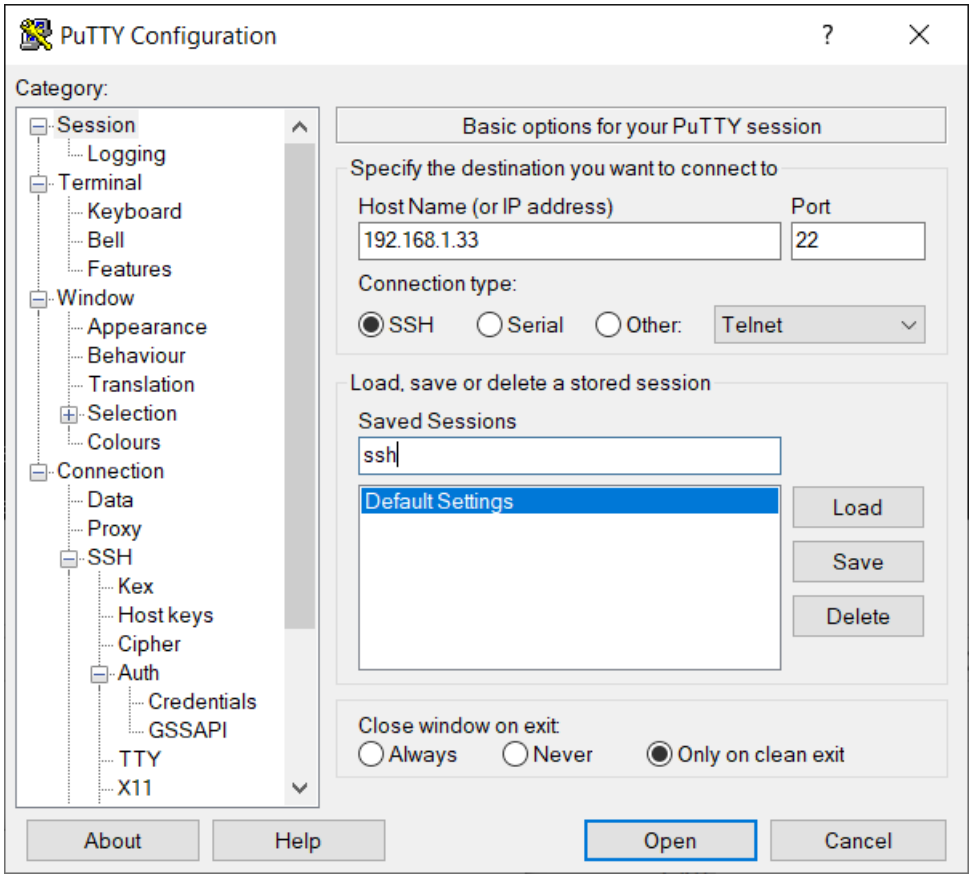




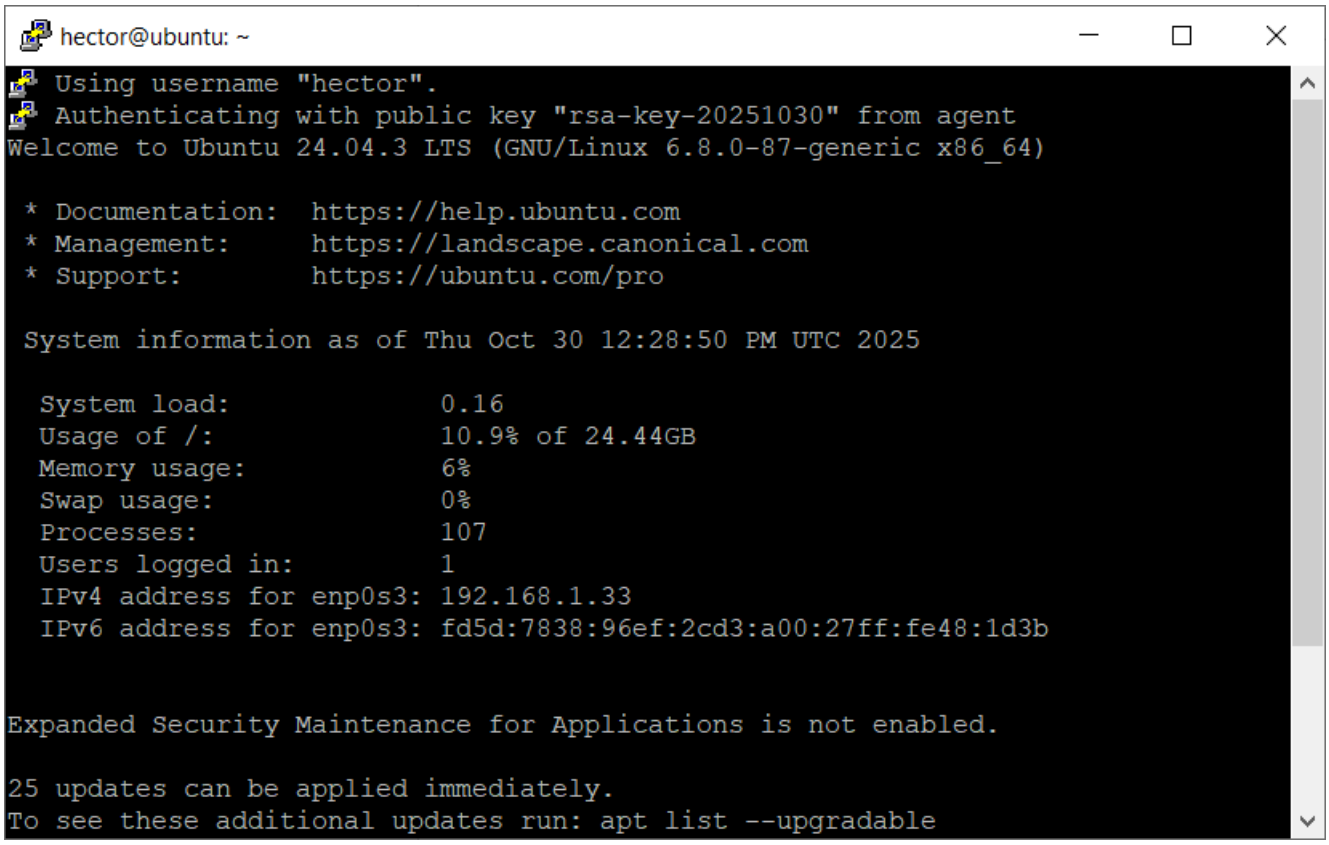
A continuación, configuramos una nueva sesión SSH en putty, integrando la clave privada y guardando la sesió



n

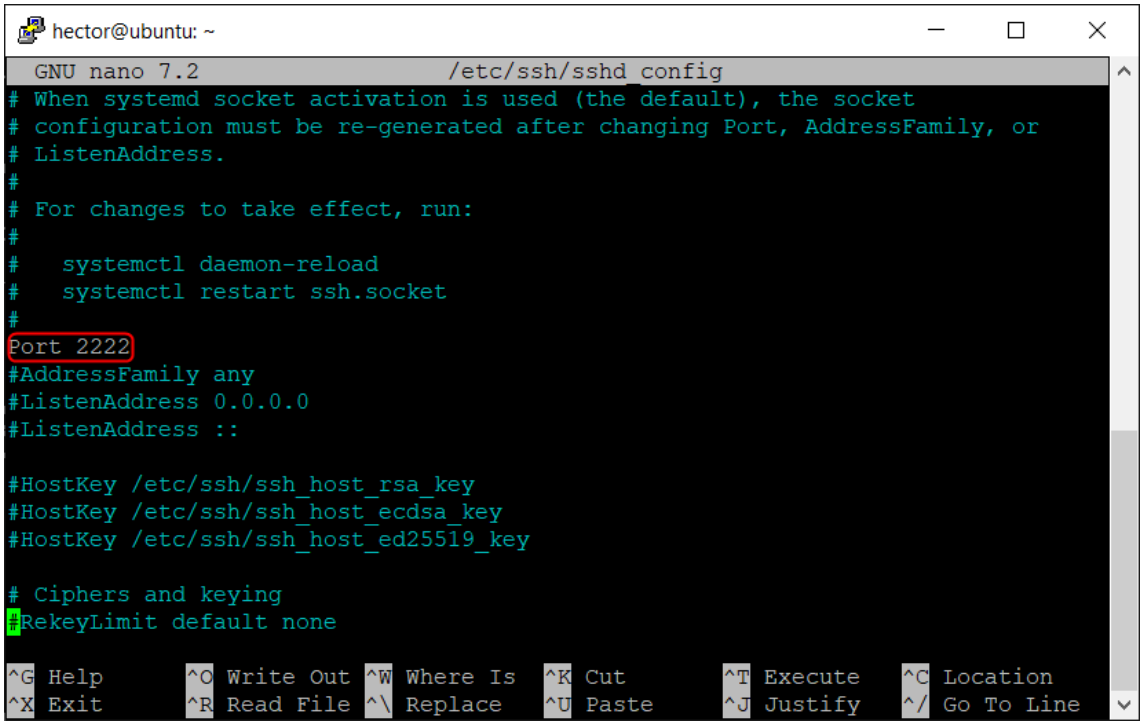

PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
SSH
Kex
Host keys
Cipher
Auth
Credentials
GSSAPI
TTY
X11
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
192.168.1.33
22
Connection type:
SSH
Serial
Other:
Telnet
Load, save or delete a stored session
Saved Sessions
ssh
Default Settings
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Help
Open
Cancel

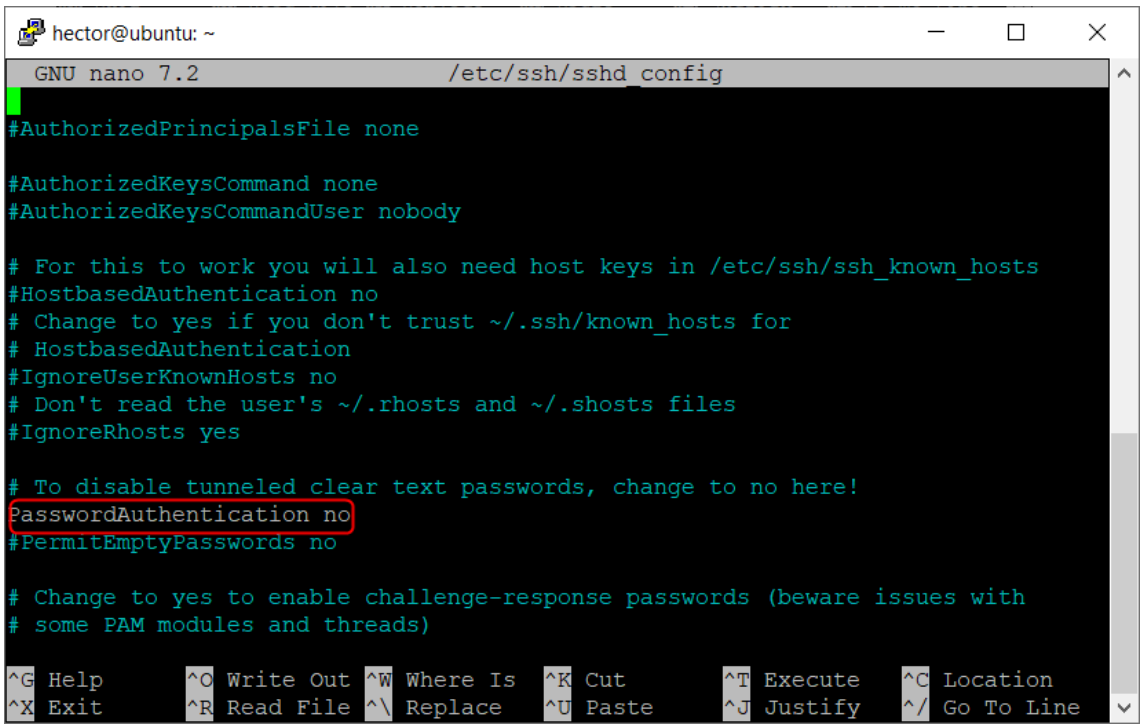
Como podemos ver podemos iniciar sesión sin la contraseña, si hubiéramos exportado las claves con frase de paso nos la pediría igualmente.



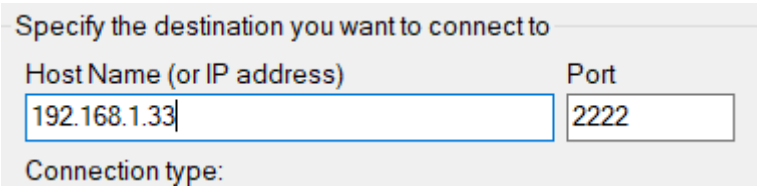
4. Bastionado adicional:

Editamos el archivo /etc/ssh/sshd_config con el comando sudo nano /etc/ssh/sshd_config y cambiamos el puerto al 2222 y desactivamos el acceso por contraseña.

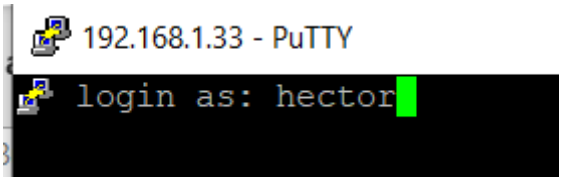


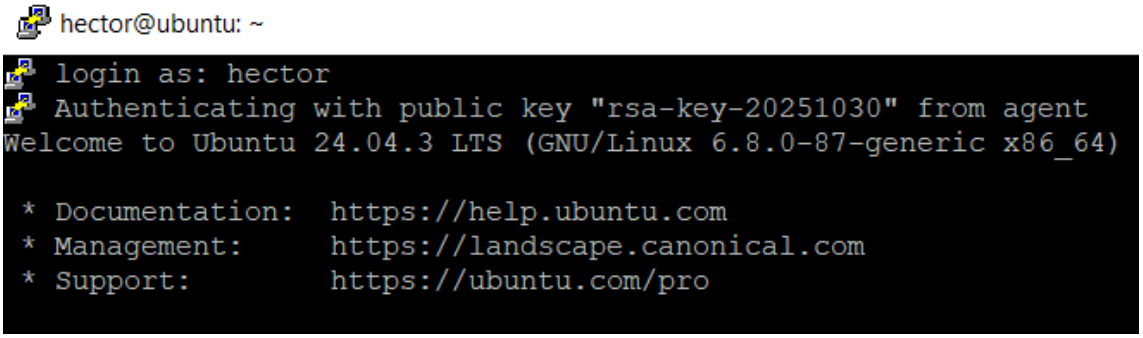


Después de reiniciar el servicio ssh tendremos que acceder por el puerto 2222



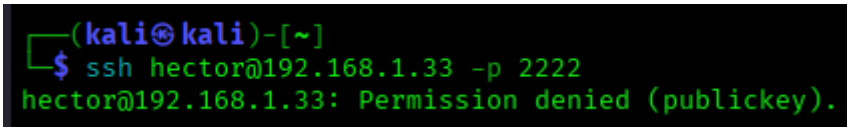
Nos pedirá con que usuario queremos logear y como nuestra clave está autorizada nos dejará acceder



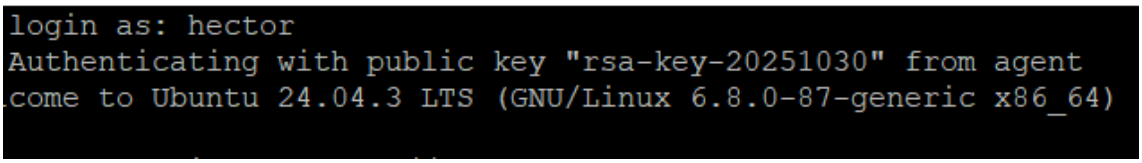


5. Verificación:

Al conectarnos por ssh desde otro cliente como hemos deshabilitado el acceso por password no nos dejara, solo nos dejaría con la clave



Pero si tuviéramos la clave autorizada si nos dejaría acceder sin introducir password



6. (Opcional Avanzado):

Procedemos a instalar Google authenticator para mejorar la seguridad

```
hector@ubuntu:~$ sudo apt-get install google-authenticator
```

Una vez instalado con el comando google-authenticator generaremos los tokens

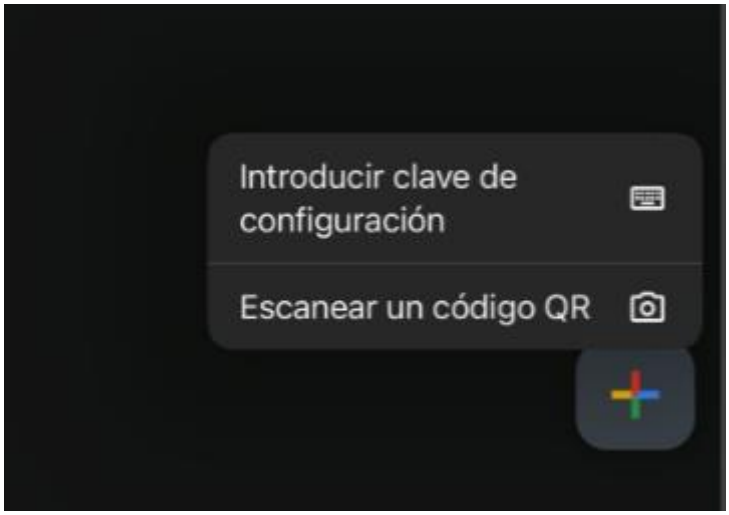
```
hector@ubuntu:~$ google-authenticator

Do you want authentication tokens to be time-based (y/n)
```

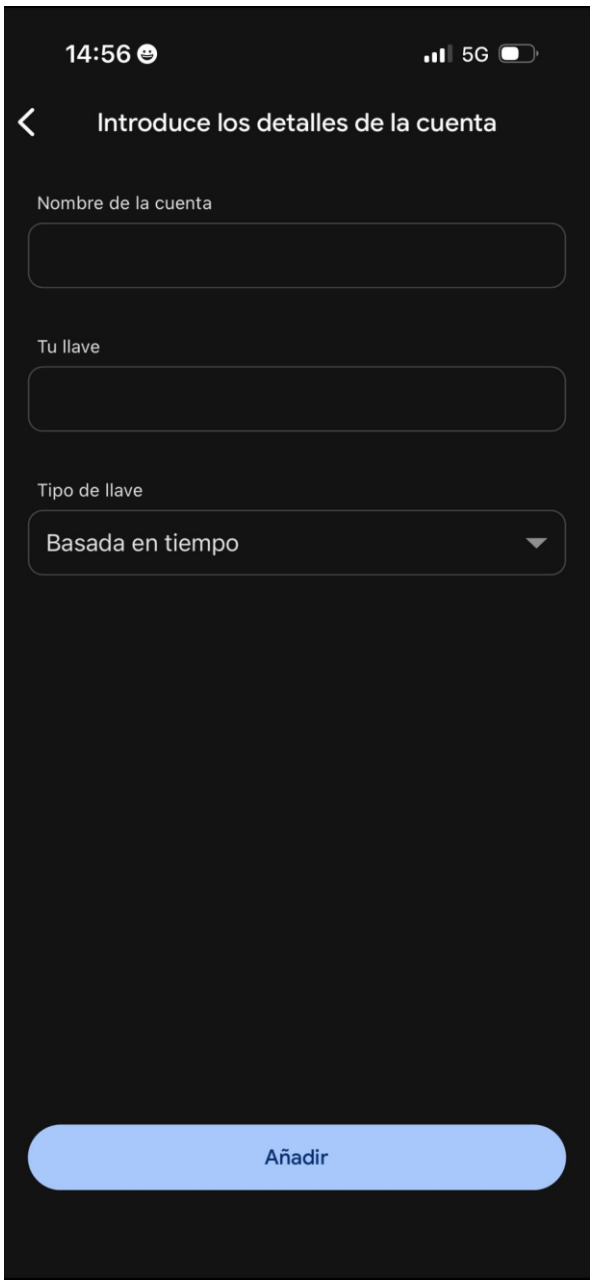
Una vez generado nos dará el código para ponerlo en el autheticator

```
Your new secret key is: SWIUPO2TAL3KKWDHVATMWY7TKQ
Enter code from app (-1 to skip):
```

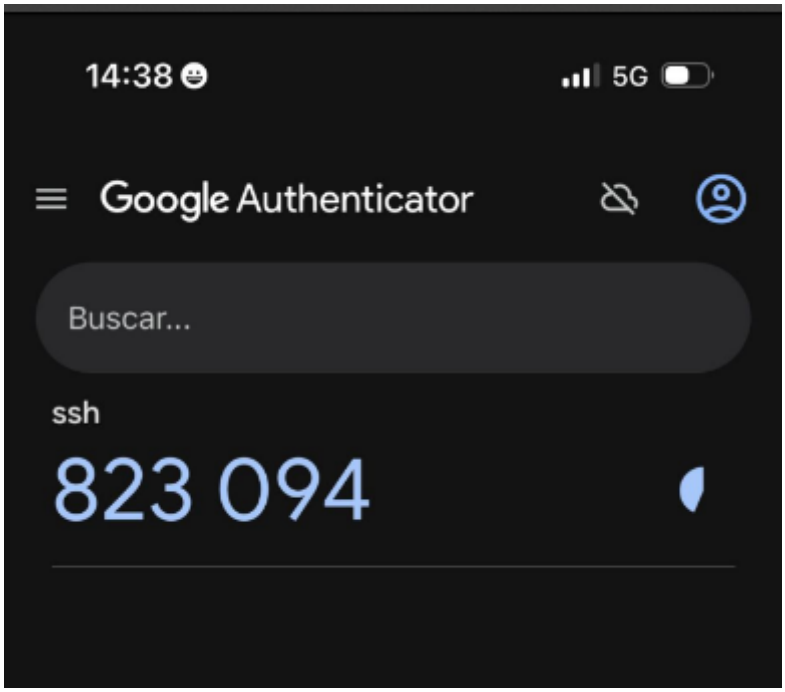
Introduciremos el código en la app de Google authenticator



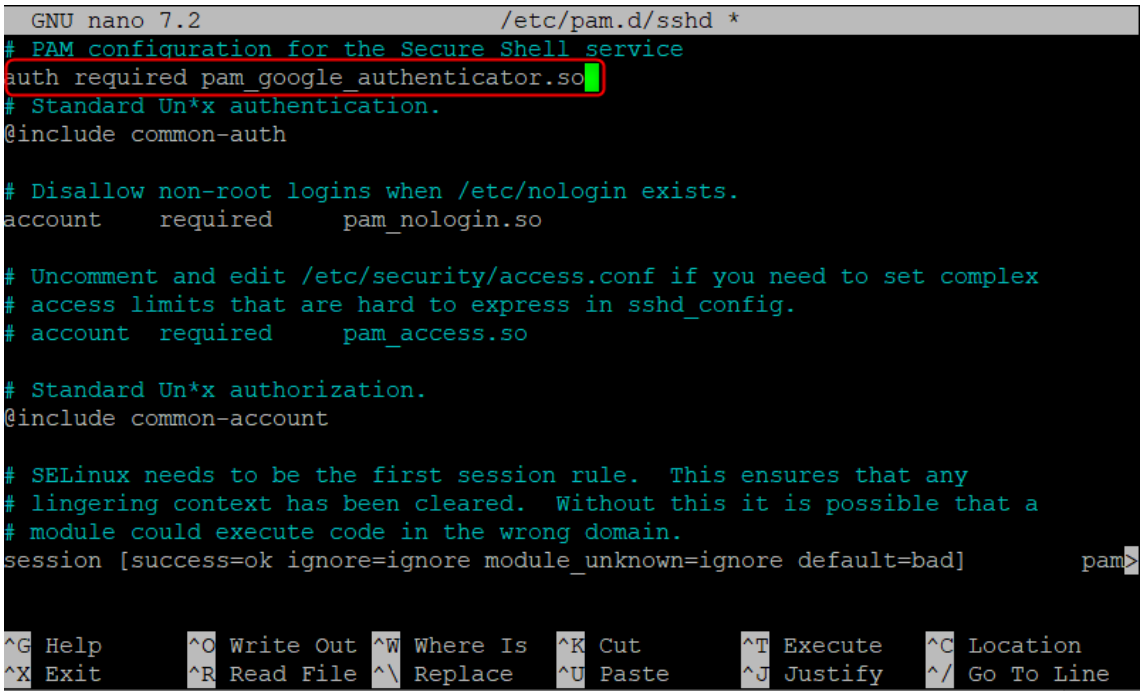
Pondremos el nombre de la cuenta y el código y le daremos añadir



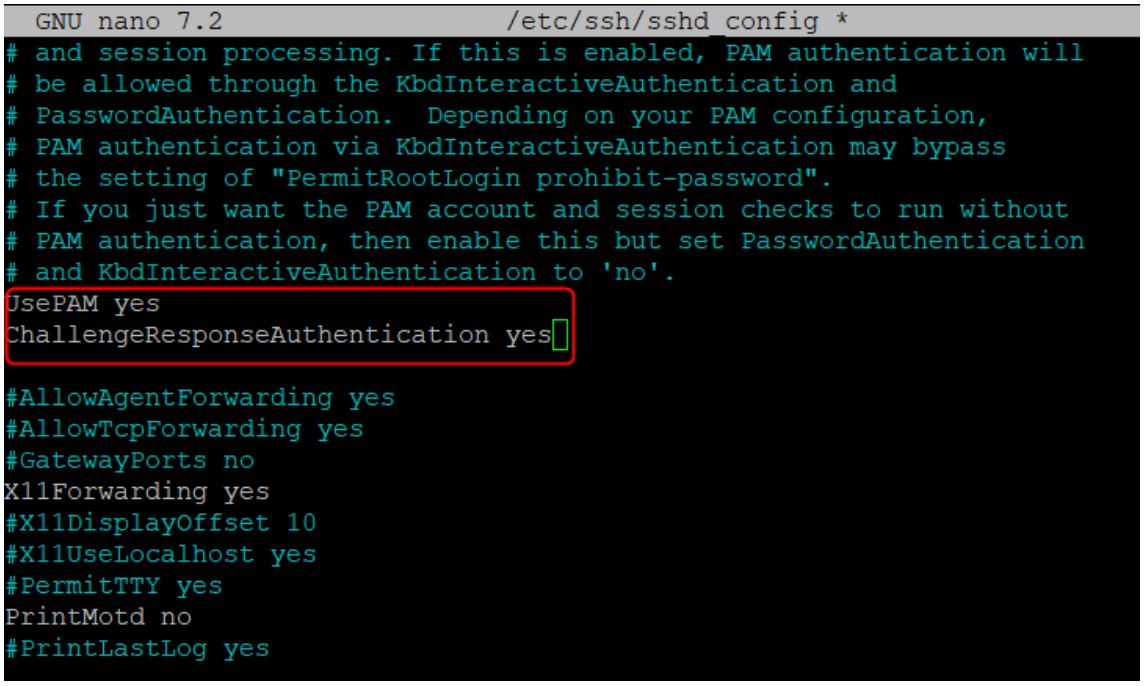
Una vez añadido ya tendremos configurado el acceso con Google authenticator

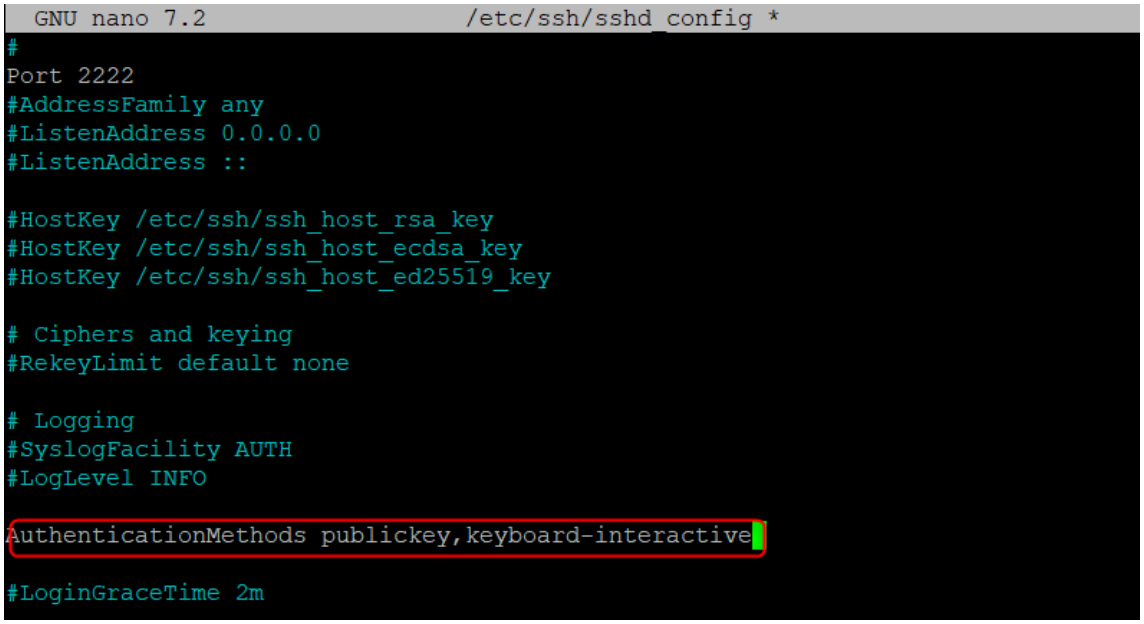


Ahora tendremos que configurar el archivo pam de ssh para decirle que use el Google authenticator como proceso de verificación para ello editaremos el archivo con el comando sudo nano /etc/pam.d/sshd y añadiremos la siguiente línea al principio



El siguiente paso es modificar la configuración del servicio SSH para que soporte el 2FA. Para ello, editaremos el archivo de configuración de OpenSSH, con el siguiente comando sudo nano /etc/ssh/sshd_config, habría que añadir las tres líneas de las imagenes





Con esta configuración, ssh pedirá primero la autenticación por clave pública y después el código de Google Authenticator, ahora solo falta reiniciar el servicio ssh para que se apliquen los cambios

Una vez reiniciado cuando nos conectemos por ssh nos pedirá la clave que este autorizada y el Código de la app Google authenticator

Universidad
Francisco de
Vitoria
Centro de
Documentación
Europea
UFV Madrid