



Universidad
Francisco de
Vitoria
*Centro de
Documentación
Europea*
UFV Madrid

Héctor Ramírez López

Índice

Contenido

1.	Generación de claves	3
2.	Transferencia de la clave pública:.....	4
3.	Integración con Putty (Windows):	6
4.	Bastionado adicional:	11
5.	Verificación:.....	13
6.	(Opcional Avanzado):	14

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.

```
hector@ubuntu:~$ ssh-keygen -t rsa -b 4096 -C "hector@servidor"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hector/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hector/.ssh/id_rsa
Your public key has been saved in /home/hector/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:3SA+90zKTuSQ//rzXumu14QvH2pM1JxBnhNan/1i6XM hector@servidor
The key's randomart image is:
----[RSA 4096]----+
| .+
| +.+ 0|
| . ...++0.|
| . + 0 =...|
| S + + +..|
| 0 = +..o|
| * = 0=E|
| 0 ..0+=+|
| 00+**=.|
-----[SHA256]-----+
hector@ubuntu:~$ _
```

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.

```
hector@ubuntu:~$ ssh-copy-id rami@192.168.56.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/hector/.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 prompted now it is to install the new keys
rami@192.168.56.1's password:
"exec" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'rami@192.168.56.1'"
and check to make sure that only the key(s) you wanted were added.

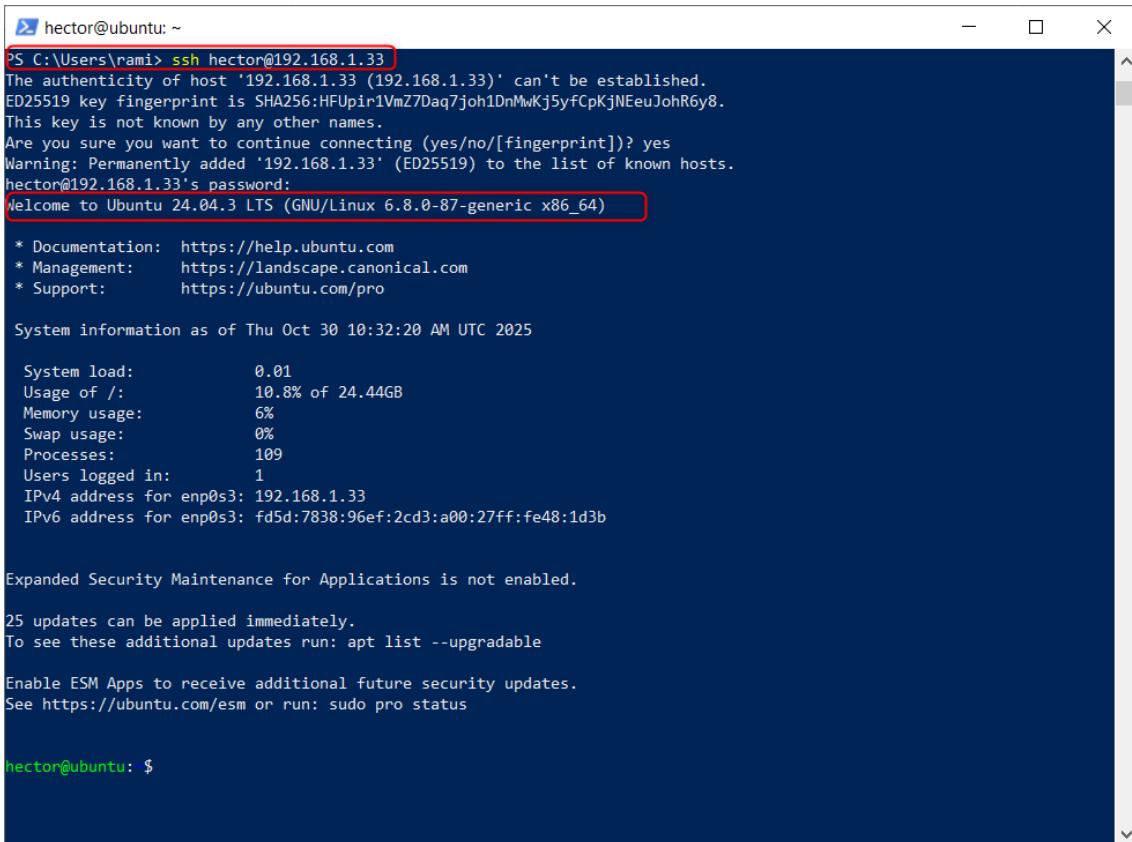
hector@ubuntu:~$
```

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.

```
Microsoft Windows [Versión 10.0.19045.6456]
(c) Microsoft Corporation. Todos los derechos reservados.

rami@DESKTOP-I6U0Q99 C:\Users\rami>
```

Ahora probaremos la conexión desde nuestro Windows a nuestro servidor Ubuntu



```
PS C:\Users\rami> ssh hector@192.168.1.33
The authenticity of host '192.168.1.33 (192.168.1.33)' can't be established.
ED25519 key fingerprint is SHA256:HFUpir1VmZ7Daq7joh1DnMwKj5yfCpKjNEeuJohR6y8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.33' (ED25519) to the list of known hosts.
hector@192.168.1.33's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-87-generic x86_64)

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

System information as of Thu Oct 30 10:32:20 AM UTC 2025

System load:          0.01
Usage of /:           10.8% of 24.44GB
Memory usage:         6%
Swap usage:          0%
Processes:            109
Users logged in:     1
IPv4 address for enp0s3: 192.168.1.33
IPv6 address for enp0s3: fd5d:7838:96ef:2cd3:a00:27ff:fe48:1d3b

Expanded Security Maintenance for Applications is not enabled.

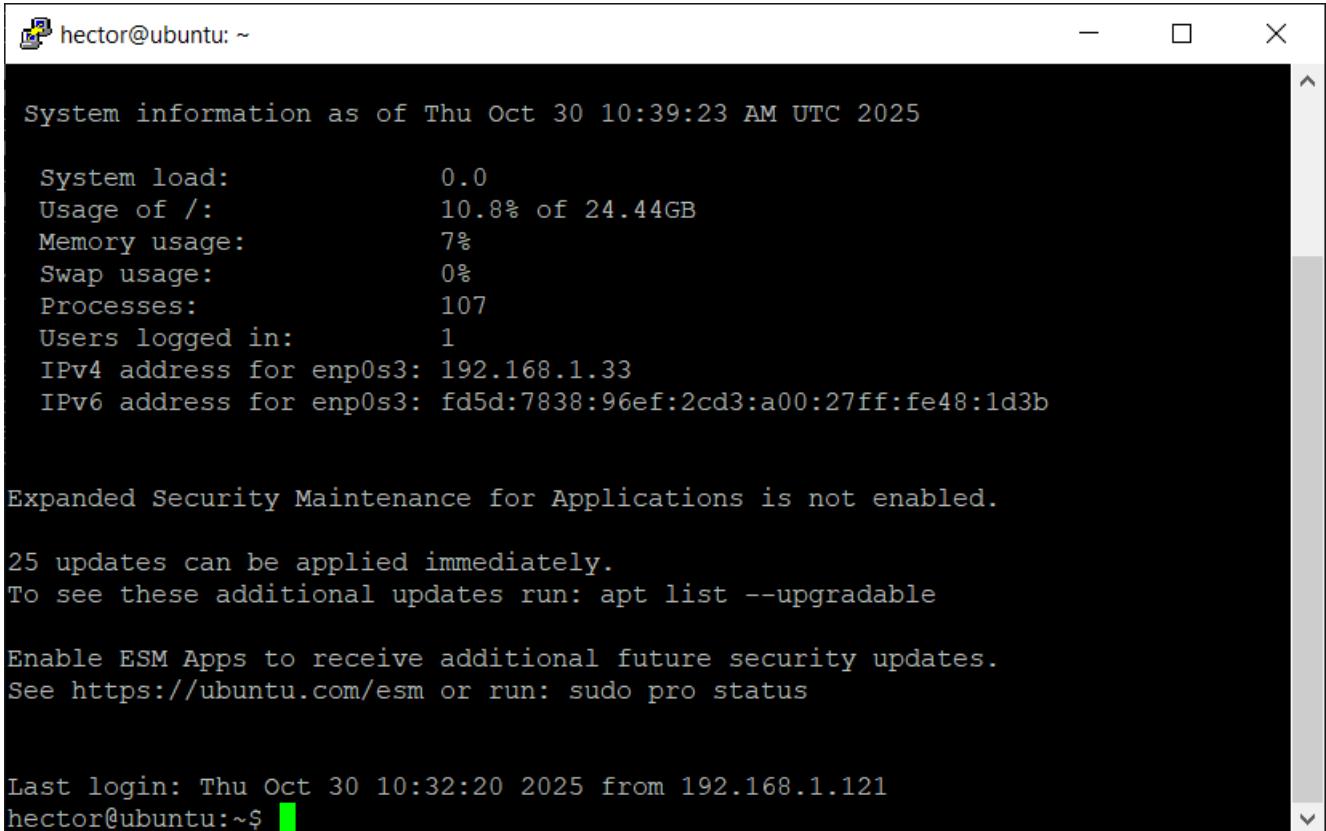
25 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

hector@ubuntu: $
```

3. Integración con Putty (Windows):

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



```
hector@ubuntu: ~
System information as of Thu Oct 30 10:39:23 AM UTC 2025
System load:          0.0
Usage of /:           10.8% of 24.44GB
Memory usage:         7%
Swap usage:           0%
Processes:            107
Users logged in:     1
IPv4 address for enp0s3: 192.168.1.33
IPv6 address for enp0s3: fd5d:7838:96ef:2cd3:a00:27ff:fe48:1d3b

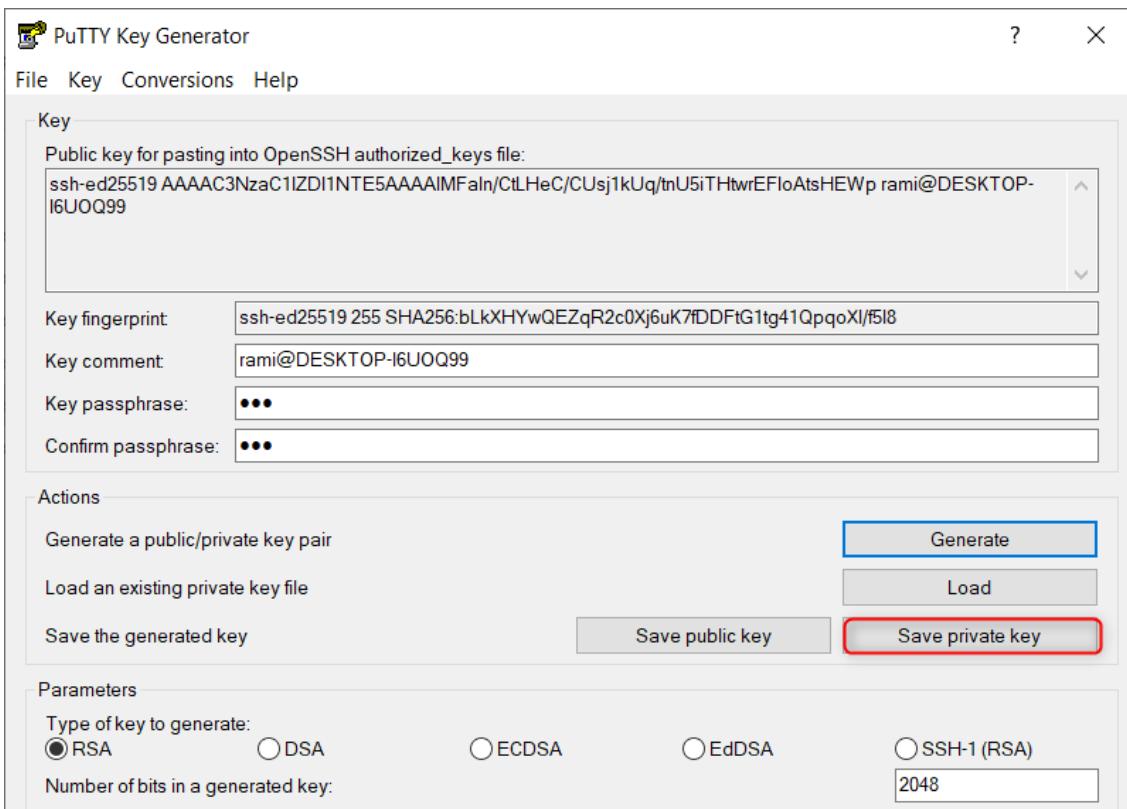
Expanded Security Maintenance for Applications is not enabled.

25 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Oct 30 10:32:20 2025 from 192.168.1.121
hector@ubuntu:~$
```

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



The PuTTY Key Generator window is open. In the 'Key' section, a public key for OpenSSH is displayed:

```
ssh-ed25519 AAAAC3NzaC1IzD1NTE5AAAIIMFain/CtLHeC/CUsj1kUq/thU5iTHTwrEFloAtsHEWp rami@DESKTOP-I6UOQ99
```

Below the key, there are fields for 'Key fingerprint' (ssh-ed25519 255 SHA256:bLkXHYwQEZqR2c0Xj6uK7fDDFtG1tg41QpqoXI/f5l8), 'Key comment' (rami@DESKTOP-I6UOQ99), 'Key passphrase' (three dots), and 'Confirm passphrase' (three dots). In the 'Actions' section, there are buttons for 'Generate' (blue), 'Load' (gray), 'Save public key' (gray), and 'Save private key' (red border). In the 'Parameters' section, 'RSA' is selected as the type of key to generate, and the number of bits is set to 2048.

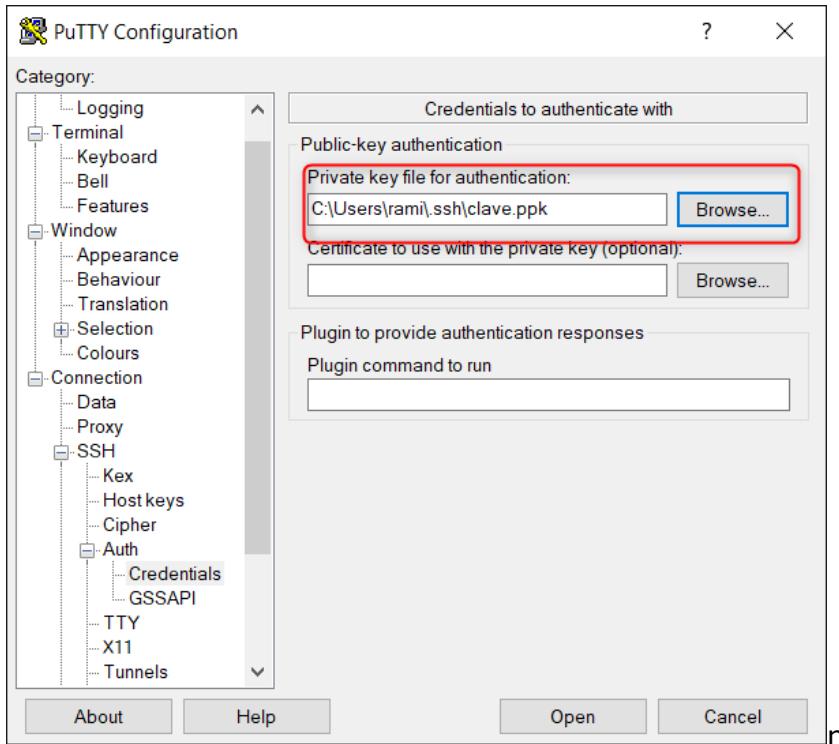
Below the generator window, a save dialog box is shown:

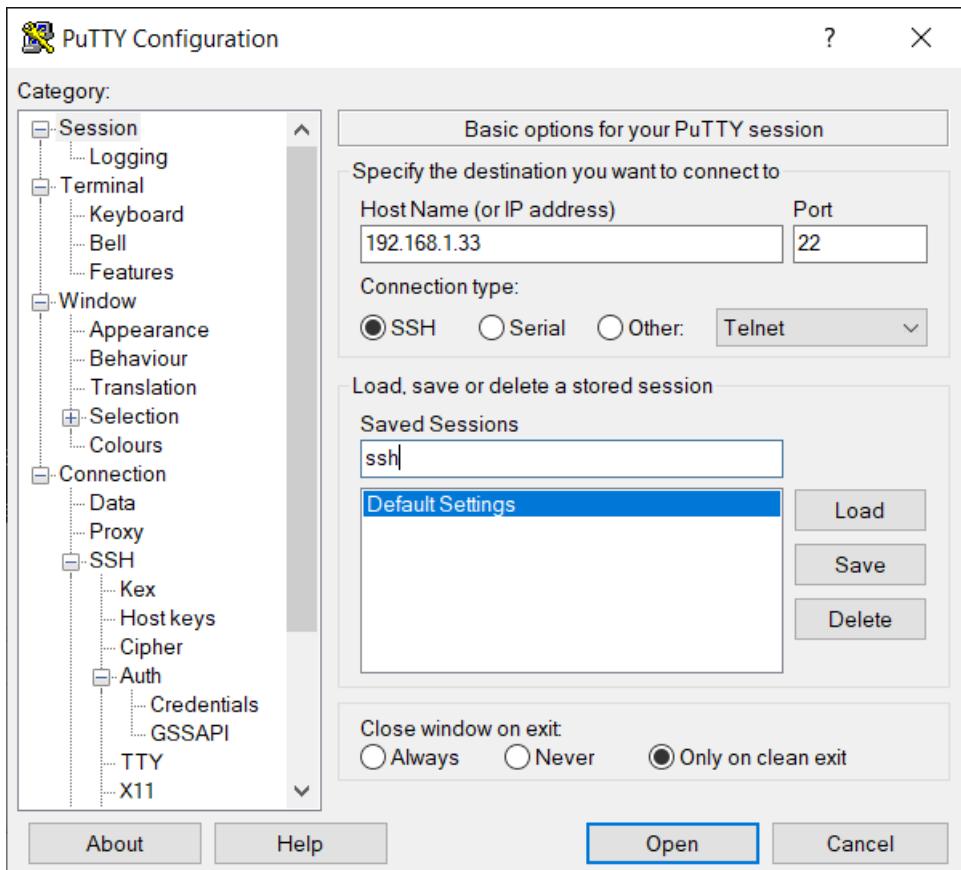
Nombre: clave

Tipo: PuTTY Private Key Files (*.ppk)

Guardar Cancelar

A continuación, configuraremos una nueva sesión SSH en putty, integrando la clave privada y guardando la sesión.





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

```
hector@ubuntu: ~
Using username "hector".
Authenticating with public key "rsa-key-20251030" from agent
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-87-generic x86_64)

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

System information as of Thu Oct 30 12:28:50 PM UTC 2025

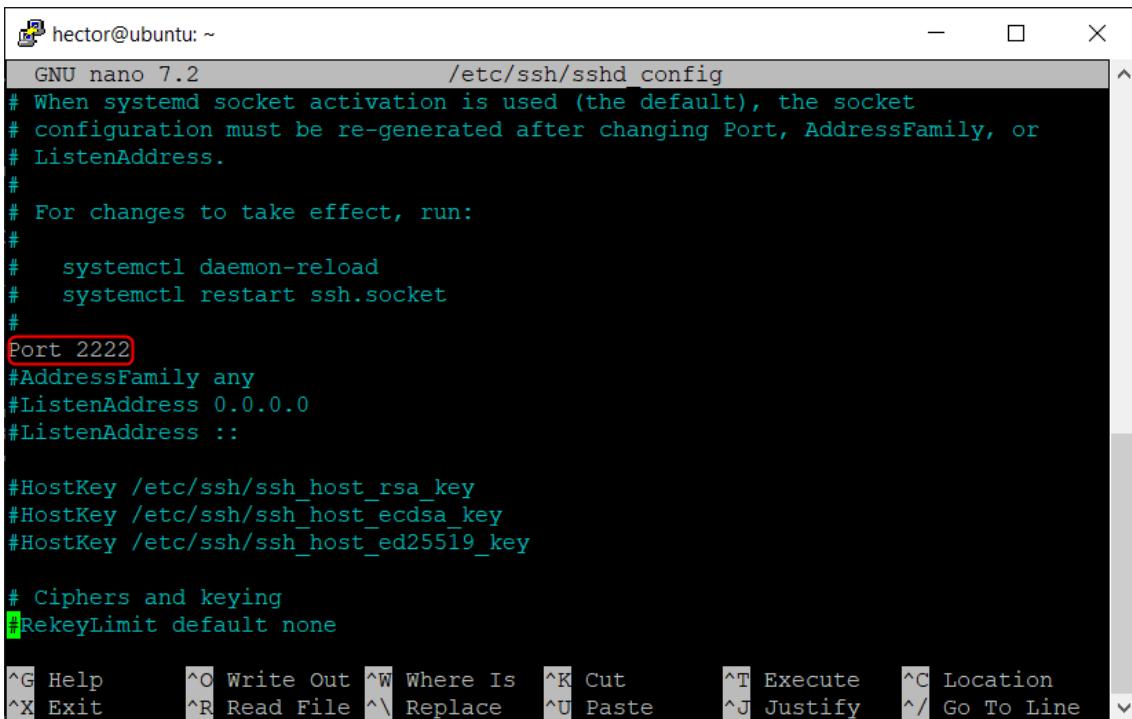
 System load:          0.16
 Usage of /:           10.9% of 24.44GB
 Memory usage:         6%
 Swap usage:           0%
 Processes:            107
 Users logged in:     1
 IPv4 address for enp0s3: 192.168.1.33
 IPv6 address for enp0s3: fd5d:7838:96ef:2cd3:a00:27ff:fe48:1d3b

Expanded Security Maintenance for Applications is not enabled.

25 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
```

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.

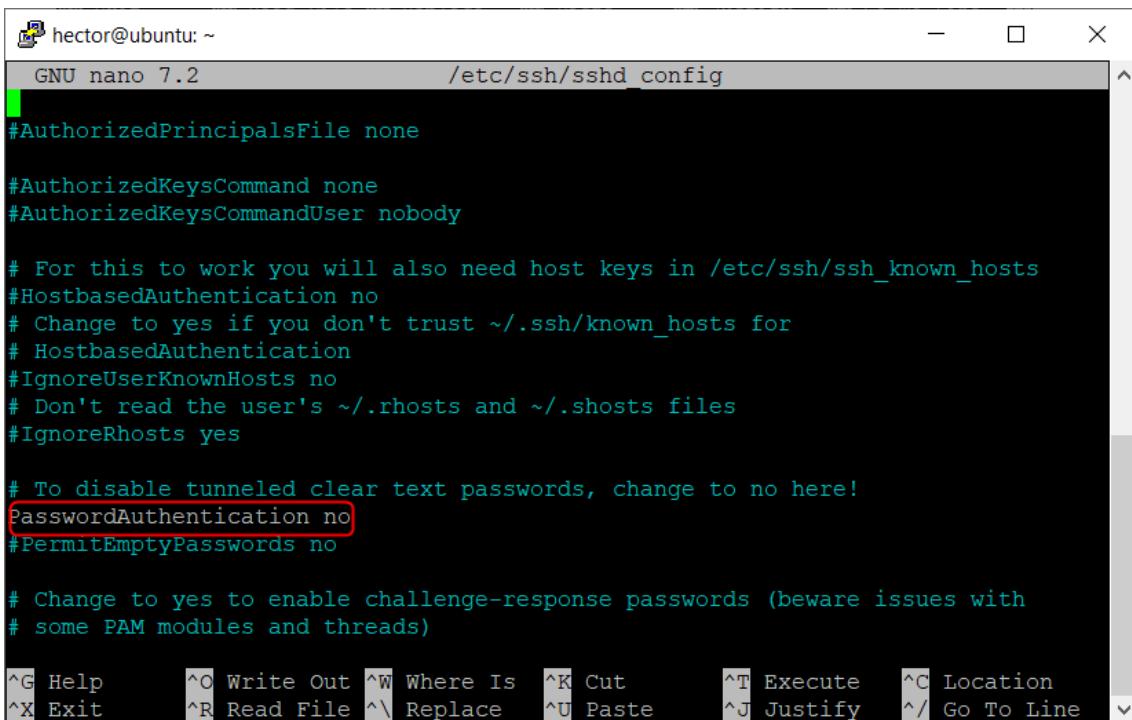


```
hector@ubuntu: ~
GNU nano 7.2          /etc/ssh/sshd config
# When systemd socket activation is used (the default), the socket
# configuration must be re-generated after changing Port, AddressFamily, or
# ListenAddress.
#
# For changes to take effect, run:
#
#   systemctl daemon-reload
#   systemctl restart ssh.socket
#
Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```



```
hector@ubuntu: ~
GNU nano 7.2          /etc/ssh/sshd config
#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

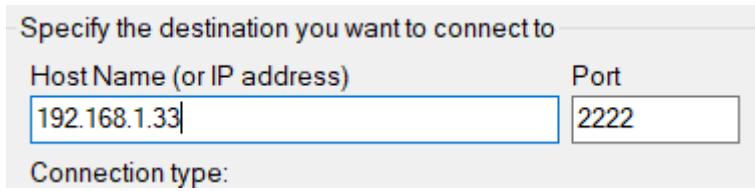
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

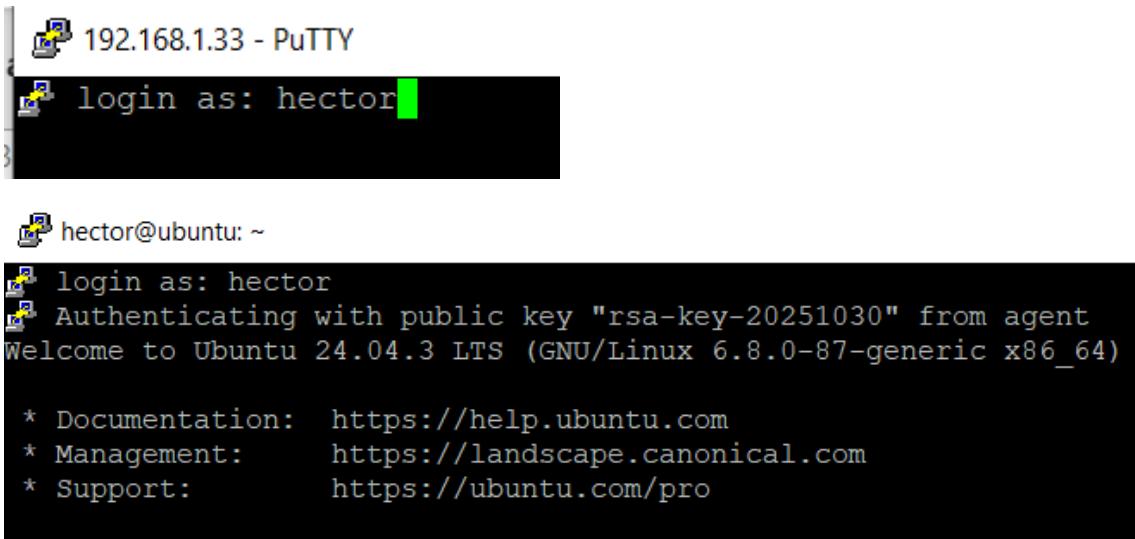
# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)

^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

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



```
192.168.1.33 - PuTTY
login as: hector

hector@ubuntu: ~
login as: hector
Authenticating with public key "rsa-key-20251030" from agent
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-87-generic x86_64)

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

5. Verificación:

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

```
(kali㉿kali)-[~]
$ ssh hector@192.168.1.33 -p 2222
hector@192.168.1.33: Permission denied (publickey).
```

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

```
login as: hector
Authenticating with public key "rsa-key-20251030" from agent
come to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-87-generic x86_64)
```

6. (Opcional Avanzado):

Procedemos a instalar Google authenticator para mejorar la seguridad

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

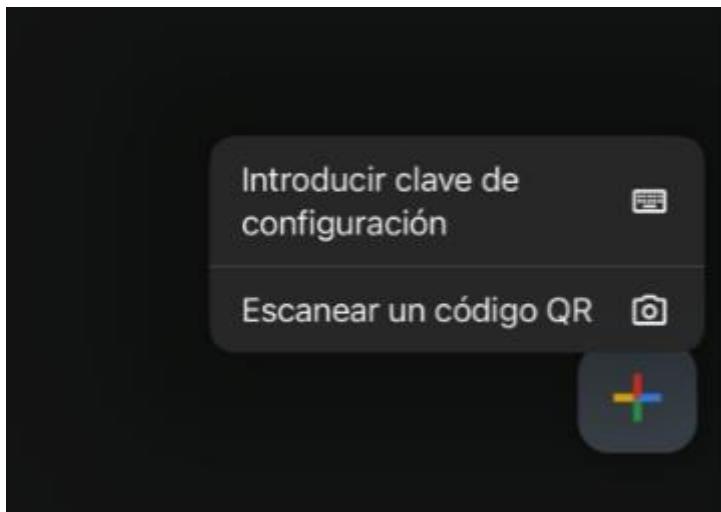
Una vez instalado con el comando google-authenticator generaremos los tokens

```
hector@ubuntu:~$ google-authenticator  
Do you want authentication tokens to be time-based (y/n) [
```

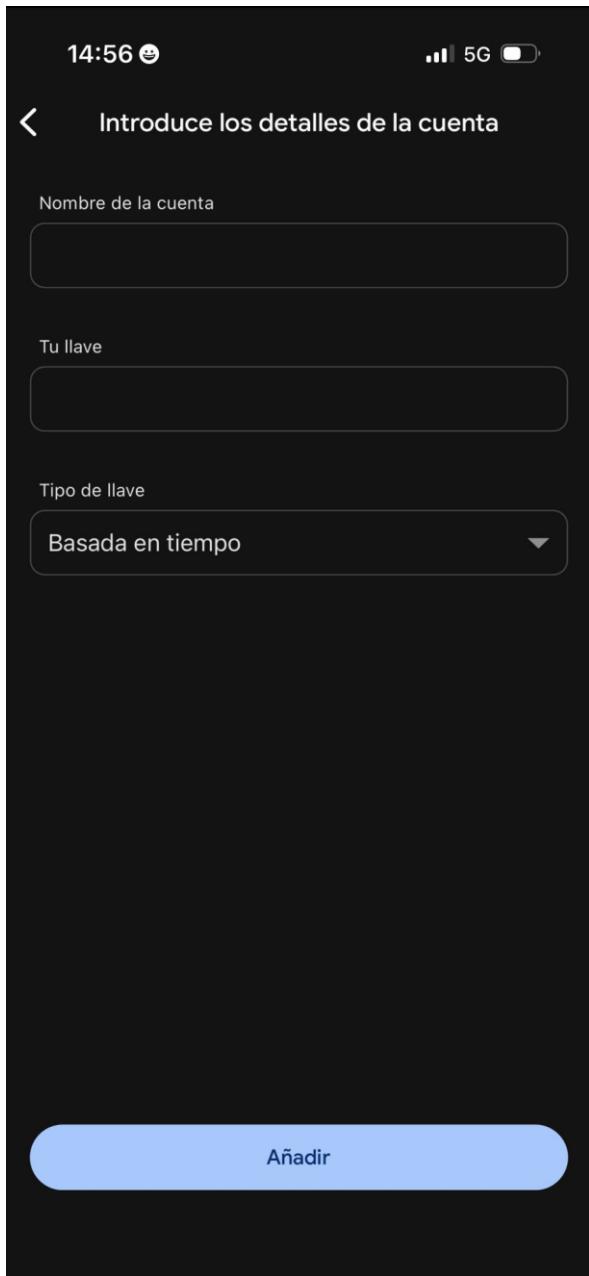
Una vez generado nos dará el código para ponerlo en el authenticator

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

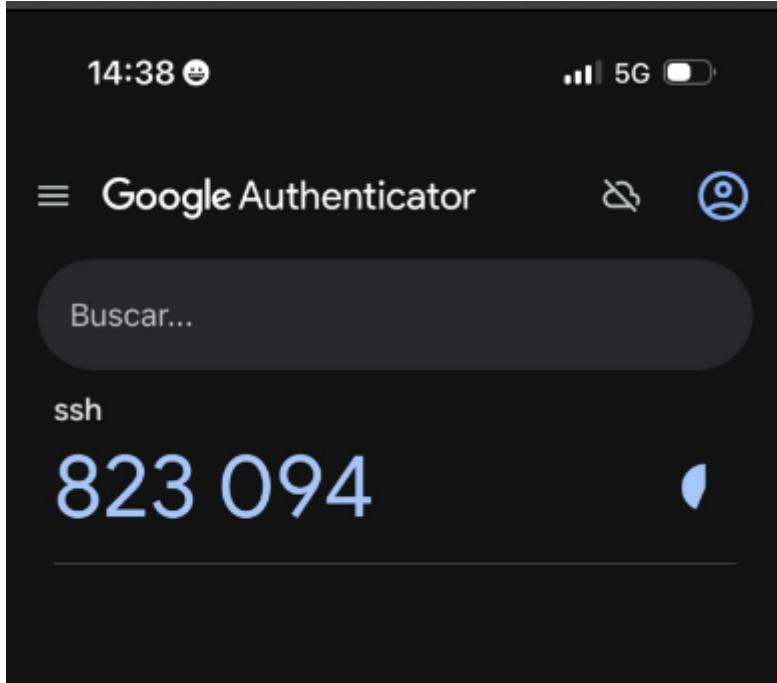
Introduciremos el código en la app de Google authenticator



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

```
GNU nano 7.2                               /etc/pam.d/sshd *
# PAM configuration for the Secure Shell service
auth required pam_google_authenticator.so
# Standard Unix authentication.
@include common-auth

# Disallow non-root logins when /etc/nologin exists.
account    required    pam_nologin.so

# Uncomment and edit /etc/security/access.conf if you need to set complex
# access limits that are hard to express in sshd_config.
# account    required    pam_access.so

# Standard Unix authorization.
@include common-account

# SELinux needs to be the first session rule. This ensures that any
# lingering context has been cleared. Without this it is possible that a
# module could execute code in the wrong domain.
session [success=ok ignore=ignore module_unknown=ignore default=bad]          pam>

^G Help      ^O Write Out ^W Where Is   ^K Cut      ^T Execute   ^C Location
^X Exit     ^R Read File  ^\ Replace   ^U Paste    ^J Justify   ^/ Go To Line
```

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 imágenes

```
GNU nano 7.2          /etc/ssh/sshd config *
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin prohibit-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
UsePAM yes
ChallengeResponseAuthentication yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
```

```
GNU nano 7.2          /etc/ssh/sshd config *
#
Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

AuthenticationMethods publickey,keyboard-interactive

#LoginGraceTime 2m
```

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