

# CONEXIÓN SEGURA A UN SERVIDOR SSH

---

>\_

SSH

---

SERVICIOS EN RED  
JUAN CARLOS NAVIDAD GARCÍA

# Índice

1. ¿Qué es SSH?:.....	3
2. ¿Qué es Webmin?:.....	3
3. ¿Cómo instalar Webmin?:.....	3
4. ¿Qué es OpenSSH?: .....	5
5. ¿Cómo instalar OpenSSH?: .....	6
6. ¿Cómo configurar el servidor SSH?: .....	7
6.1. ¿Cómo configurar el inicio de sesión por clave pública?:.....	11
7. ¿Cómo conectarse al servidor SSH?:.....	14
8. ¿Cómo modificar el mensaje de bienvenida del servidor?: .....	16
9. Uso de aplicaciones gráficas:.....	18
10. Traspaso de archivos desde el servidor al cliente: .....	19



## 1. ¿Qué es SSH?:

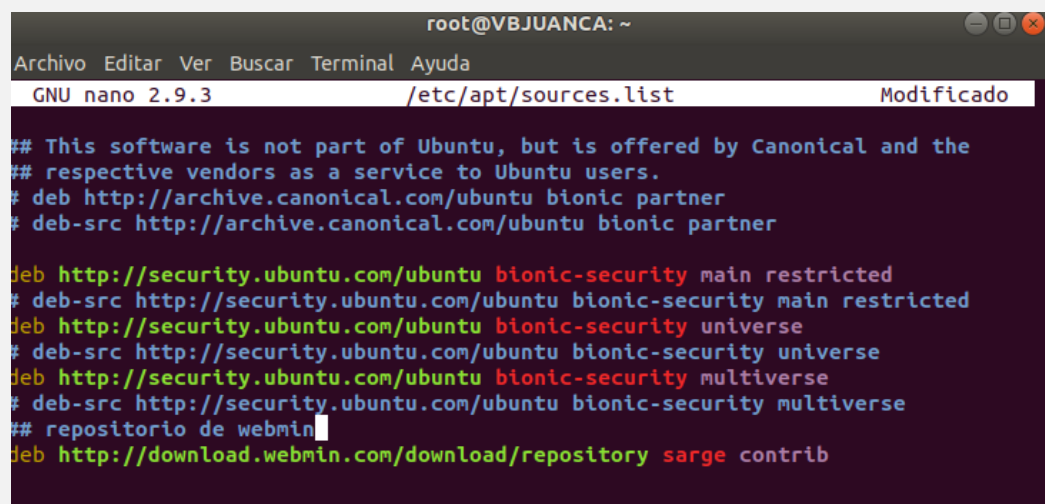
**SSH** es una sigla, o acrónimo, para el término **Secure Shell**, que significa **cápsula segura**. Es un protocolo que posibilita el acceso y la administración de un servidor a través de una **puerta trasera** (backdoor). Y, a diferencia de otros protocolos como HTTP o FTP, **SSH** establece **conexiones seguras entre los dos sistemas**. Básicamente, permite acceder a otro equipo a través de la red, ejecutar comandos en la máquina remota, mover ficheros entre dos máquinas, etc. Proveyendo **autenticación y comunicaciones seguras sobre canales inseguros**.

## 2. ¿Qué es Webmin?:

Para realizar esta práctica, vamos a utilizar un sistema Linux con **OpenSSH** mediante la interfaz gráfica de **Webmin**. Como ya he dicho en prácticas anteriores **Webmin** es una herramienta que nos proporciona una interfaz gráfica intuitiva y fácil de usar para **administrar el sistema, servidores**, etc.

## 3. ¿Cómo instalar Webmin?:

- En primer lugar, debemos añadir el repositorio **Webmin** para poder instalar y actualizar **Webmin** fácilmente usando nuestro **administrador de paquetes /etc/apt/sources.list**. Esto se hace agregando el repositorio:
  - **deb http://download.webmin.com/download/repository sarge contrib**



```
root@VBJUANCA: ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/apt/sources.list Modificado

## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu bionic partner
# deb-src http://archive.canonical.com/ubuntu bionic partner

deb http://security.ubuntu.com/ubuntu bionic-security main restricted
# deb-src http://security.ubuntu.com/ubuntu bionic-security main restricted
deb http://security.ubuntu.com/ubuntu bionic-security universe
# deb-src http://security.ubuntu.com/ubuntu bionic-security universe
deb http://security.ubuntu.com/ubuntu bionic-security multiverse
# deb-src http://security.ubuntu.com/ubuntu bionic-security multiverse
## repositorio de webmin
deb http://download.webmin.com/download/repository sarge contrib
```

- A continuación, agrego la **clave PGP de Webmin** para que el sistema confíe en el nuevo repositorio:
  - **wget http://www.webmin.com/jcameron-key.asc**
  - **sudo apt-key add jcameron-key.asc**

```

root@VBJUANCA:~# wget http://www.webmin.com/jcameron-key.asc
--2021-10-16 12:47:38-- http://www.webmin.com/jcameron-key.asc
Resolviendo www.webmin.com (www.webmin.com)... 216.105.38.11
Conectando con www.webmin.com (www.webmin.com)[216.105.38.11]:80... conectado.
Petición HTTP enviada, esperando respuesta... 301 Moved Permanently
Ubicación: https://www.webmin.com/jcameron-key.asc [siguiente]
--2021-10-16 12:47:40-- https://www.webmin.com/jcameron-key.asc
Conectando con www.webmin.com (www.webmin.com)[216.105.38.11]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1320 (1,3K) [text/plain]
Guardando como: "jcameron-key.asc"

jcameron-key.asc 100%[=====] 1,29K --.-KB/s en 0s
2021-10-16 12:47:41 (603 MB/s) - "jcameron-key.asc" guardado [1320/1320]

root@VBJUANCA:~# sudo apt-key add jcameron-key.asc
OK
root@VBJUANCA:~#

```

- Luego, actualizo la lista de paquetes para que incluya el repositorio **Webmin**:
  - **sudo apt update**

```

root@VBJUANCA:~# sudo apt update
Obj:1 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease
Obj:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease
Ign:5 http://download.webmin.com/download/repository sarge InRelease
Des:6 http://download.webmin.com/download/repository sarge Release [16,9 kB]
Des:7 http://download.webmin.com/download/repository sarge Release.gpg [173 B]
Des:8 http://download.webmin.com/download/repository sarge/contrib amd64 Package
s [1.387 B]
Des:9 http://download.webmin.com/download/repository sarge/contrib i386 Packages
[1.387 B]
Descargados 19,8 kB en 2s (9.481 B/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 308 paquetes. Ejecute «apt list --upgradable» para verlos.
root@VBJUANCA:~#

```

- Finalmente instalo **Webmin**:
  - **sudo apt install webmin**

```
root@VBJUANCA: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
root@VBJUANCA:~# sudo apt install webmin  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.  
fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0  
gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0  
grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1  
libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1  
libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcms-0.5-5v5  
libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libedataserverui-1.2-2 libeot0  
libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14  
libfreerdp-client2-2 libfreerdp2-2 libglibc2 libgee-0.8-2 libgexiv2-2  
libgom-1.0-0 libgpgmepp6 libgpod-common libgpod4 liblangtag-common  
liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0 libmshpub-0.1-1  
libodfgen-0.1-1 libqwing2v5 libraw16 libvenge-0.0-0 libsgutils2-2  
libssh-4 libsuitesparseconfig5 libvncclient1 libwinpr2-2 libxapian30  
libxmlsec1 libxmlsec1-nss lp-solve media-player-info python3-mako  
python3-markupsafe syslinux syslinux-common syslinux-legacy  
usb-creator-common  
Utilice «sudo apt autoremove» para eliminarlos.  
Se instalarán los siguientes paquetes adicionales:  
libauthen-pam-perl
```

#### 4. ¿Qué es OpenSSH?:

**OpenSSH** es la versión de **código abierto** de las herramientas de **Secure Shell (SSH)** que usan los administradores de **Linux**. Básicamente funciona como un conjunto de aplicaciones que permiten realizar **comunicaciones cifradas a través de una red**, usando el **protocolo SSH**.



## 5. ¿Cómo instalar OpenSSH?:

**OpenSSH** se instala de una manera bastante fácil, simplemente hay que introducir el comando:

- **sudo apt-get install openssh-server** (En caso de servidor)
- **sudo apt-get install openssh-client** (En caso de cliente)
- **sudo apt-get install openssh-server openssh-client** (Si queremos instalar los dos.)

```
jnav@ROGZephyrus:~$ sudo apt-get install openssh-client openssh-server
[sudo] contraseña para jnav:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssh-client ya está en su versión más reciente (1:8.2p1-4ubuntu0.4).
fijado openssh-client como instalado manualmente.
Se instalarán los siguientes paquetes adicionales:
ncurses-term openssh-sftp-server ssh-import-id
```

Una vez introducido el comando, **OpenSSH** ya se nos habrá instalado, para configurarlo nos iremos a **Webmin**.

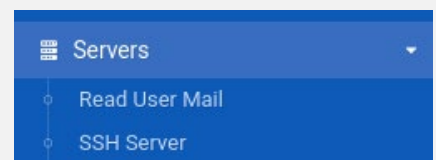
Nos dirigimos al navegador y escribimos **https://localhost:10000** para acceder a **Webmin**. Una vez iniciada la sesión en **Webmin**, nos dirigimos al apartado "**Un-used Modules**". Allí nos encontraremos con el botón **SSH Server**.



Para mayor comodidad, le daremos al botón "**Refresh Modules**" para subir el **servidor SSH** al apartado de **servidores**.

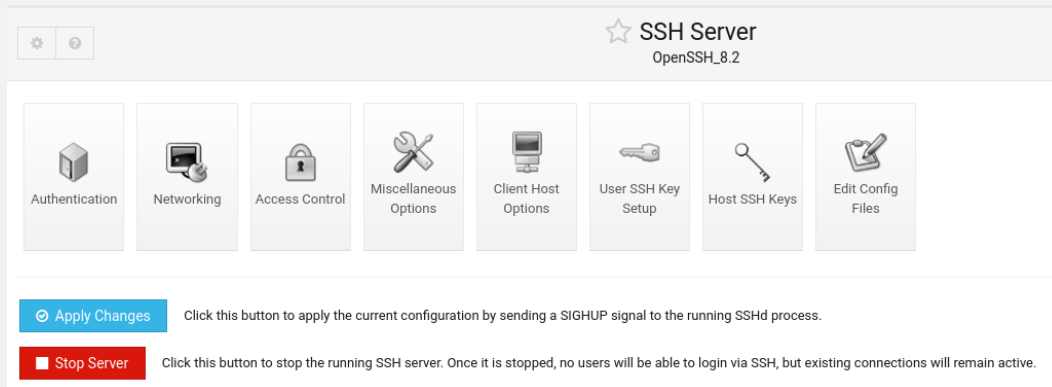


Cuando le hayamos dado al botón anterior, si nos dirigimos al apartado de **servidores**, nos encontraremos con el botón "**SSH Server**" junto a los demás servidores:



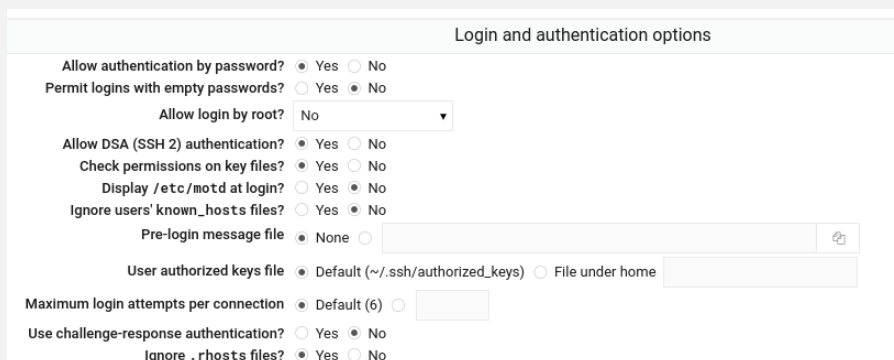
## 6. ¿Cómo configurar el servidor SSH?:

Cuando hayamos realizado todos los pasos anteriores, nos iremos al **servidor SSH**:

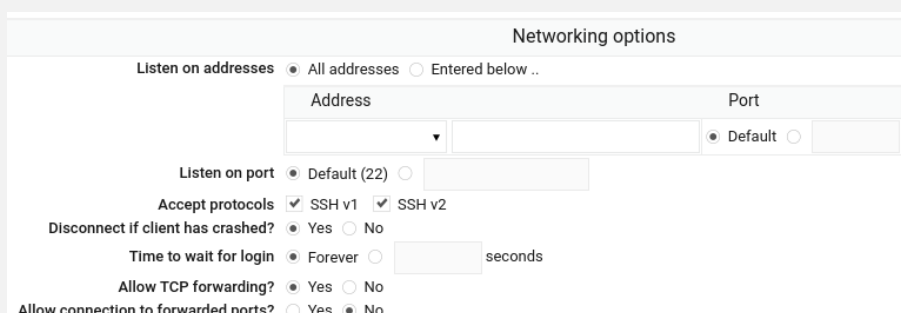


Esta será la pantalla de configuración de **OpenSSH**, en la cual nos encontraremos con las siguientes opciones:





- **Autenticación:** podremos configurar todos los parámetros de acceso al **servidor SSH**, como puede ser el acceso con usuario y contraseña, acceso con clave, permitir el usuario root, etc.



- **Red:** podremos configurar el protocolo SSH, permitir acceso a direcciones IP introducidas, cambiar el puerto por defecto, etc.



- **Control de acceso:** configurar el acceso para usuarios, grupos de usuarios, denegar el acceso, etc.

Network and login access control options			
Only allow users	<input checked="" type="radio"/> All	<input type="radio"/>	
Only allow members of groups	<input checked="" type="radio"/> All	<input type="radio"/>	
Deny users	<input checked="" type="radio"/> All	<input type="radio"/>	
Deny members of groups	<input checked="" type="radio"/> All	<input type="radio"/>	

- **Opciones avanzadas:** activar el uso de interfaz gráfica entre otras opciones.

### Other miscellaneous SSH server options

Allow X11 connection forwarding? ☒ Yes ☐ No

X11 display offset ☒ Default ☐

Full path to xauth program ☒ Default ☐

System log facility ☒ Default ☐ AUTH

Logging level ☒ Default ☐ QUIET

Server key size ☒ Default ☐  bits

Server key regeneration interval ☒ Never ☐  seconds

PID file ☒ Default ☐

Use separate unprivileged process? ☒ Yes ☐ No


- **Opciones de host:** al igual que con DHCP, podemos configurar los clientes directamente desde el servidor.

←

?

☆ Client Host Options

⊞ Add options for client host



All Hosts

⊞ Add options for client host

- **Clave del servidor SSH:** crear clave privada y pública para el acceso al servidor SSH

This page allows you to configure the automatic setup of SSH for new Unix users created on your system. If configured, new users will not have to run `ssh-keygen` before using SSH.


Setup SSH key for new Unix users? ☐ Yes ☒ No

Copy new `identity.pub` to `authorized_keys`? ☒ Yes ☐ No

Use password as key passphrase? ☐ Yes ☒ No

Key type <Automatic> ▼



- **Claves de los clientes SSH:** organizar todas las claves de los clientes SSH.

Key filename:  



```
1  ssh -rsa AAAAB3NzaC1yc2EAAAADAQABAAQgC6+qX1y0KKve1Yk2sU1yF5yuCn/em9n5+Q3Rk9BcoCbACHwT9MCQIwhwU1MuWpR70YUWb2JrwTARj6fnFQ05Y9GtXnQy8A:
2
```





- **Editar los archivos de configuración:** te permite editar directamente desde Webmin los archivos de configuración del servidor y cliente SSH.

Editing config file:   



```
1 # $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $
2
3 # This is the sshd server system-wide configuration file. See
4 # sshd_config(5) for more information.
5
6 # This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
7
8 # The strategy used for options in the default sshd_config shipped with
9 # OpenSSH is to specify options with their default value where
10 # possible, but leave them commented. Uncommented options override the
11 # default value.
12
13 Include /etc/ssh/sshd_config.d/*.conf
14
15 #Port 22
16 #AddressFamily any
17 #ListenAddress 0.0.0.0
18 #ListenAddress ::
19
20 #HostKey /etc/ssh/ssh_host_rsa_key
21 #HostKey /etc/ssh/ssh_host_ecdsa_key
22 #HostKey /etc/ssh/ssh_host_ed25519_key
23
24 # Ciphers and keying
25 #RekeyLimit default none
```

 Save  Save and close

**/etc/sshd\_config** es el archivo de configuración del servidor.

Editing config file:   

```
1 # This is the ssh client system-wide configuration file. See
2 # ssh_config(5) for more information. This file provides defaults for
3 # users, and the values can be changed in per-user configuration files
4 # or on the command line.
5
6 # Configuration data is parsed as follows:
7 # 1. command line options
8 # 2. user-specific file
9 # 3. system-wide file
10 # Any configuration value is only changed the first time it is set.
11 # Thus, host-specific definitions should be at the beginning of the
12 # configuration file, and defaults at the end.
13
14 # Site-wide defaults for some commonly used options. For a comprehensive
15 # list of available options, their meanings and defaults, please see the
16 # ssh_config(5) man page.
17
18 Include /etc/ssh/ssh_config.d/*.conf
19
20 Host *
21 # ForwardAgent no
22 # ForwardX11 no
23 # ForwardX11Trusted yes
24 # PasswordAuthentication yes
25 # HostbasedAuthentication no
```

 Save  Save and close

**/etc/ssh\_config** es el archivo de configuración del cliente.

El **servidor SSH** funciona desde el momento en el que se instala. Simplemente, si queremos, podemos editar los parámetros del servidor para aumentar su seguridad o comodidad a nuestro gusto.

Para esta práctica, vamos a configurar el servidor para que acepte:

- Uso del puerto 22;
- Uso de interfaz gráfica;
- Uso de la versión 2 del protocolo SSH;
- Uso de la autenticación por usuario y contraseña;
- Uso de la autenticación por clave pública RSA del cliente.

```
Include /etc/ssh/sshd_config.d/*.conf

protocol 2
port 22
HostKey /etc/ssh/ssh_host_rsa_key
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
X11Forwarding yes
PasswordAuthentication yes
```

El archivo **/etc/ssh/sshd\_config** debe contener las líneas descritas anteriormente.

En el caso del cliente, debe contener las siguientes líneas para esta práctica:

- Uso del puerto 22;
- Uso de interfaz gráfica;
- Uso de la versión 2 del protocolo SSH;
- Uso de la autenticación por usuario y contraseña;
- Uso de la autenticación por clave pública RSA del cliente;
- Checkeo de la clave pública utilizada.

```
Include /etc/ssh/ssh_config.d/*.conf

Host *
    ForwardX11 yes
    PasswordAuthentication yes
    StrictHostKeyChecking ask
    IdentityFile ~/.ssh/id_rsa
    Port 22
    Protocol 2
    PubkeyAuthentication yes
```

El archivo **/etc/ssh/ssh\_config** del cliente debe contener las líneas descritas anteriormente.

Podemos comprobar que el **servidor SSH** está en funcionamiento y que todo está correcto con el comando:

- **sudo service ssh status**

```
jnav@ROGZephyrus:~$ sudo service ssh stop
jnav@ROGZephyrus:~$ sudo service ssh start
jnav@ROGZephyrus:~$ sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en
   Active: active (running) since Thu 2022-01-20 09:06:30 CET; 3s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 18996 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 18997 (sshd)
   Tasks: 1 (limit: 18353) server. Once it is stopped, no users will be able to login via SSH, but
   Memory: 1.0M
   CGroup: /system.slice/ssh.service
           └─18997 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

ene 20 09:06:30 ROGZephyrus systemd[1]: Starting OpenBSD Secure Shell server...
ene 20 09:06:30 ROGZephyrus sshd[18997]: Server listening on 0.0.0.0 port 22.
ene 20 09:06:30 ROGZephyrus sshd[18997]: Server listening on :: port 22.
ene 20 09:06:30 ROGZephyrus systemd[1]: Started OpenBSD Secure Shell server.
jnav@ROGZephyrus:~$
```

## 6.1. ¿Cómo configurar el inicio de sesión por clave pública?:

Una vez hechas las configuraciones anteriores, debemos **crear una clave pública** en el cliente para que se pueda autenticar en el **servidor SSH**.

Para poder hacer esto, primero, crearemos un **par de claves, pública y privada**, de las cuales solo utilizaremos la **pública**.

En el cliente, ejecutaremos el siguiente comando para crear el **par de claves**:

- **ssh-keygen -t rsa**

Nos pedirá asignarle una contraseña, aunque si le damos a **Enter**, podemos dejar las claves sin contraseña.

```
jnav@jnav-vb:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jnav/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jnav/.ssh/id_rsa.
Your public key has been saved in /home/jnav/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:BtMa051BY5KH1mq2HPV8MBnWYiTBmkDiB8B6l1pxsGE jnav@jnav-vb
The key's randomart image is:
+---[RSA 2048]---+
|o.o.E. .+BO+
| o +oo.o=*O+.
|. . o==+o=o=
|. .+ o*+ + .
|. + .oSo .
|. . .o
|. . .
|. . .
+-----[SHA256]-----+
jnav@jnav-vb:~$
```

El par de claves se creará en el directorio oculto **/home/user/.ssh**.

Desde ese directorio, copiaremos la **clave pública**, nombrada por defecto como **"id\_rsa.pub"** al directorio **/home/user** del servidor.

Podemos utilizar el comando **scp** para facilitar la copia.

```
jnav@jnav-vb:~$ scp id_rsa.pub jnav@192.168.103.232:/home/jnav
id_rsa.pub
100% 394 396.3KB/s 00:00
jnav@jnav-vb:~$
```

Una vez tengamos la clave pública del cliente en el servidor, debemos crear el archivo **authorized\_keys** (en el caso de que no esté creado) en el directorio **/home/user/.ssh**.

```
jnav@ROGZephyrus:~/ssh$ touch authorized_keys
jnav@ROGZephyrus:~/ssh$ ls
authorized_keys
```

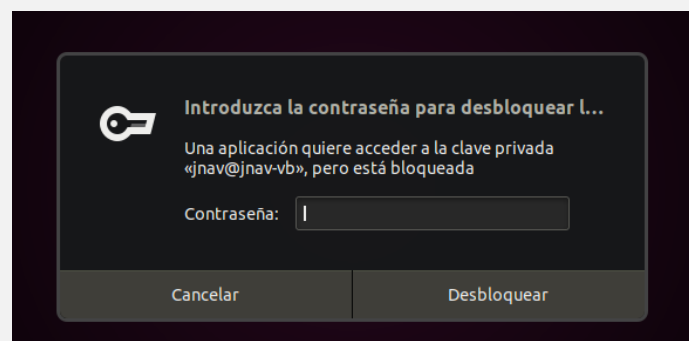
Cuando lo hayamos creado, copiaremos el contenido de la clave pública del cliente dentro de este archivo. Podemos utilizar el comando **cat** con el redireccionamiento de comandos:

```
jnav@ROGZephyrus:~$ cat id_rsa.pub >> .ssh/authorized_keys
jnav@ROGZephyrus:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCuFFF6zynYGqSmg5WIf/yWAV9eGH
Ia1tkPsPQ28u+s1lHdqZf7ufhpXMl1WuibNoZzDMAy/dbFewRA1qKKIMUyWwGM3NF
656kRoS5YqsN83wt05uoEJSliZIBECm/DzP76ok7xTS5pBBKZc0aJ4u2sHLp2P5Wux
jnav@ROGZephyrus:~$
```

Después de realizar todos estos pasos, comprobaremos que realmente nos pide la contraseña de la clave pública cuando nos conectamos al servidor:

Ahora, al intentar conectarnos al servidor, nos pedirá autenticar la clave. Solo nos la pedirá una vez después de autenticarla, aunque cada vez que reiniciemos el cliente, volverá a pedirnos la contraseña:

**Al acceder con la clave, no nos pedirá el usuario y contraseña del servidor.**



```
jnav@jnav-vb:~/.ssh$ ssh 192.168.103.232
Welcome to elementary OS 6 Jólnir (GNU/Linux 5.13.0-27-generic x86_64)
Built on
Website: https://elementary.io
Last login: Thu Jan 20 13:45:33 2022 from 192.168.103.252
jnav@ROGZephyrus:~$
```

## 7. ¿Cómo conectarse al servidor SSH?:

Si tenemos bien configurado el **servidor** y el **cliente SSH**, nos podremos conectar sin problemas y nos podremos autenticar tanto con **usuario** y **contraseña**, como con la **clave pública** del cliente.

Los comandos más comunes para acceder al servidor SSH son:

- **ssh <ip del servidor>** en el caso de que exista el mismo usuario en el cliente y en el servidor;
- **ssh <usuario@ip del servidor>** para conectarnos a un usuario en específico del servidor.
- **ssh -j <usuario@ip del servidor:puerto>** para conectarnos al servidor mediante un puerto en específico configurado.

Un ejemplo sería este:

```
jnav@jnav-vb:~$ ssh jnav@192.168.103.232
Bienvenido al servidor SSH de Juan Carlos Navidad

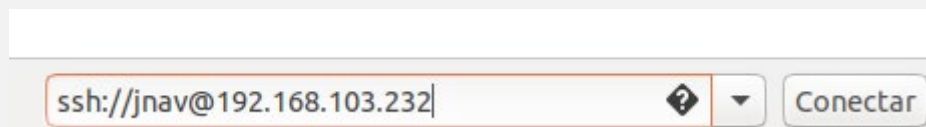
SSH Juan Carlos
Last login: Mon Jan 24 08:25:35 2022 from 192.168.103.87
```

Si estamos conectados a un servidor DNS configurado con dominios para los equipos, podemos cambiar la dirección IP por el dominio de destino:

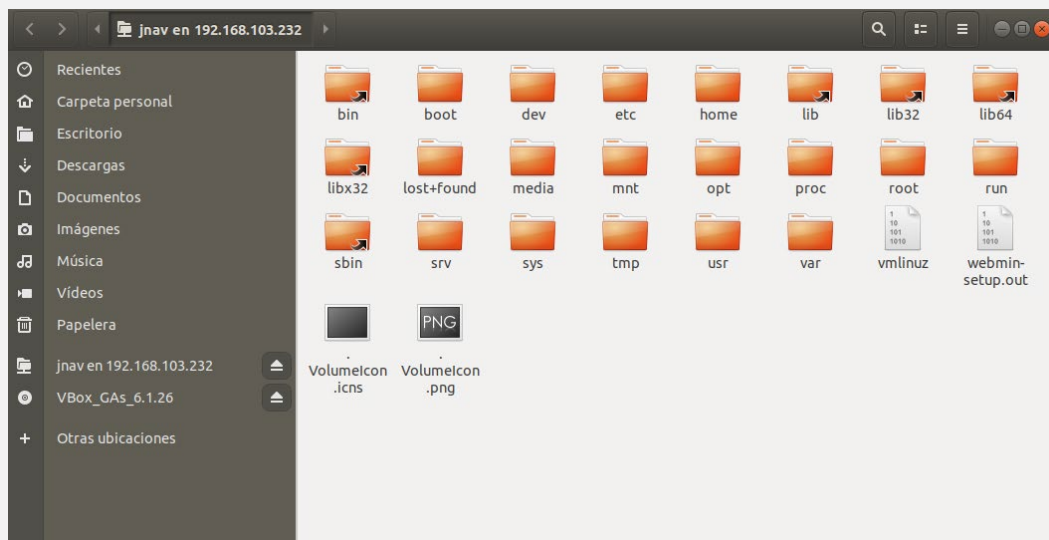
```
jnav@jnav-vb:~$ ssh jnav@rog232.aulaSER232.com.
The authenticity of host 'rog232.aulaser232.com. (192.168.103.232)' can't be established.
ECDSA key fingerprint is SHA256:NKCEJ4VxCHhP61LgCt8ijG/8w12oengLL4n3b4zMS2k.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'rog232.aulaser232.com.' (ECDSA) to the list of known hosts.
jnav@rog232.aulaser232.com.'s password:
Welcome to elementary OS 6 Jólnir (GNU/Linux 5.13.0-27-generic x86_64)
Built on
Website: https://elementary.io
Last login: Thu Jan 20 07:47:23 2022 from 192.168.103.21
jnav@ROGZephyrus:~$
```

También nos podemos conectar desde el explorador de archivos del entorno gráfico:

- Abrimos el explorador;
- Nos dirigimos a "**Otras ubicaciones**";
- En la parte inferior derecha tenemos una barra para conexión a servidores;
- Introducimos la dirección del servidor de la siguiente manera:  
**ssh://<usuario@ip del servidor>;**
- Nos pedirá la contraseña;
- Y ya estaríamos dentro.



Una vez iniciada la sesión en el **servidor**, nos aparecerá en el apartado izquierdo del explorador hasta que nos salgamos de la sesión:







Ahora eliminaremos el mensaje del sistema que aparece nada más conectarse al **servidor SSH**:

```
jnav@jnav-vb:~$ ssh jnav@192.168.103.232
Welcome to elementary OS 6 Jólnir (GNU/Linux 5.13.0-27-generic x86_64)
Built on
Website: https://elementary.io
Last login: Thu Jan 20 14:14:54 2022 from 192.168.103.252
```

Para eliminar este mensaje simplemente debemos **quitarle permisos de ejecución** a los archivos de la carpeta **/etc/update-motd.d/**:

```
sudo chmod -x /etc/update-motd.d/*
```

También se pueden editar esos archivos, pero sería más tedioso.

Una vez realizados todos los pasos anteriores, podemos observar el resultado:

```
jnav@jnav-vb:~$ ssh jnav@192.168.103.232
Bienvenido al servidor SSH de Juan Carlos Navidad
SSH Juan Carlos
Last login: Mon Jan 24 08:25:35 2022 from 192.168.103.87
```

## 9. Uso de aplicaciones gráficas:

Desde una **consola SSH**, podemos iniciar aplicaciones con **interfaz gráfica**. Pero previamente debemos activar la opción tanto en el archivo de configuración del servidor **/etc/sshd\_config** como en el del cliente **/etc/ssh\_config**.

En el archivo **/etc/sshd\_config** del servidor deberemos comprobar que está la siguiente línea o añadirla:

- **X11Forwarding yes**

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

En el archivo **/etc/ssh\_config** del cliente deberemos quitarle la almohadilla y añadirle "yes" al final a la línea:

- **#ForwardX11 no**

```
Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
```

Los sustituiríamos por:

- **ForwardX11 yes**

```
Host *
# ForwardAgent no
ForwardX11 yes
# ForwardX11Trusted yes
```

Si hemos realizado todos los pasos anteriores, cuando accedamos como cliente a un **servidor SSH**, podremos ejecutar aplicaciones con **interfaz gráfica**.

Para hacer el ejemplo, he probado con la aplicación **Xeyes**.

**Xeyes** es una aplicación gráfica que muestra dos ojos que siguen con la mirada el movimiento del cursor por la pantalla.

```
jnav@jnav-vb:~$ ssh jnav@192.168.103.232
Welcome to elementary OS 6 Jólnir (GNU/Linux 5.13.0-27-generic x86_64)
Built on
Website: https://elementary.io
Last login: Thu Jan 20 14:14:54 2022 from 192.168.103.252
jnav@ROGZephyrus:~$ xeyes
xeyes (o... - - -)
  (o) (o)
```

## 10. Traspaso de archivos desde el servidor al cliente:

Existe un comando para realizar traspasos de archivos desde el cliente sin iniciar en el **servidor SSH** e incluso copiar archivos de dentro del servidor.

Para poder transferir archivos se utiliza el comando **scp** de la siguiente manera:

- **scp <archivo> usuario@ip:<destino>**
- **scp usuario@ip:<archivo> <destino>**

En este caso, he creado el archivo **JuanCarlos.txt** relleno con mi nombre, DNI, nombre del cliente y dirección IP.

```
GNU nano 2.9.3 JuanCarlos.txt
Juan Carlos Navidad
26529341Z
jnav-vb:192.168.103.232
```

Y ahora, para transferirlo al directorio personal del servidor, he utilizado el comando:

- **scp <archivo> usuario@ip: <destino>**

```
jnav@jnav-vb:~$ nano JuanCarlos.txt
jnav@jnav-vb:~$ scp JuanCarlos.txt jnav@192.168.103.232:/home/jnav
JuanCarlos.txt                                100% 56   96.2KB/s   00:00
jnav@jnav-vb:~$
```

Finalmente, vamos a comprobar si el archivo se encuentra completo en el directorio del usuario en el servidor:

```
jnav@jnav-vb:~$ ssh jnav@192.168.103.232
Bienvenido al servidor SSH de Juan Carlos Navidad

SSH Juan Carlos

Last login: Mon Jan 24 08:25:35 2022 from 192.168.103.87
jnav@ROGZephyrus:~$ cat JuanCarlos.txt
Juan Carlos Navidad

26529341Z

jnav-vb:192.168.103.232
jnav@ROGZephyrus:~$
```