

|  |
| --- |
|  |
| SCP. Copias por SSH. uGeek Blog  CONEXIÓN SEGURA A UN SERVIDOR SSH |
|  |

SERVICIOS EN RED

JUAN CARLOS NAVIDAD GARCÍA

**Índice**

[1. ¿Qué es SSH?: 3](#_Toc93948784)

[2. ¿Qué es Webmin?: 3](#_Toc93948785)

[3. ¿Cómo instalar Webmin?: 3](#_Toc93948786)

[4. ¿Qué es OpenSSH?: 5](#_Toc93948787)

[5. ¿Cómo instalar OpenSSH?: 6](#_Toc93948788)

[6. ¿Cómo configurar el servidor SSH?: 7](#_Toc93948789)

[6.1. ¿Cómo configurar el inicio de sesión por clave pública?: 11](#_Toc93948790)

[7. ¿Cómo conectarse al servidor SSH?: 14](#_Toc93948791)

[8. ¿Cómo modificar el mensaje de bienvenida del servidor?: 15](#_Toc93948792)

[9. Uso de aplicaciones gráficas: 17](#_Toc93948793)

[10. Traspaso de archivos desde el servidor al cliente: 18](#_Toc93948794)



### ¿Qué es SSH?:

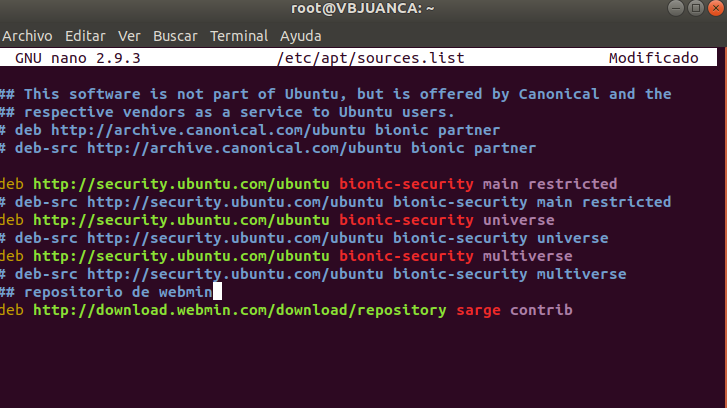
SSH es una sigla, o acrónimo, para el término Secure Shell, que significa cápsula segura. Es un protocolo que posibilita el acceso y la administración de un servidor a través de una puerta trasera (backdoor). Y, a diferencia de otros protocolos como HTTP o FTP, SSH establece conexiones seguras entre 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.

### ¿Qué es Webmin?:

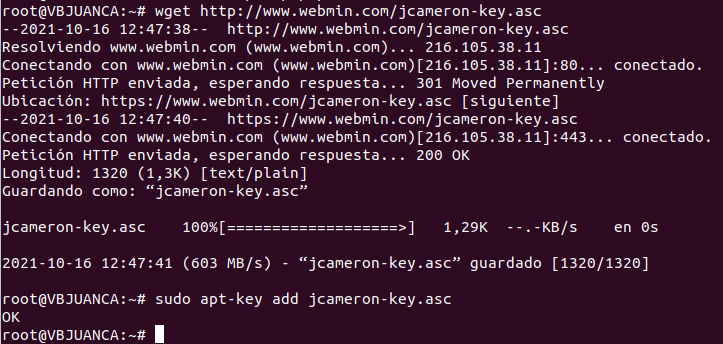
Para realizar esta práctica, vamos a utilizar un sistema Linux con OpenSSH mediante la interfaz gráfica de Webmin. Como ya he dicho en prácticas anteriores Webmin es una herramienta que nos proporciona una interfaz gráfica intuitiva y fácil de usar para administrar el sistema, servidores, etc.

### ¿Cómo instalar Webmin?:

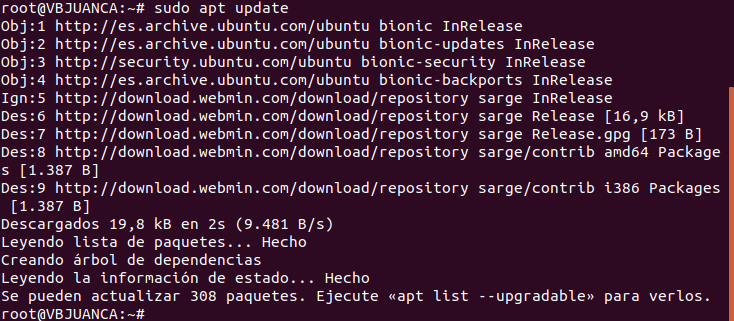
* En primer lugar, debemos añadir el repositorio **Webmin** para poder instalar y actualizar **Webmin** fácilmente usando nuestro **administrador de paquetes** **/etc/apt/sources.list**. Esto se hace agregando el repositorio:
* **deb http://download.webmin.com/download/repository sarge contrib**



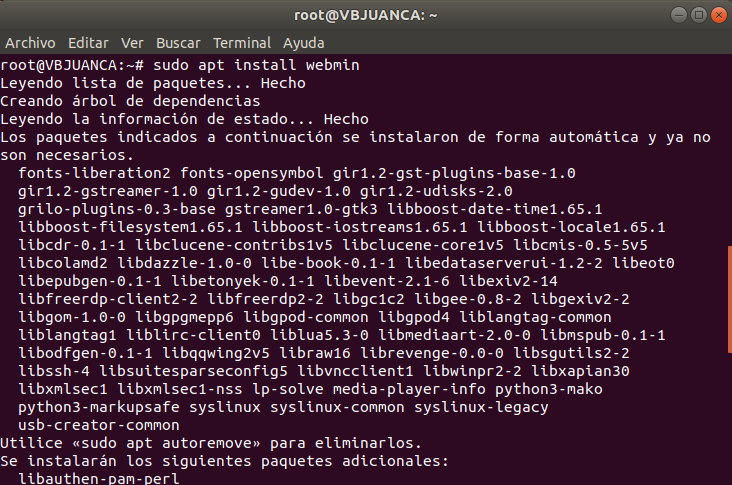
* A continuación, agrego la **clave PGP de Webmin** para que el sistema confíe en el nuevo repositorio:
* **wget http://www.webmin.com/jcameron-key.asc**
* **sudo apt-key add jcameron-key.asc**



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



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



### ¿Qué es OpenSSH?:

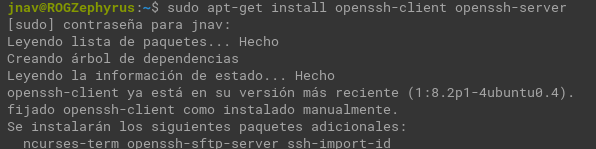
**OpenSSH** es la versión de **código abierto** de las herramientas de **Secure Shell (SSH)** que usan los administradores de **Linux**. Básicamente funciona como un conjunto de aplicaciones que permiten realizar **comunicaciones cifradas a través de una red**, usando el **protocolo SSH**.



### ¿Cómo instalar OpenSSH?:

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

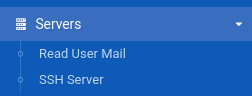
* **sudo apt-get install openssh-server** (En caso de servidor)
* **sudo apt-get install openssh-client** (En caso de cliente)
* **sudo apt-get install openssh-server openssh-client** (Si queremos instalar los dos.)



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

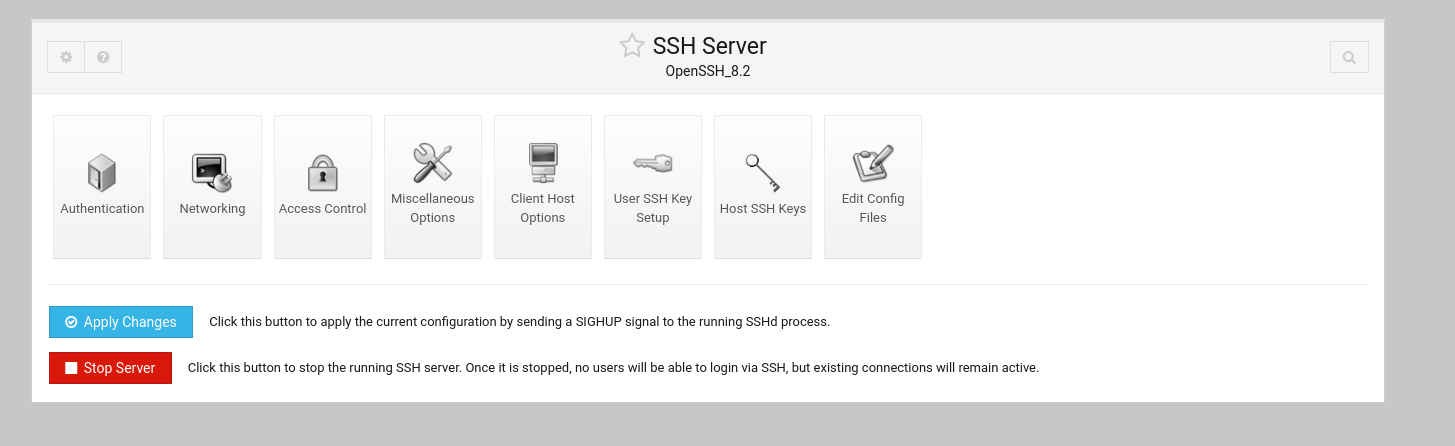
Nos dirigimos al navegador y escribimos **https://localhost:10000** para acceder a **Webmin**. Una vez iniciada la sesión en **Webmin**, nos dirigimos al apartado “**Un-used Modules**”. Allí nos encontraremos con el botón **SSH Server**.

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:

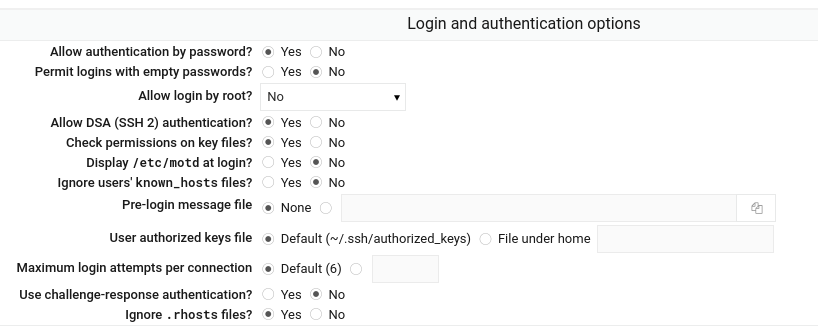
### ¿Cómo configurar el servidor SSH?:

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

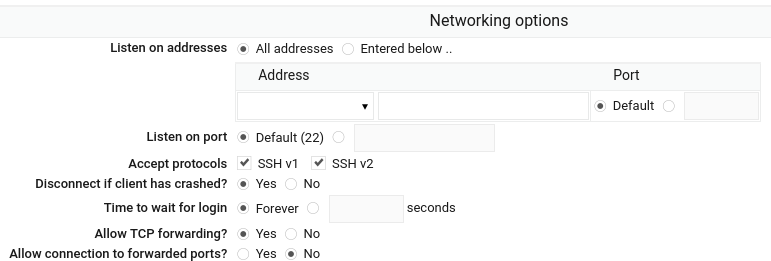


Esta será la pantalla de configuración de **OpenSSH**, en la cual nos encontraremos con las siguientes opciones:

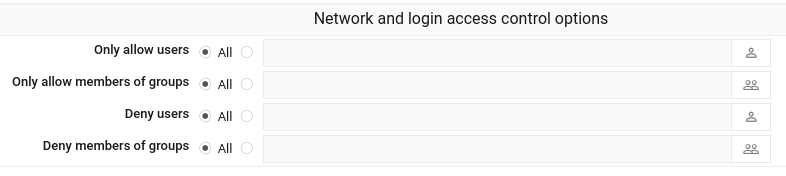
* **Autenticación**: podremos configurar todos los parámetros de acceso al **servidor SSH**, como puede ser el acceso con usuario y contraseña, acceso con clave, permitir el usuario root, etc.



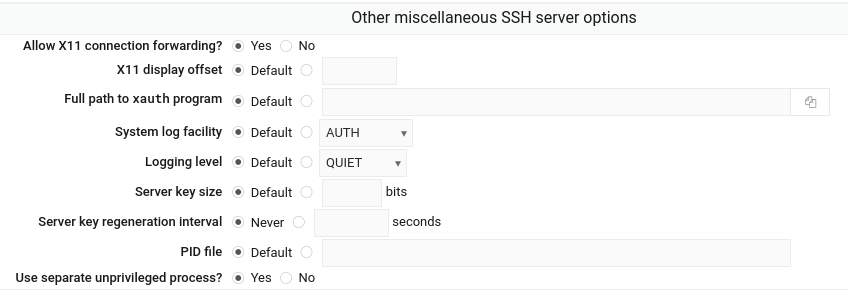
* **Red**: podremos configurar el protocolo SSH, permitir acceso a direcciones IP introducidas, cambiar el puerto por defecto, etc.



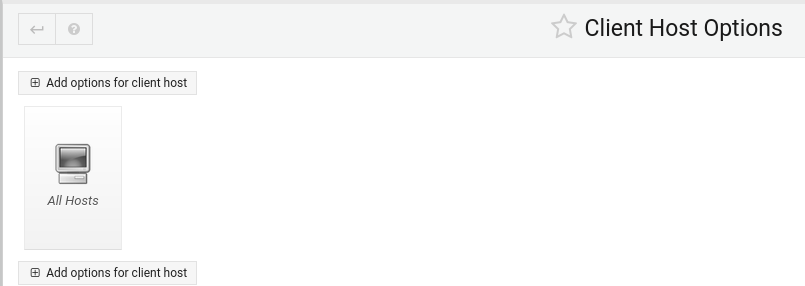
* **Control de acceso**: configurar el acceso para usuarios, grupos de usuarios, denegar el acceso, etc.



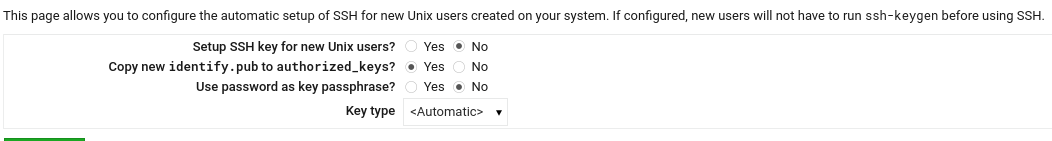
* **Opciones avanzadas**: activar el uso de interfaz gráfica entre otras opciones.



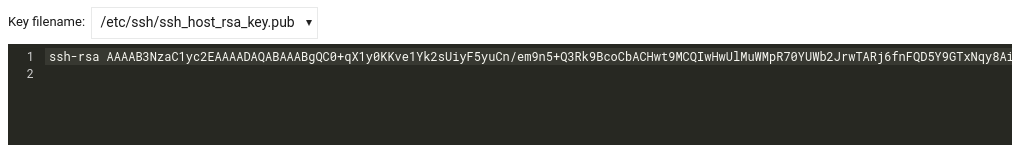
* **Opciones de host**: al igual que con DHCP, podemos configurar los clientes directamente desde el servidor.



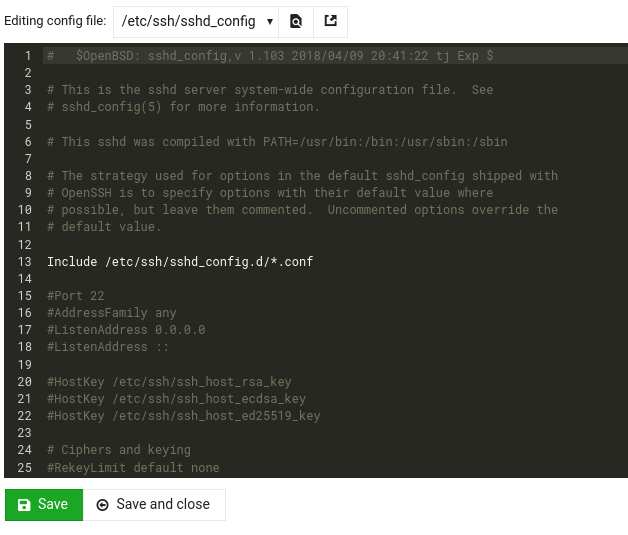
* **Clave del servidor SSH**: crear clave privada y pública para el acceso al servidor SSH



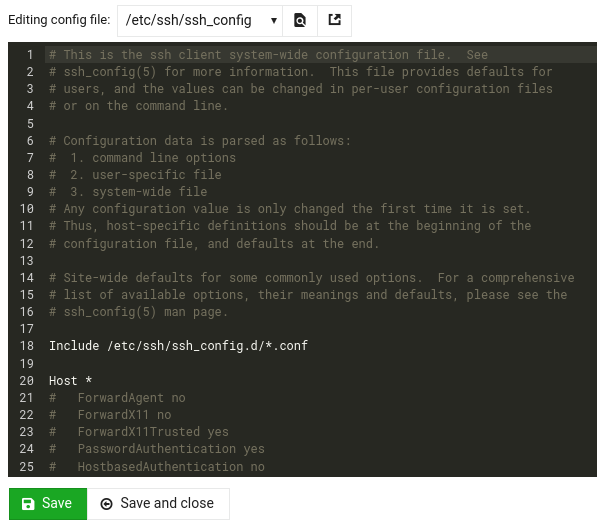
* **Claves de los clientes SSH**: organizar todas las claves de los clientes SSH.



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



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

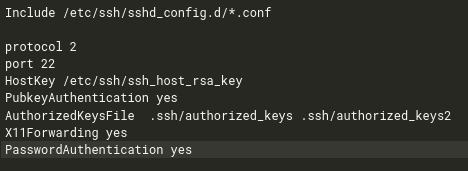


**/etc/ssh\_config** es el archivo de configuración del cliente.

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.

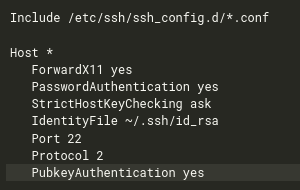
Para esta práctica, vamos a configurar el servidor para que acepte:

* 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.



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

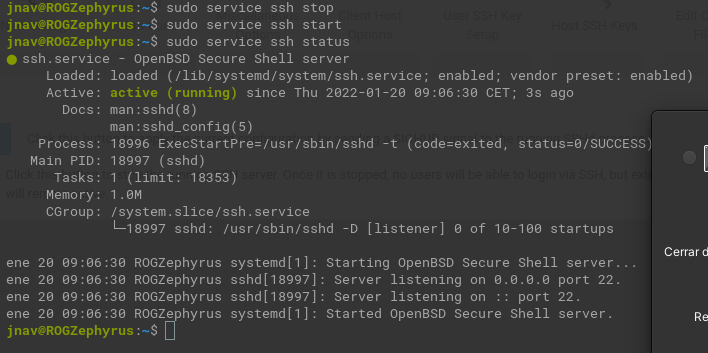
En el caso del cliente, debe contener las siguientes líneas para esta práctica:

* 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.

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**



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

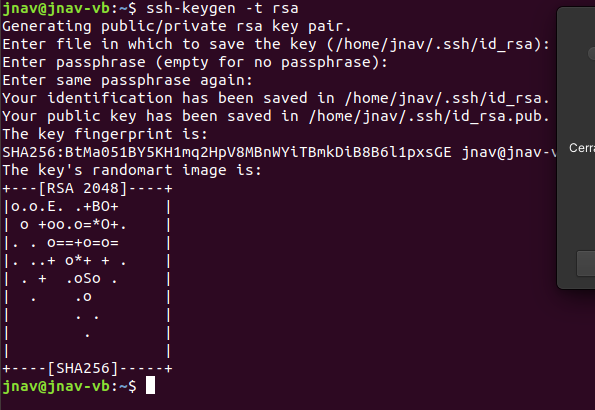
Una vez hechas las configuraciones anteriores, debemos **crear una clave pública** en el cliente para que se pueda autenticar en el **servidor SSH**.

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 clave**s:

* **ssh-keygen -t rsa**

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



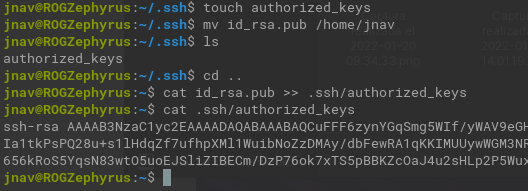
El par de claves se creará en el directorio oculto **/home/user/.ssh**.

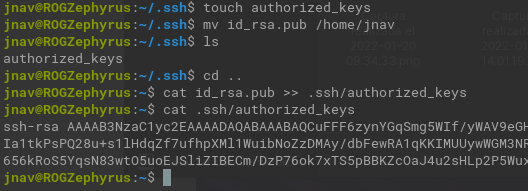
Desde ese directorio, copiaremos la **clave pública**, nombrada por defecto como “**id\_rsa.pub**” al directorio /**home/user del servidor**.

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

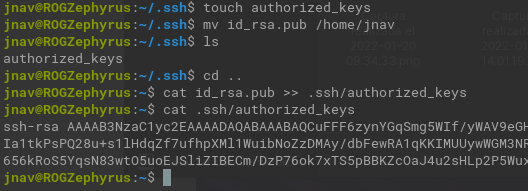


Una vez tengamos la clave pública del cliente en el servidor, debemos crear el archivo **authorized\_keys** (en el caso de que no esté creado) en el directorio **/home/user/.ssh.**





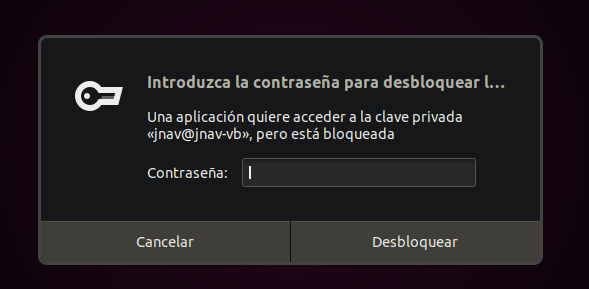
Cuando lo hayamos creado, copiaremos el contenido de la clave pública del cliente dentro de este archivo. Podemos utilizar el comando **cat** con el redireccionamiento de comandos:

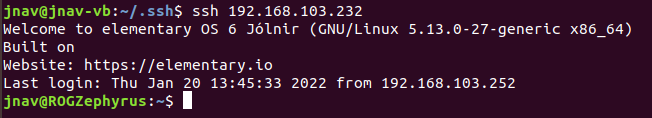


Después de realizar todos estos pasos, comprobaremos que realmente nos pide la contraseña de la clave pública cuando nos conectamos al servidor:

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.**





### ¿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.

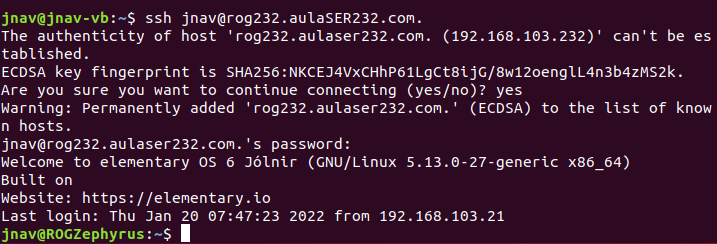
Los comandos más comunes para acceder al servidor SSH son:

* **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:

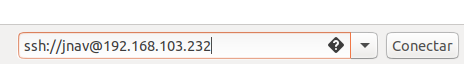


Si estamos conectados a un servidor DNS configurado con dominios para los equipos, podemos cambiar la dirección IP por el dominio de destino:

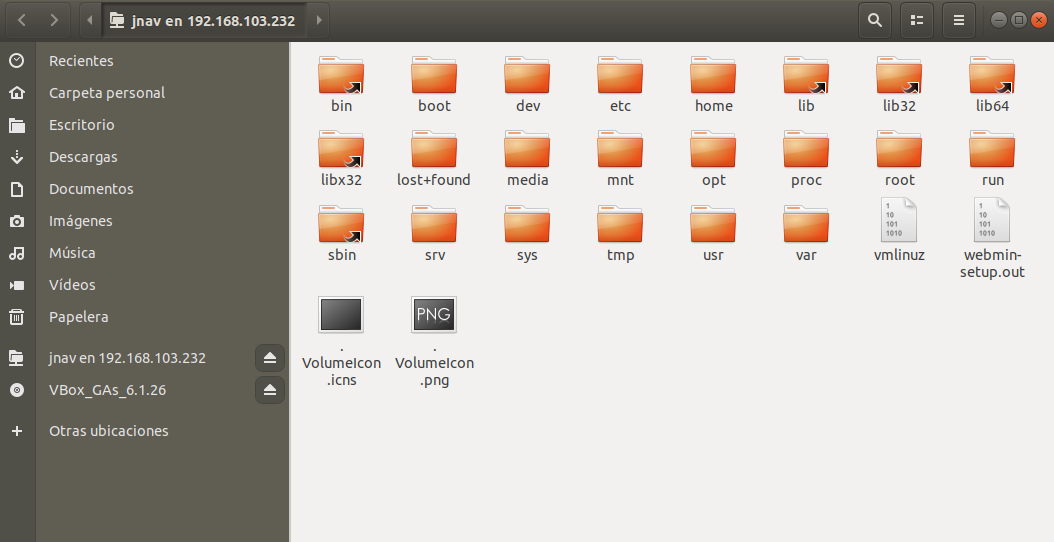


También nos podemos conectar desde el explorador de archivos del entorno gráfico:

* 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:



### ¿Cómo modificar el mensaje de bienvenida del servidor?:

Editar el **mensaje de bienvenida del servidor** es muy fácil, simplemente hay que modificar el archivo **/etc/motd** y añadirle lo que queramos que salga como mensaje de bienvenida.

En mi caso para que quede más bonito, he utilizado una aplicación llamada **figlet**, que cambia la fuente de un texto plano. La cual se puede instalar con el comando:

* **sudo apt-get install figlet**

Utilizarlo es muy fácil, simplemente introducimos el comando:

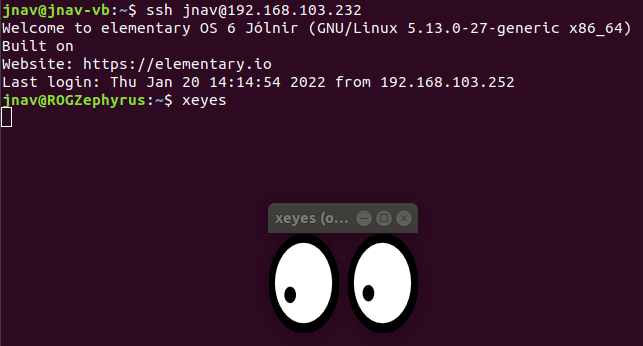
* **figlet <texto>**:



Para introducir ese texto de **figlet** dentro del archivo **/etc/motd**, utilizaremos el redireccionamiento de salidas “**>>**” para añadir la salida del comando **figlet** dentro del archivo:



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



Para eliminar este mensaje simplemente debemos **quitarle permisos de ejecución** a los archivos de la carpeta **/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:

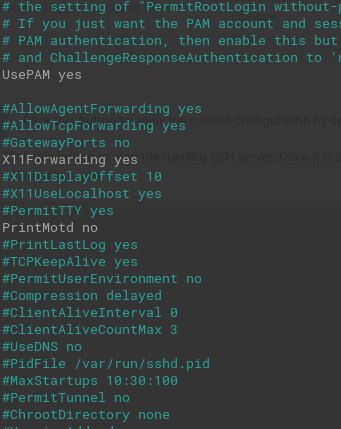


### Uso de aplicaciones gráficas:

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**.

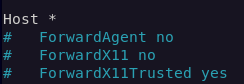
En el archivo **/etc/sshd\_config** del servidor deberemos comprobar que está la siguiente línea o añadirla:

* **X11Forwarding yes**



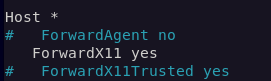
En el archivo **/etc/ssh\_config** del cliente deberemos quitarle la almohadilla y añadirle “yes” al final a la línea:

* **#ForwardX11 no**



Los sustituiríamos por:

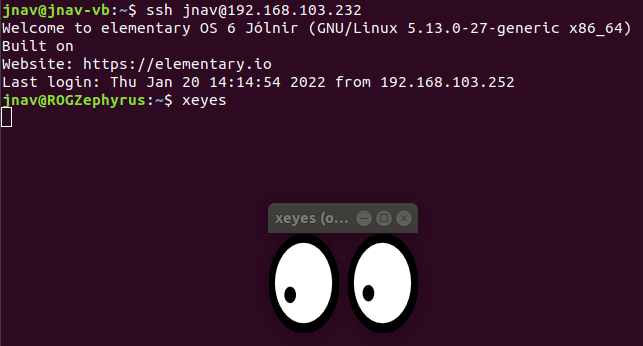
* **ForwardX11 yes**



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

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.



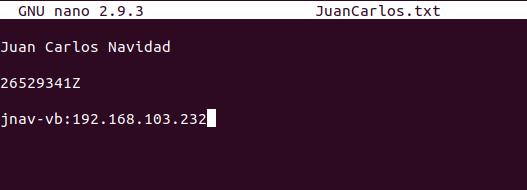
### Traspaso de archivos desde el servidor al cliente:

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 trasferir archivos se utiliza el comando **scp** de la siguiente manera:

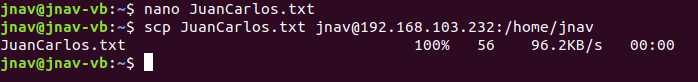
* **scp <archivo> usuario@ip:<destino>**
* **scp usuario@ip:<archivo> <destino>**

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



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

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



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

