

194 SSH

Emilio Cabo Gomis

Sumario

Comprobaciones iniciales.....	3
Configuración del servicio SSH con contraseña.....	3
Inicio de sesión en la máquina cliente mediante contraseña.....	5
Configuración del servicio SSH sin contraseña (clave público-privada).....	6
Generación de claves.....	6
Compartir las claves.....	7
Inicio de sesión en la máquina cliente mediante clave público-privada.....	9
Copiado de archivos mediante scp.....	9
Como proporcionar ssh la contraseña de manera no interactiva.....	10
Automatización del setup de SSH.....	13
Bibliografía.....	15

Comprobaciones iniciales

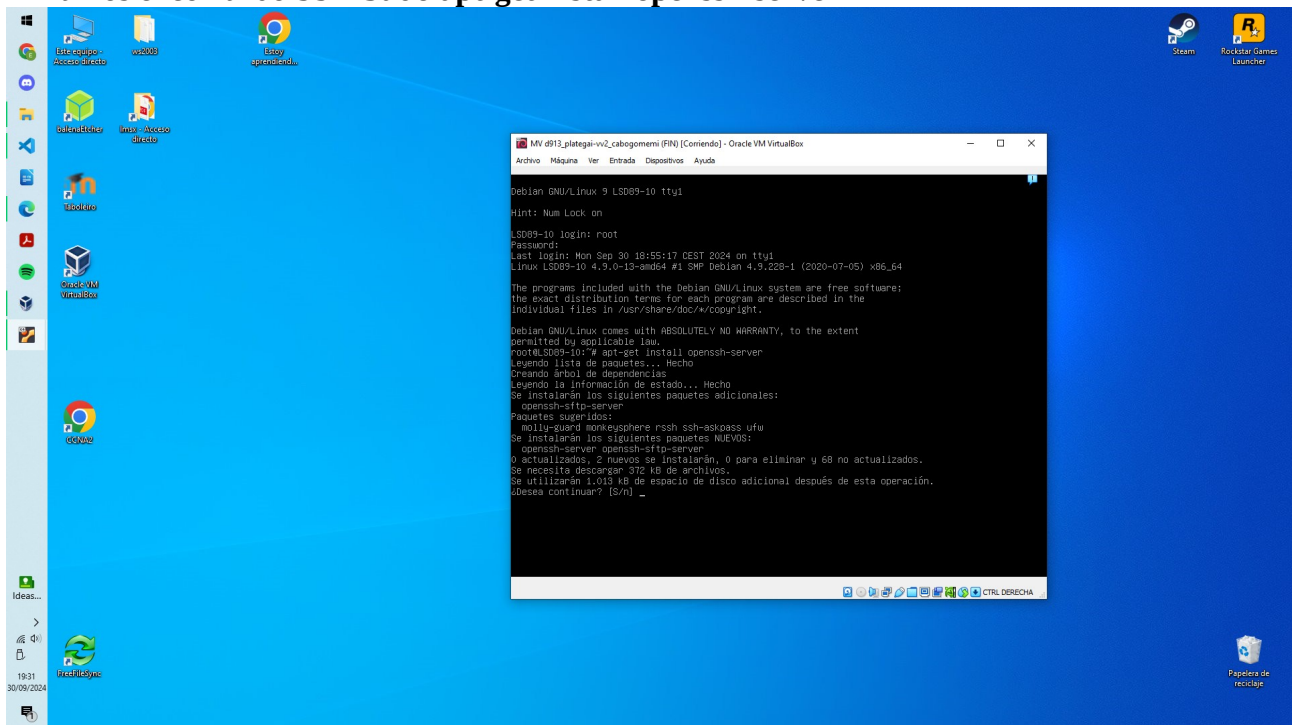
Deberemos tener dos máquinas, una cliente y una servidor en la misma red. O habilitando el reenvío de puertos.

El puerto SSH es el puerto 22.

Configuración del servicio SSH con contraseña

Primero comenzamos instalando el servicio de SSH.

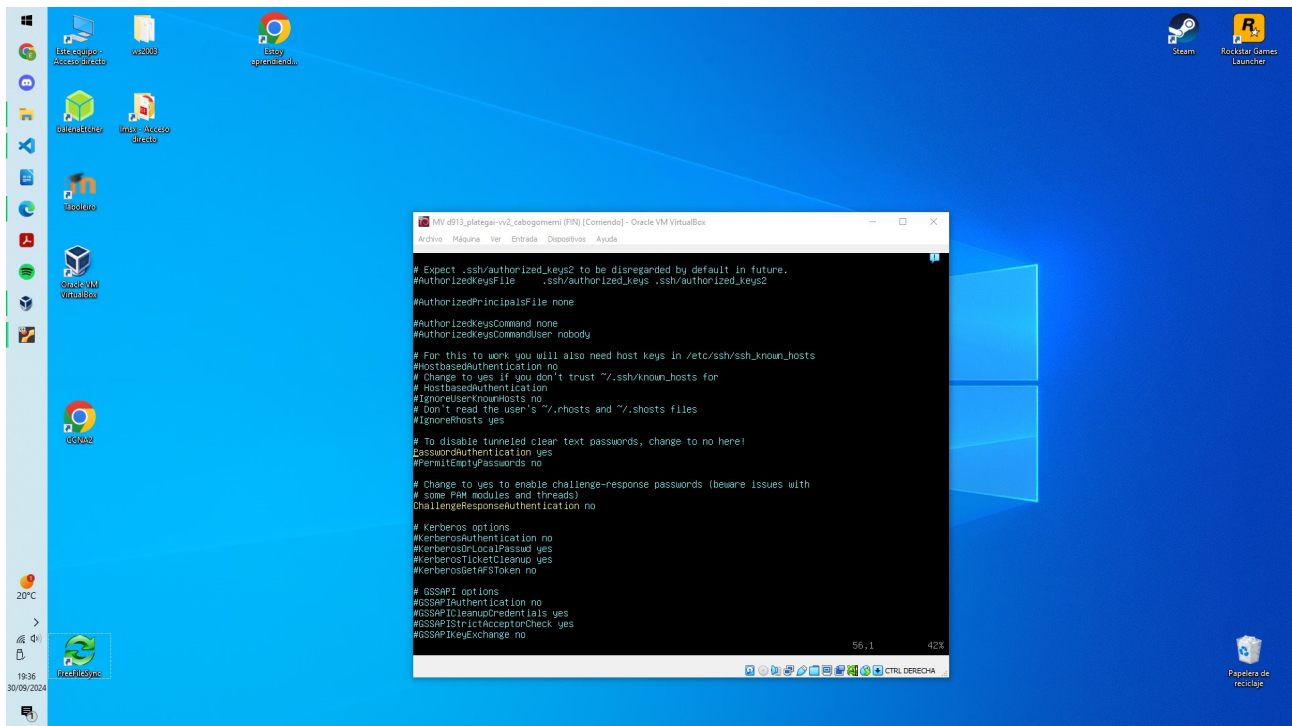
Emitimos el comando SSH **Sudo apt-get install openssh-server**



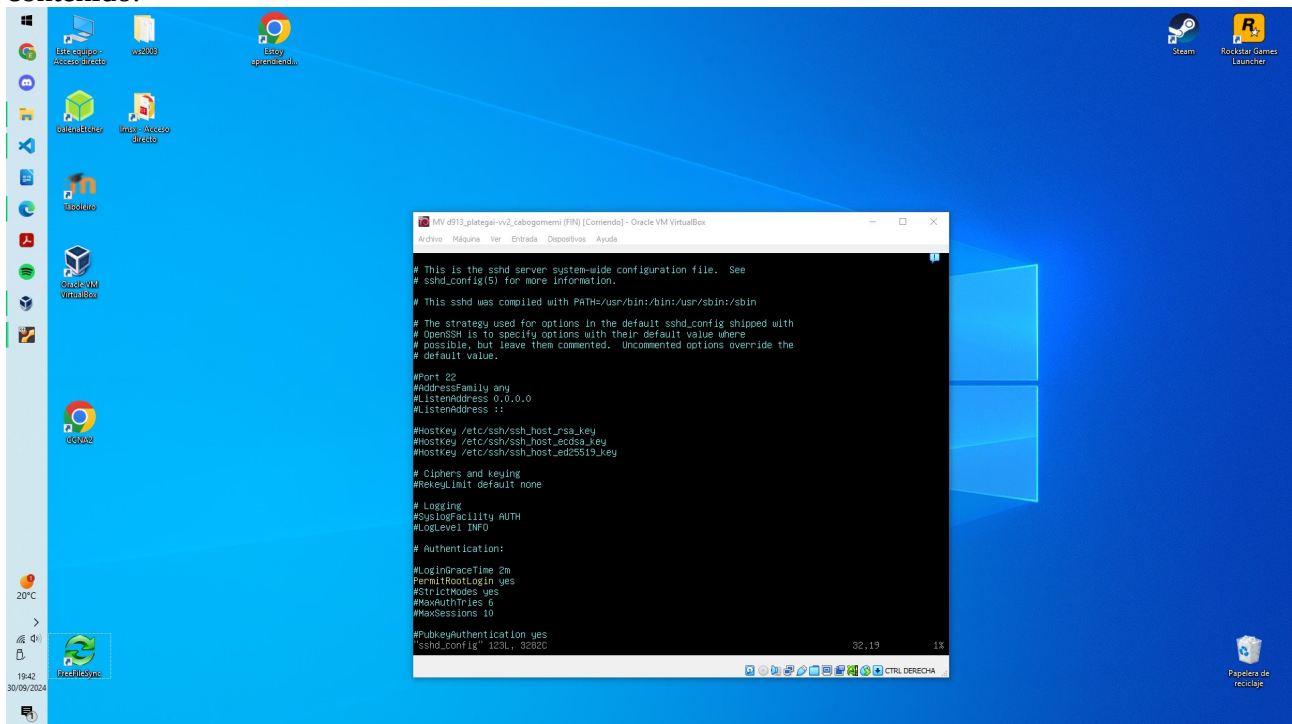
Realizamos una copia de seguridad del archivo de configuración antes de modificarlo de ninguna manera.

sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak

Entramos al archivo de configuración.
Descomentamos la línea #PasswordAuthentication yes es un ajuste que cambiaremos posteriormente.

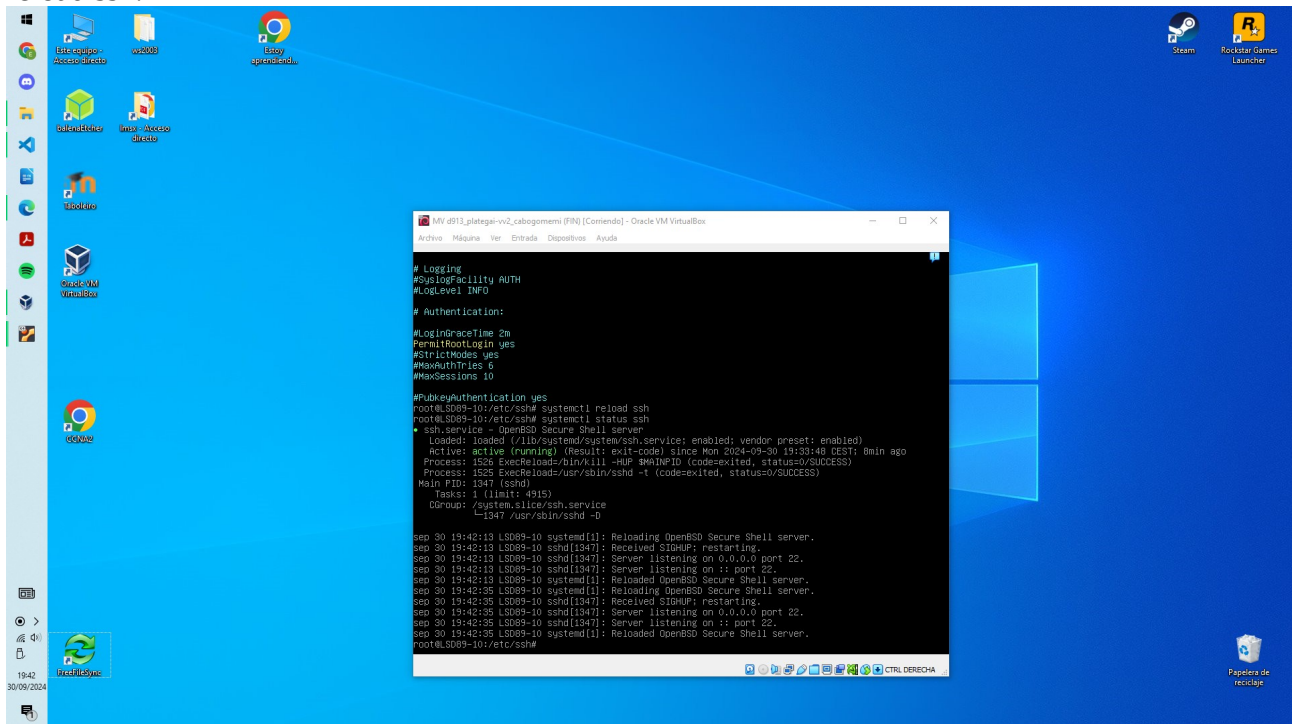


Editamos la línea que contiene #PermitRootLogin prohibitpassword para que tenga el siguiente contenido.



Guardamos el contenido.

Hacemos que la nueva configuración tome efecto en el servicio mediante el comando `systemctl reload ssh`.

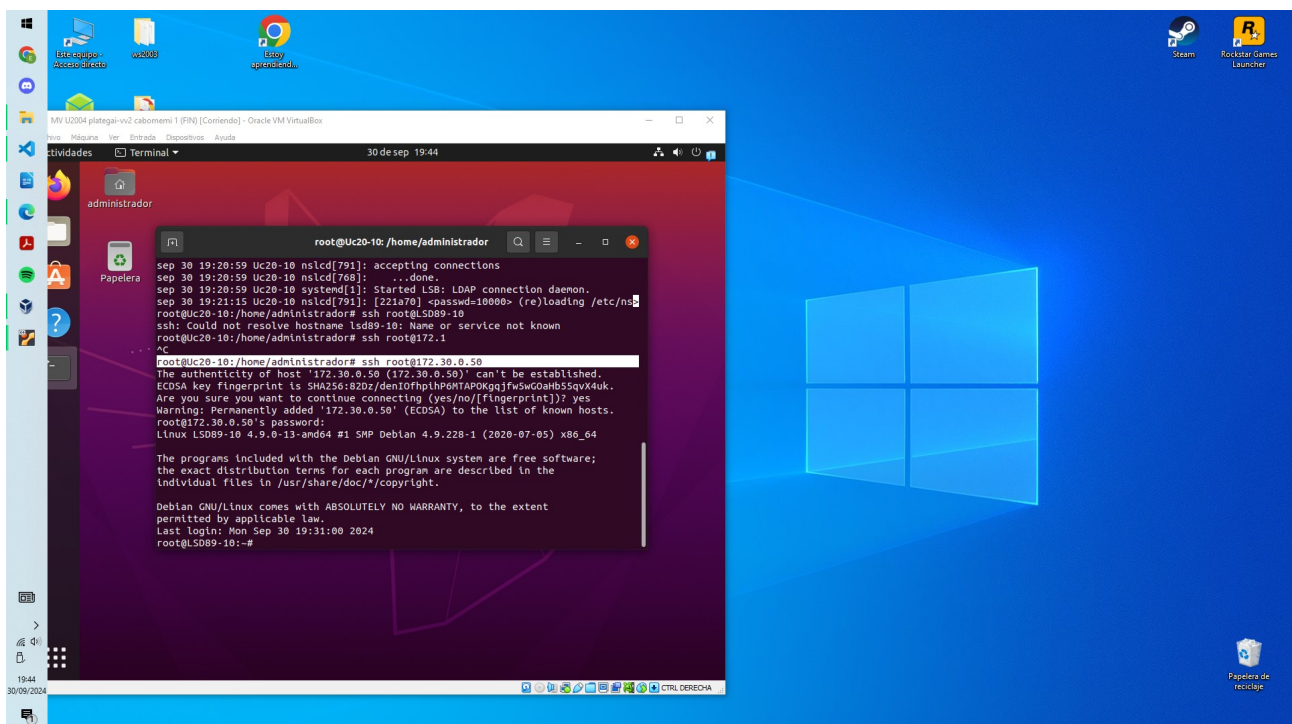


Inicio de sesión en la máquina cliente mediante contraseña

Iniciamos la conexión desde la máquina cliente.

Ssh <nombredeusuario>@<ip//nombreservidor>

Se nos dirá que introduzcamos la contraseña.



Configuración del servicio SSH sin contraseña (clave público-privada)

Generación de claves

Comenzamos generando la clave público y privada en nuestra máquina cliente mediante el comando.

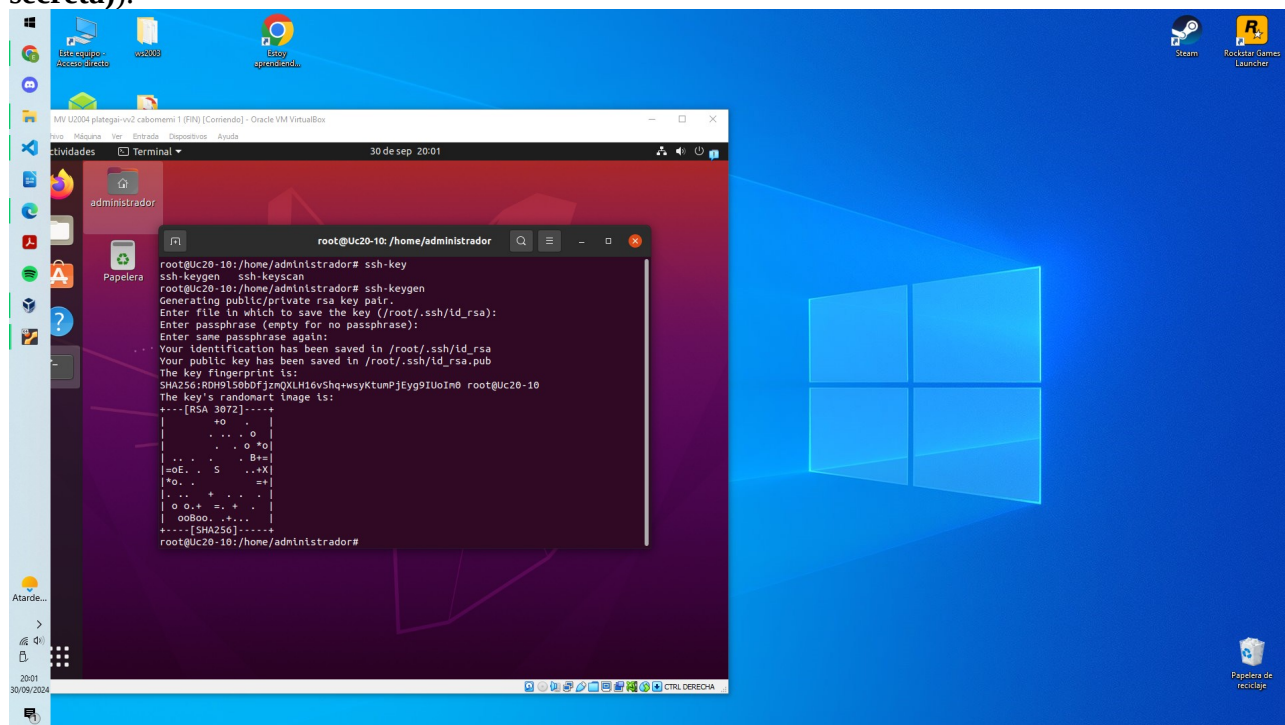
ssh-keygen

Se nos pedirá el directorio dónde lo queremos guardar. Por defecto este será el `/root/.ssh/id_rsa`

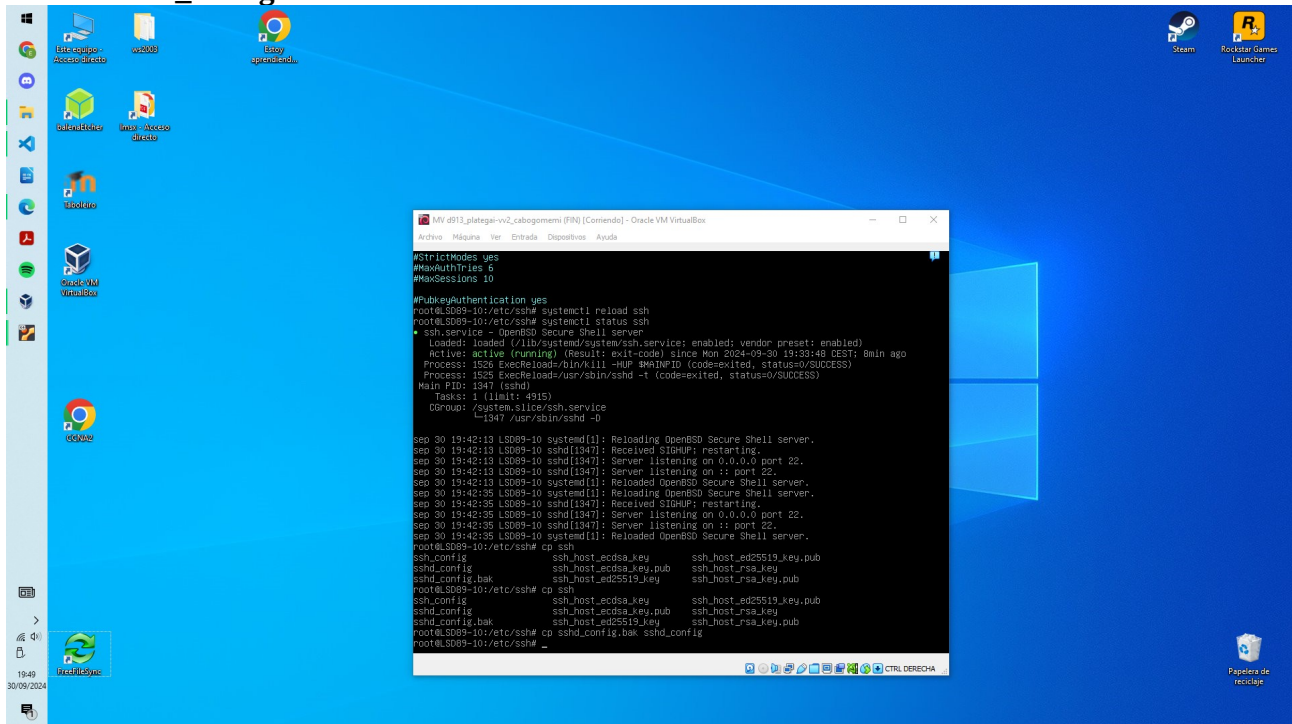
Recomiendo no tocar este ajuste dado que de otra manera tendremos que cambiar la ubicación en la que el servidor tendrá que ir a buscar las claves.

Esto genera 2 archivos `~/.ssh/id_rsa` y `~/.ssh/id_rsa.pub`. El primero contiene la clave privada, que debemos mantener segura y no compartir. El segundo archivo contiene la clave pública, que compartiremos con los servidores SSH a los que querramos conectarnos.

Y una contraseña desde la cual se generará la clave pública. (Recomiendo `abc123`.(clave súper secreta)).

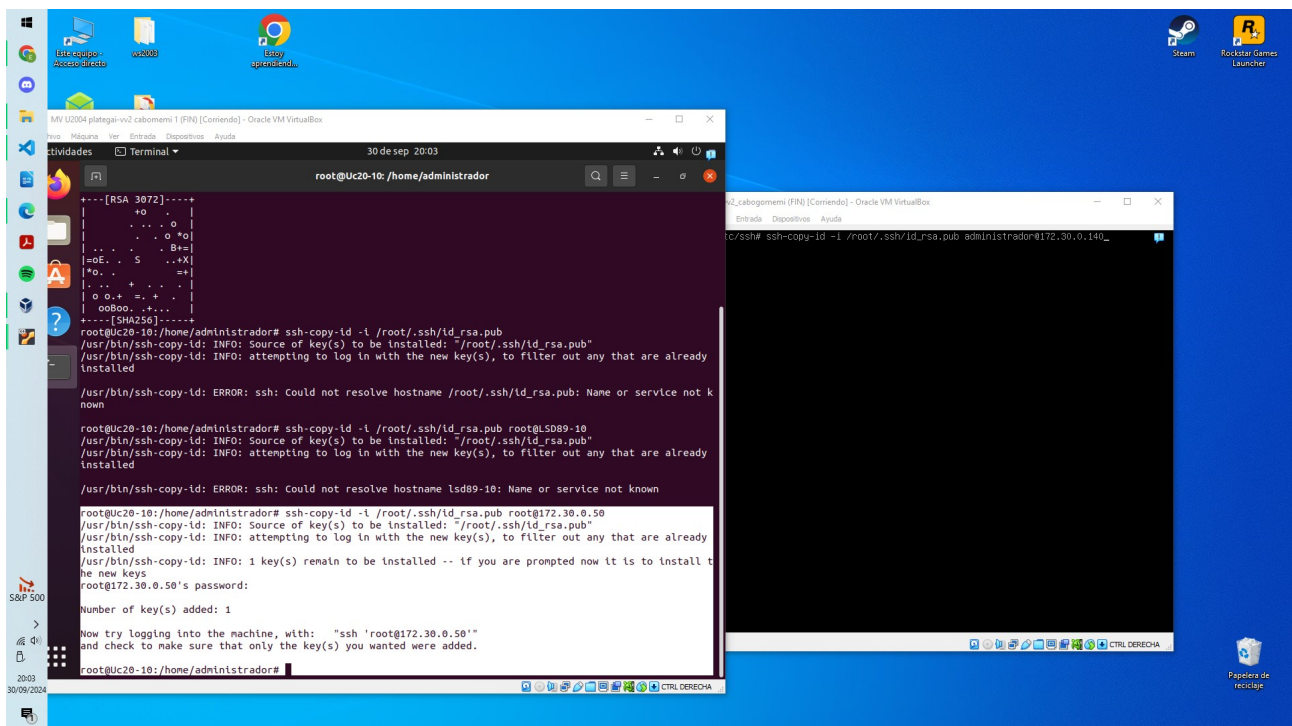


Revertimos la configuración mediante el comando `sudo cp /etc/ssh/sshd_config.bak /etc/ssh/sshd_config`



Compartir las claves

Compartimos las claves mediante el comando `ssh-copy-id -i /root/.ssh/id_rsa.pub <username>@<ip//nombreServidor>`

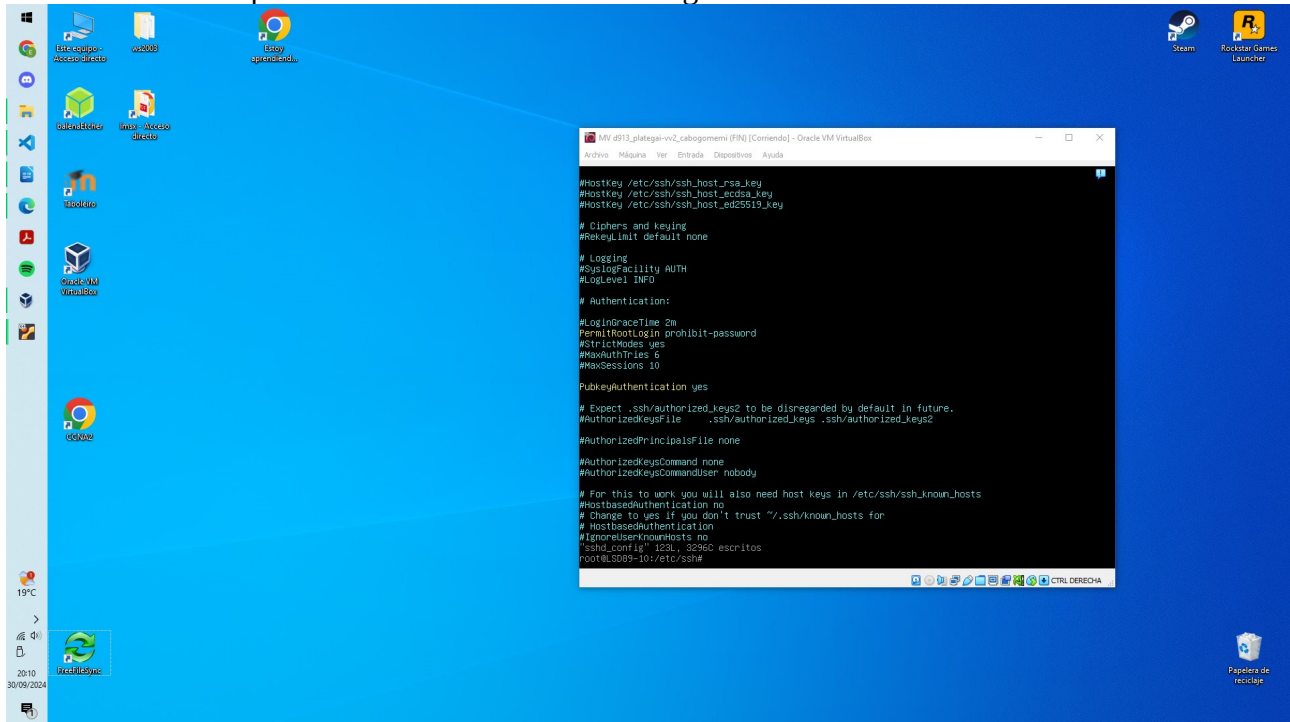


Clave compartida con éxito

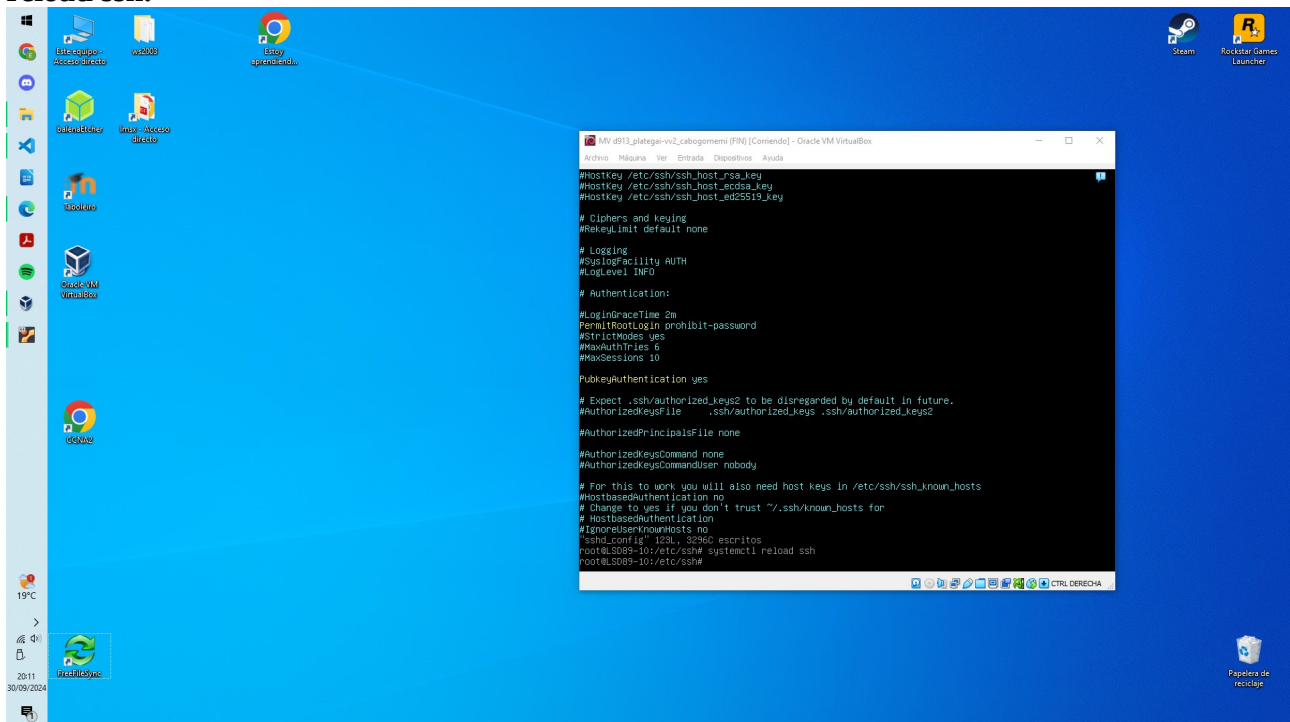
Antes de intentar iniciar sesión como nos dice el output del comando. Tendremos que modificar algunos aspectos del archivo de configuración.

Activamos la autenticación mediante autenticación con clave público-privada que es más segura, que la autenticación mediante contraseña.

Permitimos que podamos iniciar sesión con el usuario root pero no mediante contraseña. Estos son los dos elementos que he tocado del archivo de configuración.

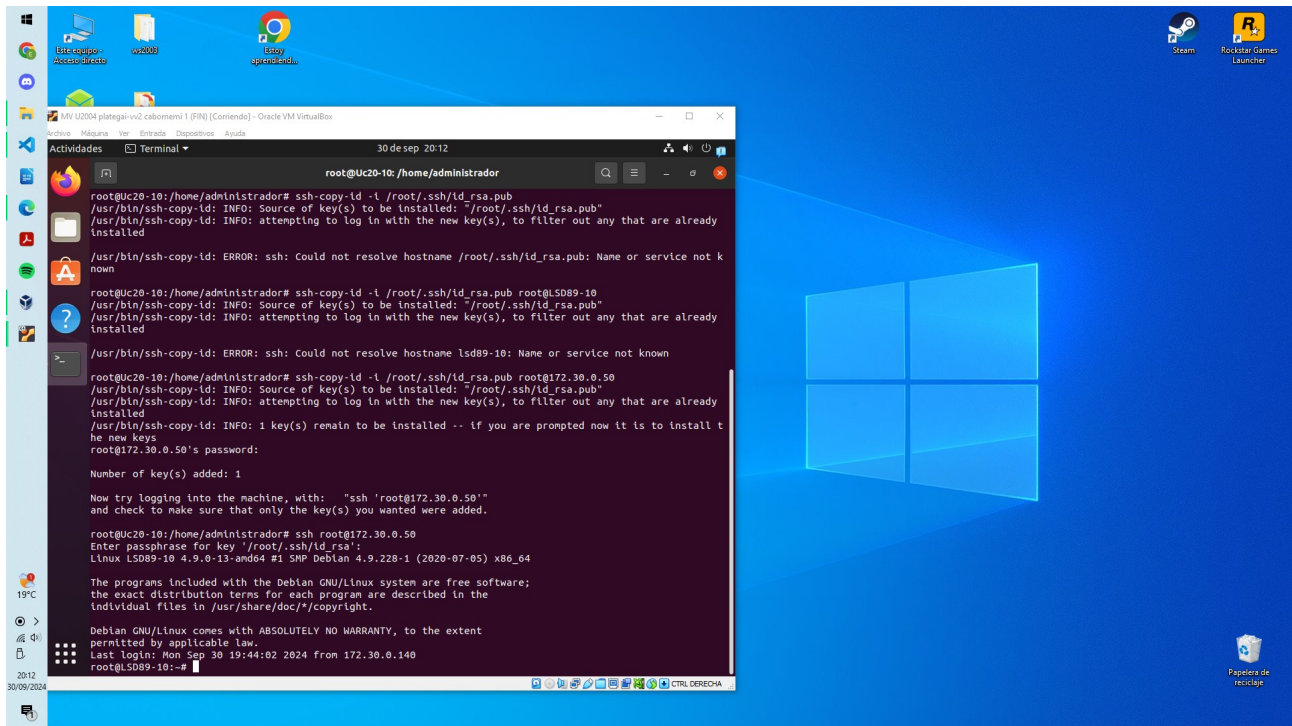


Hacemos que la nueva configuración tome efecto en el servicio mediante el comando **systemctl reload ssh.**



Inicio de sesión en la máquina cliente mediante clave público-privada

Nos pedirá la *passphrase*.



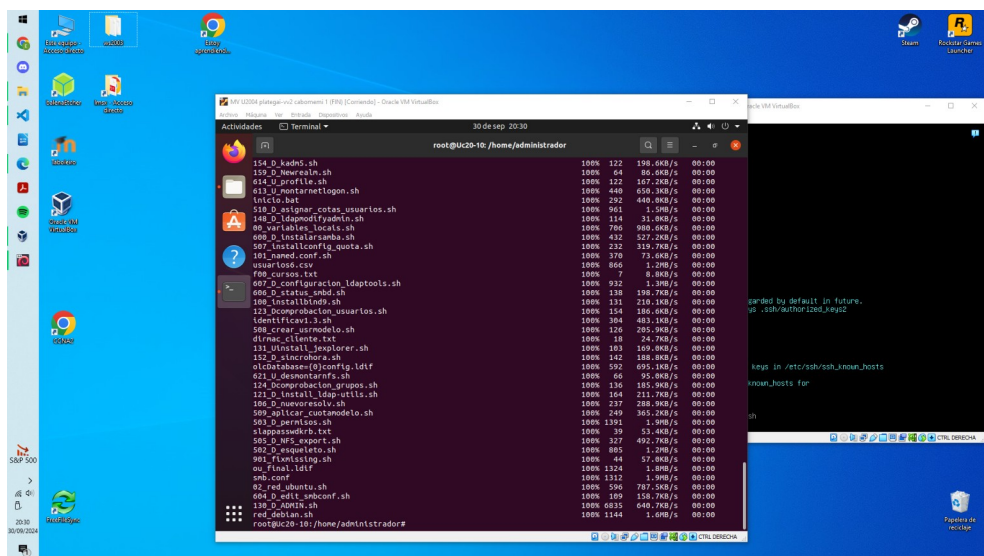
Inicio de sesión exitoso mediante clave público-privada.

Copiado de archivos mediante scp

Para copiar todos los contenidos de una carpeta, por ejemplo la carpeta que contiene mis herramientas linux, ejecutamos el siguiente comando.

Es importante que pongamos la opción **-r** dado que copiará recursivamente.

scp -r ./herramientas_linux root@172.30.0.50:~/herramientas_linux



Como proporcionar ssh la contraseña de manera no interactiva

Sencillo, mediante **sshpass**.

REQUISITOS:

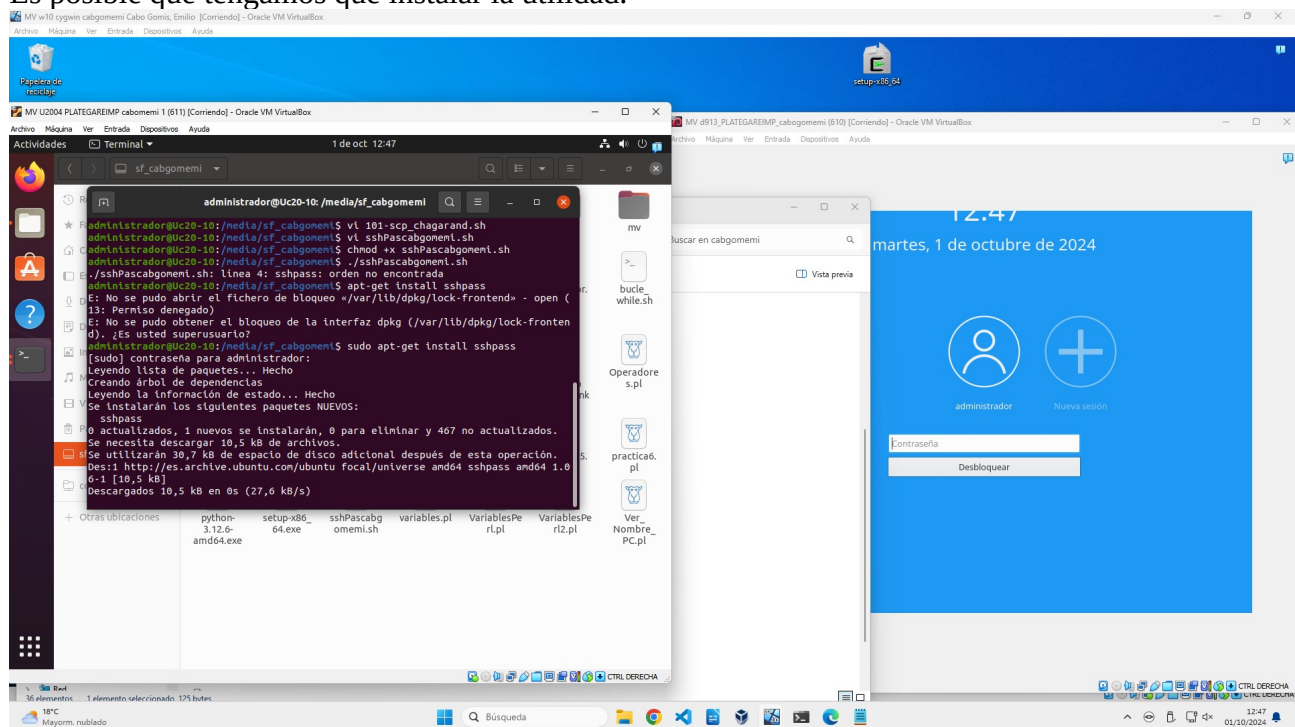
Tener el servidor ssh configurado para aceptar contraseñas y no claves.

Necesitaremos editar el siguiente documento las siguientes líneas **/etc/ssh/sshd_config**

- Descomentamos la línea **#PasswordAuthentication** yes
- Descomentamos la línea que contiene **#PermitRootLogin prohibitpassword** y la editamos para que tenga **PermitRootLogin** yes

Reloadeamos el servicio.

Es posible que tengamos que instalar la utilidad.



Esta es la sintaxis **sshpass -p \$PASS ssh \$usuario@\$HOST**

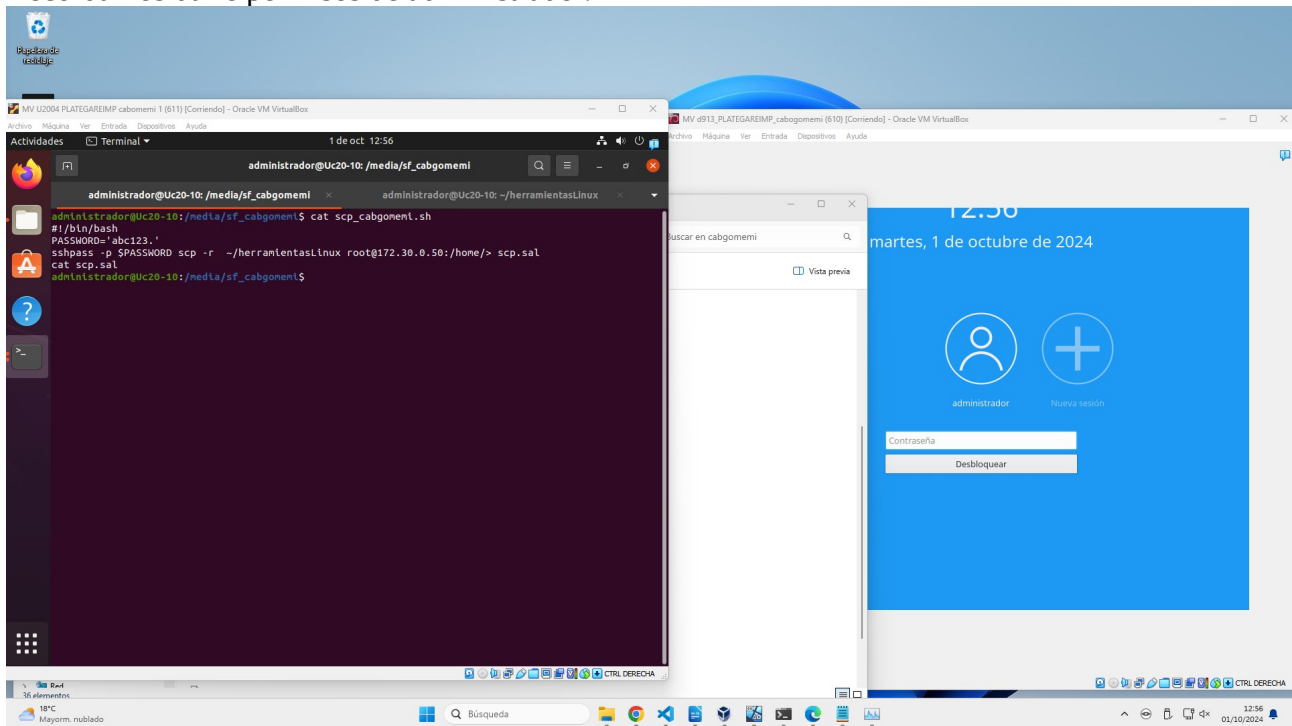
(En este caso emplea variables para introducir el contenido pero no es necesario(aunque sí útil)).

Introducimos esto en un script.

Copiado de archivos automático

Ahora creamos un script con el siguiente contenido.

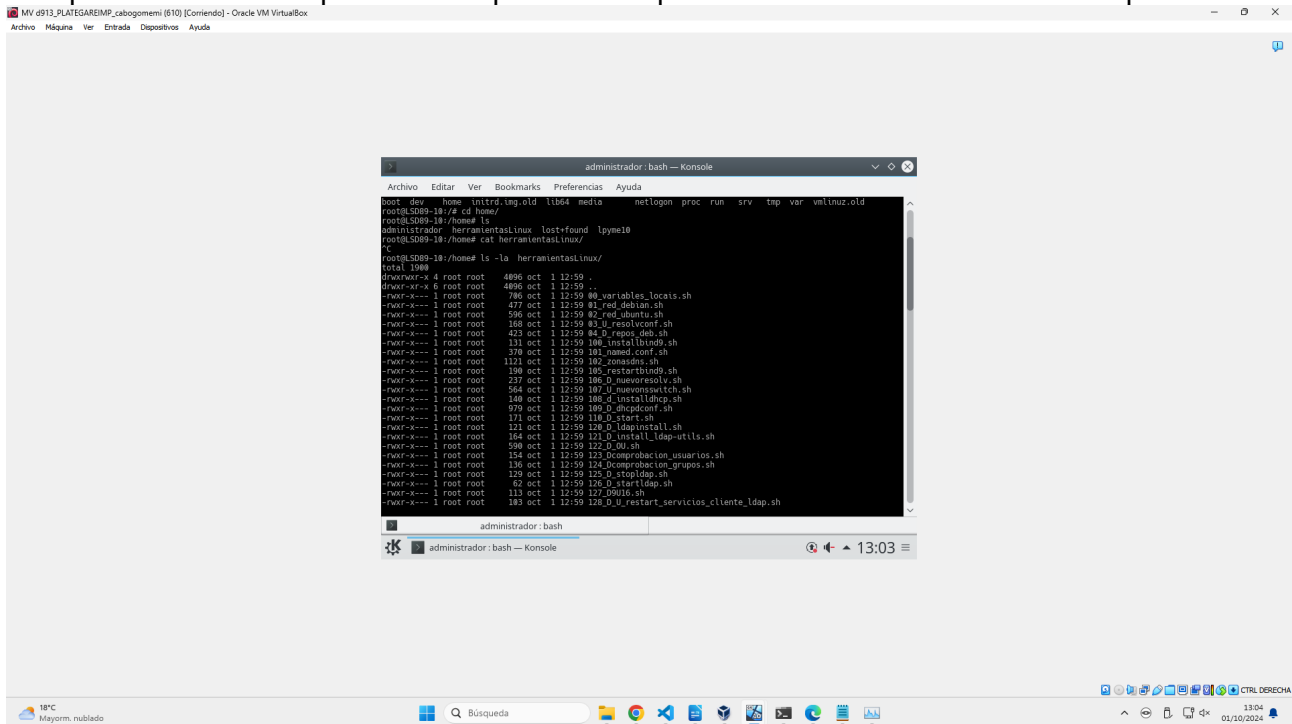
Recordamos darle permisos de administrador.



En este script vamos a copiar las herramientas que tengo desde la máquina ubuntu hacia el servidor para poder ejecutarlas

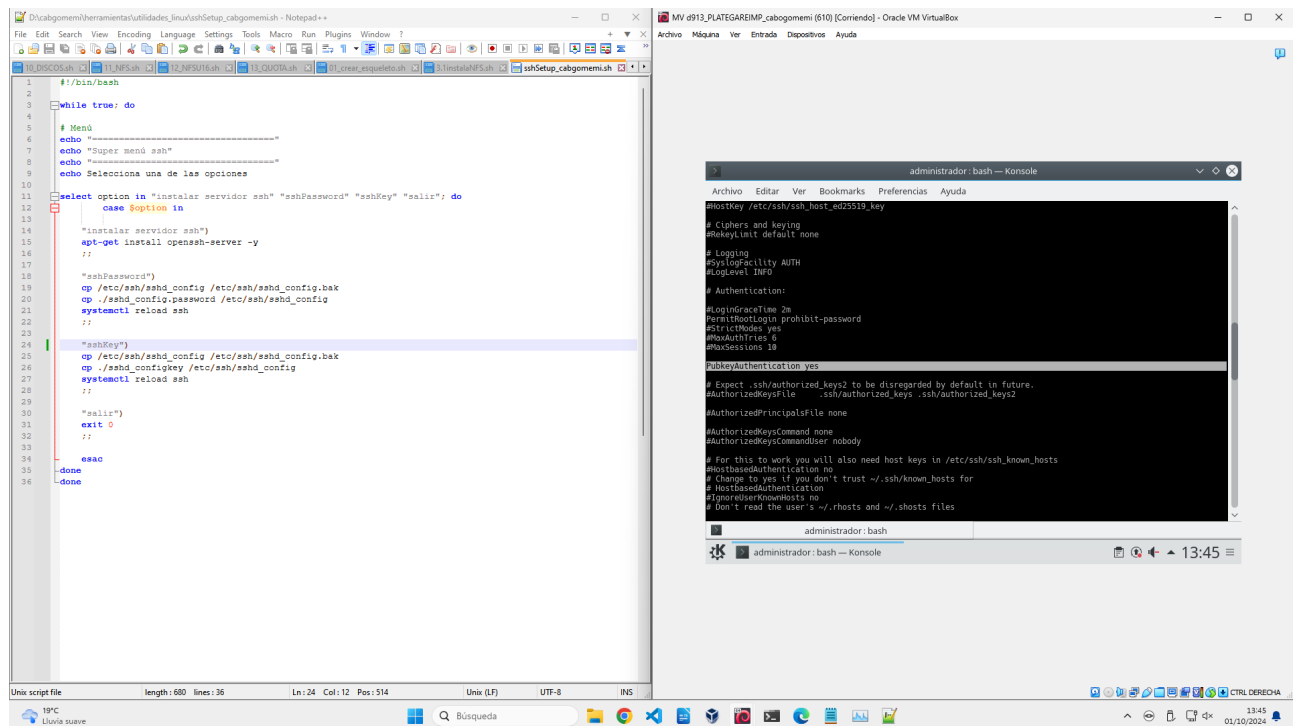
Ejecutamos el script.

Comprobamos en la máquina debian que se han copiado los archivos en el directorio especificado.



Automatización del setup de SSH

Creamos un script con el siguiente contenido.



The screenshot shows a Notepad++ editor on the left with a Bash script named `sshSetup_cabogemini.sh`. The script is a menu-driven program that allows the user to install the SSH server, configure it with a password or a key, and reload the service. The script uses `apt-get` to install `openssh-server`, `cp` to copy configuration files, and `systemctl` to reload the service. The terminal window on the right shows the output of the script, displaying the SSH configuration options and the user's selection to install the server and configure it with a key.

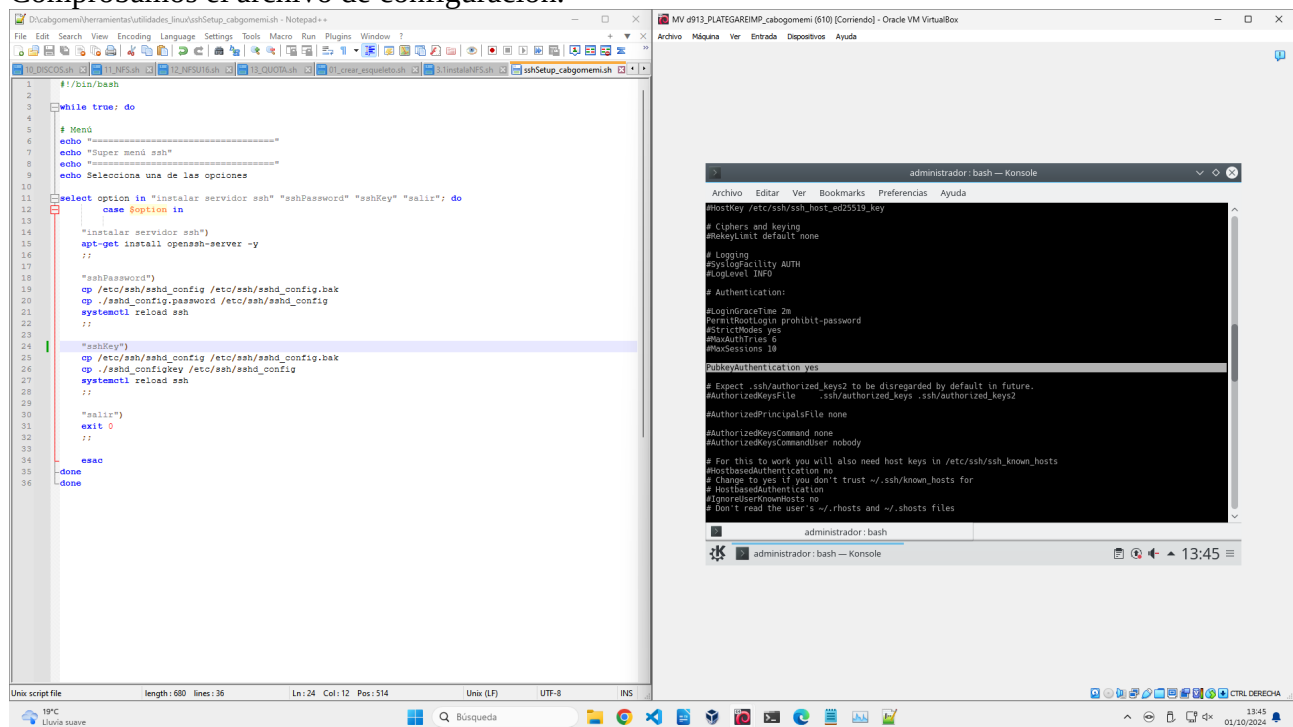
```
#!/bin/bash

while true; do
    # Menú
    echo "=====
    echo "Super menú ssh"
    echo "=====
    echo "Selecciona una de las opciones

    select option in "instalar servidor ssh" "sshPassword" "sshKey" "salir"; do
        case $option in
            "instalar servidor ssh")
                apt-get install openssh-server -y
                ;;
            "sshPassword")
                cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
                cp ./sshd_config.password /etc/ssh/sshd_config
                systemctl reload ssh
                ;;
            "sshKey")
                cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
                cp ./sshd_configkey /etc/ssh/sshd_config
                systemctl reload ssh
                ;;
            "salir")
                exit 0
                ;;
        esac
    done
done
```

```
administrador: bash -- Konsole
HostKey /etc/ssh/ssh_host_ed25519_key
# Ciphers and keying
#RekeyLimit default none
# Logging
#LogLevel INFO
# Authentication:
#loginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
PubkeyAuthentication yes
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
#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
```

Ejecutamos la opción para configurar el inicio de sesión mediante key.
Comprobamos el archivo de configuración.



The screenshot shows a Notepad++ editor on the left with the same Bash script as before. The terminal window on the right shows the output of the script, displaying the SSH configuration options and the user's selection to install the server and configure it with a key. The terminal output is identical to the previous screenshot, showing the SSH configuration options and the user's selection to install the server and configure it with a key.

```
#!/bin/bash

while true; do
    # Menú
    echo "=====
    echo "Super menú ssh"
    echo "=====
    echo "Selecciona una de las opciones

    select option in "instalar servidor ssh" "sshPassword" "sshKey" "salir"; do
        case $option in
            "instalar servidor ssh")
                apt-get install openssh-server -y
                ;;
            "sshPassword")
                cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
                cp ./sshd_config.password /etc/ssh/sshd_config
                systemctl reload ssh
                ;;
            "sshKey")
                cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
                cp ./sshd_configkey /etc/ssh/sshd_config
                systemctl reload ssh
                ;;
            "salir")
                exit 0
                ;;
        esac
    done
done
```

```
administrador: bash -- Konsole
HostKey /etc/ssh/ssh_host_ed25519_key
# Ciphers and keying
#RekeyLimit default none
# Logging
#LogLevel INFO
# Authentication:
#loginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
PubkeyAuthentication yes
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
#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
```

Ejecutamos la opción para configurar el inicio de sesión mediante contraseña. Comprobamos el archivo de configuración.

The image shows a Windows desktop environment. In the foreground, a Notepad++ window is open, displaying a shell script for installing and configuring OpenSSH server on a Linux system. The script is written in a light blue font on a white background. It starts with a shebang line `#!/bin/bash` and a `while true; do` loop. Inside the loop, it prompts the user to install the OpenSSH server using `apt-get` and then prompts for a password and a key. The script then uses `systemctl` to reload the SSH service and restarts it. Finally, it prompts for a user name and a password, and then exits the loop. The script is saved as `sshSetup\_cabgonemini.sh`. In the background, a VMware Workstation window is open, showing a virtual machine named 'Ubuntu-20.04'. The virtual machine is running, and a terminal window is open, displaying the output of the script execution. The terminal output shows the installation of the OpenSSH server and the configuration of the SSH service. The terminal window is titled 'administrador: bash - Konsole'. The desktop background is a dark blue gradient with a white grid pattern. The taskbar at the bottom shows various application icons, including the Start button, a search bar, and several open applications. The system tray in the bottom right corner shows the date and time as '01/10/2022 13:46'.

Bibliografía

José arrate blog

[file:///F:/curso_1/iso/9.Resolucion%20Incidencias%20y%20servicio%20tecnico/900%20SSH%20sin%20contrase%C3%B1as%20\(y%20seguro\)%20-%20Jos%C3%A9%20Arrarte.htm](file:///F:/curso_1/iso/9.Resolucion%20Incidencias%20y%20servicio%20tecnico/900%20SSH%20sin%20contrase%C3%B1as%20(y%20seguro)%20-%20Jos%C3%A9%20Arrarte.htm)

ARCHWIKI

https://wiki.archlinux.org/title/SSH_keys

Wikipedia

https://es.wikipedia.org/wiki/Secure_Shell

Ubuntu

<https://ubuntu.com/server/docs/openssh-server>