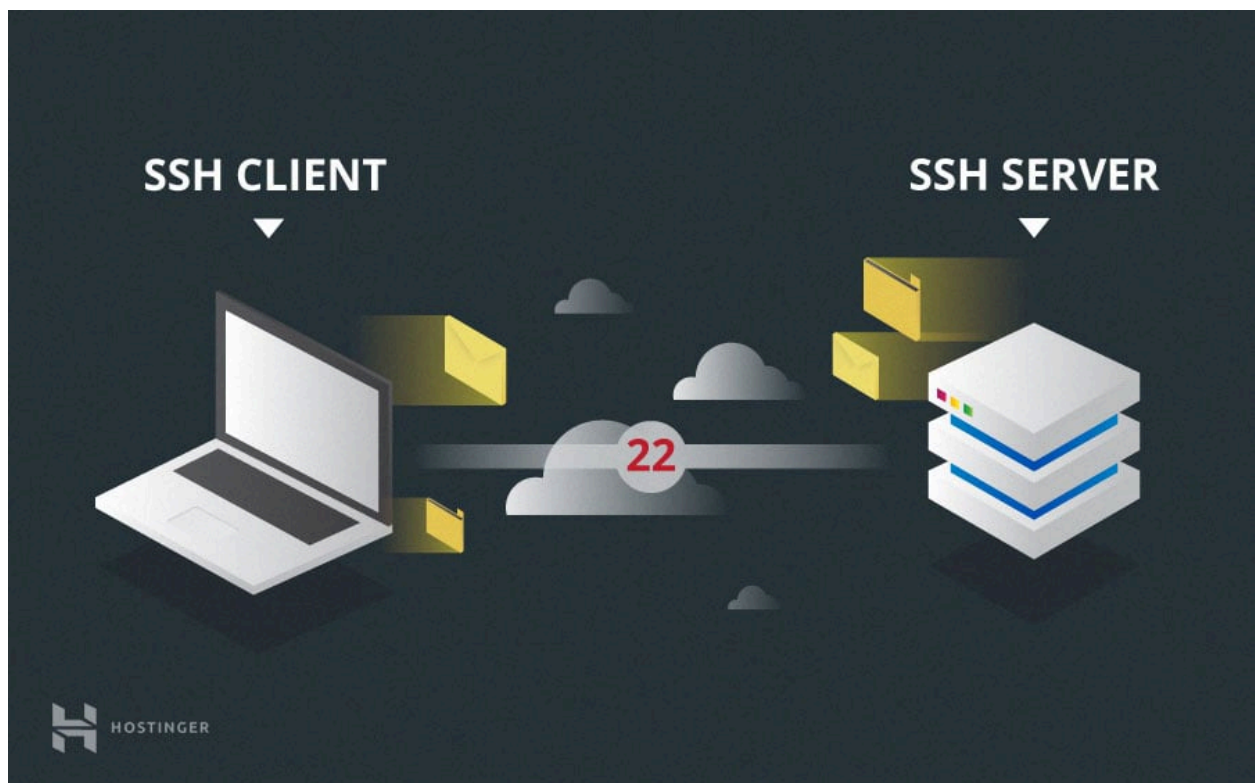




IES Zaidín Vergeles

\*\*\*\*\*

# SSH (Secure Shell)



Jose Almirón López, 16 de Abril de 2024

# Tabla de contenidos

<b>Introducción</b>	<b>2</b>
<b>Instalación del servidor SSH en Linux</b>	<b>2</b>
<b>MobanSSH</b>	<b>6</b>
<b>OpenSSH Windows</b>	<b>8</b>
<b>Bitvise</b>	<b>10</b>
<b>Mobaxterm</b>	<b>13</b>

## Introducción

.....

La administración remota de sistemas informáticos es una habilidad fundamental en el ámbito de la informática y las tecnologías de la información. La práctica de configuración de un servidor SSH (Secure Shell) y el uso de herramientas asociadas constituye un paso crucial en la formación de profesionales de TI, permitiendo el acceso seguro y la gestión eficiente de sistemas a través de redes.

## Instalación del servidor SSH en Linux

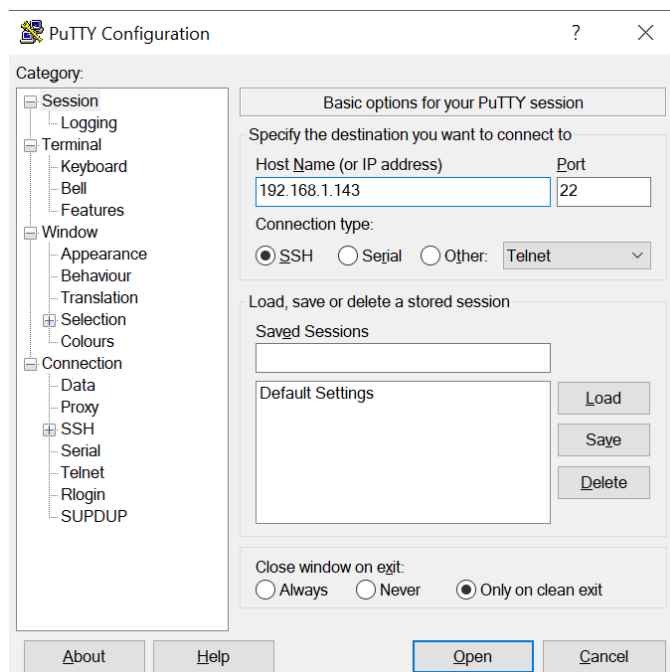
Para instalar un servidor SSH en Linux, necesitaremos instalar los paquetes de OpenSSH, que incluyen tanto el cliente como el servidor de SSH. En este escenario, utilizaré una máquina virtual ejecutando Ubuntu Server como servidor, mientras que mi máquina principal, que ejecuta Windows, actuará como cliente.

```
jose@jose:~$ sudo apt install openssh-client openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-client is already the newest version (1:8.9p1-3ubuntu0.7).
openssh-client set to manually installed.
openssh-server is already the newest version (1:8.9p1-3ubuntu0.7).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
jose@jose:~$ _
```

Una vez instalado, procederemos a verificar la dirección IP asignada al servidor, ya que la necesitaremos para establecer la conexión, junto con las credenciales de usuario correspondientes

```
jose@jose:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:fc:9a:36 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.143/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 86101sec preferred_lft 86101sec
    inet6 2a0c:5a82:2104:1c00:a00:27ff:fe9c:9a36/64 scope global mngtmpaddr noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe9c:9a36/64 scope link
        valid_lft forever preferred_lft forever
jose@jose:~$ _
```

Una vez tengamos la dirección IP, abrimos PuTTY, el programa que utilizaremos como cliente para establecer las conexiones. En el campo '**Host Name**', introducimos la IP del servidor y el puerto se establece automáticamente en 22. Una vez completado esto, hacemos clic en 'Open' para iniciar la conexión



A continuación, se abrirá una ventana similar a una terminal en la que se nos solicitará el nombre de usuario y la contraseña. Si se ingresan correctamente, tendremos acceso al servidor desde PuTTY

```
jose@jose: ~  
login as: jose  
jose@192.168.1.143's password:  
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-102-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro  
  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your  
Internet connection or proxy settings  
  
Last login: Thu Apr 18 16:08:56 2024  
jose@jose:~$
```

Podemos mejorar la seguridad de nuestro servidor SSH realizando ajustes en su configuración. Para ello, nos dirigimos al archivo de configuración ubicado en `'/etc/ssh/sshd_config'`. En mi caso, lo abriré con el editor de texto 'nano'. Aquí encontraremos una variedad de configuraciones que podemos ajustar. En mi caso, uno de los primeros cambios que haré será modificar el puerto predeterminado de SSH, cambiándolo de 22 a 122

```
GNU nano 6.2 /etc/ssh/sshd_config *
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/

# The strategy used for options in the default sshd_config shipped
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override
# default value.

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

Port 122
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Vamos a establecer un tiempo máximo de 15 segundos para iniciar sesión, deshabilitamos el acceso como usuario root y limitaremos el número máximo de intentos de inicio de sesión a 3

```
GNU nano 6.2 /etc/ssh/sshd_config *
Include /etc/ssh/sshd_config.d/*.conf

Port 122
#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

# Authentication:

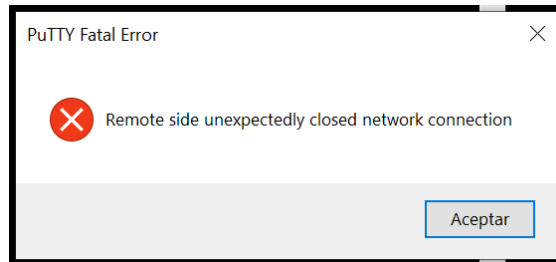
LoginGraceTime 15
PermitRootLogin no
#StrictModes yes
MaxAuthTries 3
#MaxSessions 10
```

.....

Para aplicar los cambios, reiniciamos el servicio SSH.

```
jose@jose:~$ sudo /etc/init.d/ssh restart
Restarting ssh (via systemctl): ssh.service.
jose@jose:~$ _
```

Ahora, al intentar establecer una conexión a través de PuTTY, si no ingresamos rápidamente nuestras credenciales de inicio de sesión, recibiremos un error debido al límite de tiempo de 15 segundos que hemos establecido



Para aumentar aún más la seguridad, podemos crear un usuario específico para SSH. He creado este usuario utilizando el comando 'useradd' y luego he añadido la línea '**AllowUsers**' en el archivo de configuración SSH, especificando el nuevo usuario que acabo de crear."

```
GNU nano 6.2 /etc/ssh/sshd_config *
# possible, but leave them commented. Uncommented options override the
# default value.

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

Port 122
#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

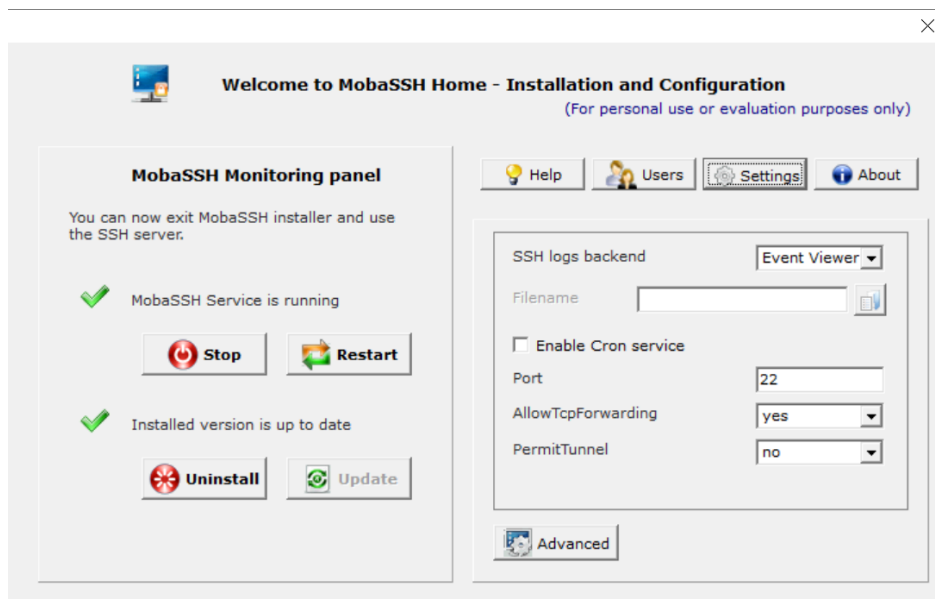
# Logging
#SyslogFacility AUTH
#LogLevel INFO
AllowUsers security_
```

Finalmente, solo será posible realizar la conexión por SSH utilizando el nuevo usuario específico que hemos creado. Si intentamos utilizar el usuario anterior, en este caso 'jose', recibiremos un mensaje de 'Acceso denegado'.

```
security@jose: ~  
login as: security  
security@192.168.1.143's password:  
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-102-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro  
  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your  
Internet connection or proxy settings  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
security@jose:~$
```

## MobanSSH

La instalación de MobaSSH es un proceso bastante sencillo, al igual que su interfaz. Una vez instalado, veremos que el servicio está en ejecución.



En este caso, estoy utilizando una máquina Windows que está actuando como servidor, por lo que procedo a verificar la dirección IP asignada

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.3803]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\user>ipconfig

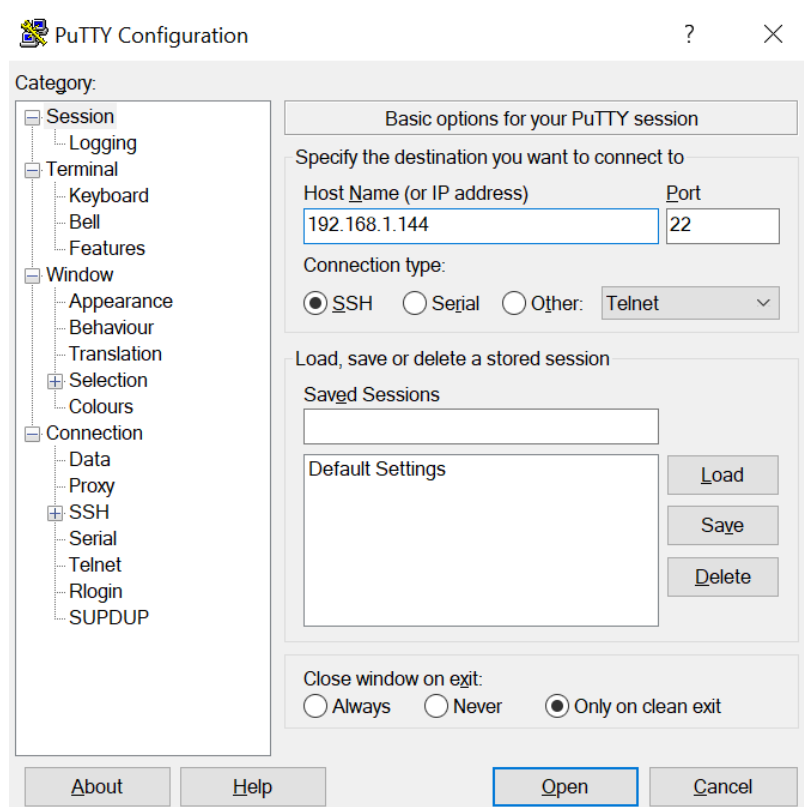
Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión. . . :
    Dirección IPv6 . . . . . : 2a0c:5a82:2104:1c00::2
    Vínculo: dirección IPv6 local. . . : fe80::5f1e:bea4:4ed2:2252%6
    Dirección IPv4. . . . . : 192.168.1.144
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 192.168.1.1

C:\Users\user>
```

Al igual que en el caso anterior, utilizar PuTTY para establecer la conexión.





Nos solicitará el nombre de usuario y la contraseña, y una vez autenticados, tendremos acceso al servidor a través de MobaSSH.

```
DESKTOP-SDFU56M
login as: user
user@192.168.1.144's password:

-----
                        MobaSSH Home v1.60
                (SSH server for Win32 based on Cygwin/OpenSSH)

Important
- Your computer drives are accessible through the /cygdrive directory
- Network shares are accessible by typing //<remote_computer>
- The Windows registry is browsable through the /registry path

Useful commands
- MobaHwInfo: detailed information about OS and hardware
- MobaSwInfo: installed programs list
- MobaTaskList, MobaKillTask: list/kill Windows tasks
- TCPCapture: Network packets and ports monitoring tool
- scp, sftp: transfer files through the crypted ssh connexion
- nedit, vim: text editors with syntax highlighting
- rsync, wget: sync local directories with network computers

This version is for personal use or evaluation purposes only
For information please visit: http://mobassh.mobatek.net/versions.php
-----

[Thu Apr 18 - 19:31:40] ~
[user.DESKTOP-SDFU56M] $
```

## OpenSSH Windows

En el caso de la característica OpenSSH de Windows, mediante PowerShell, vamos a verificar si este servicio está instalado.

```
PS C:\Windows\system32> Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'

Name : OpenSSH.Client~~~~0.0.1.0
State : Installed

Name : OpenSSH.Server~~~~0.0.1.0
State : NotPresent

PS C:\Windows\system32>
```

Observó que el servidor no está instalado, lo cual es precisamente lo que necesito, por lo tanto, procedo a instalarlo.

```
PS C:\Windows\system32> Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

Path      :
Online    : True
RestartNeeded : False
```

Perfecto, una vez instalado, podemos ejecutar el comando 'Start-Service sshd' para iniciar el servicio


```
PS C:\Windows\system32> Start-Service sshd
PS C:\Windows\system32>
```

Como configuración adicional, podemos establecer que el servicio se inicie de forma automática. Además, es importante confirmar si las reglas de firewall necesarias están correctamente configuradas

```
Administrador: Windows PowerShell

PS C:\Windows\system32> Start-Service sshd
PS C:\Windows\system32> Set-Service -Name sshd -StartupType 'Automatic'
PS C:\Windows\system32> if (!(Get-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -ErrorAction SilentlyContinue | Select-Object Name, Enabled)) {
>> Write-Output "Firewall Rule 'OpenSSH-Server-In-TCP' does not exist, creating it..."
>> New-NetFirewallRule -Name 'OpenSSH-Server-In-TCP' -DisplayName 'OpenSSH Server (sshd)' -Enabled True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22
>> } else {
>> Write-Output "Firewall rule 'OpenSSH-Server-In-TCP' has been created and exists."
>> }
Firewall rule 'OpenSSH-Server-In-TCP' has been created and exists.
PS C:\Windows\system32>
```

Finalmente, procedemos a establecer la conexión utilizando PuTTY.

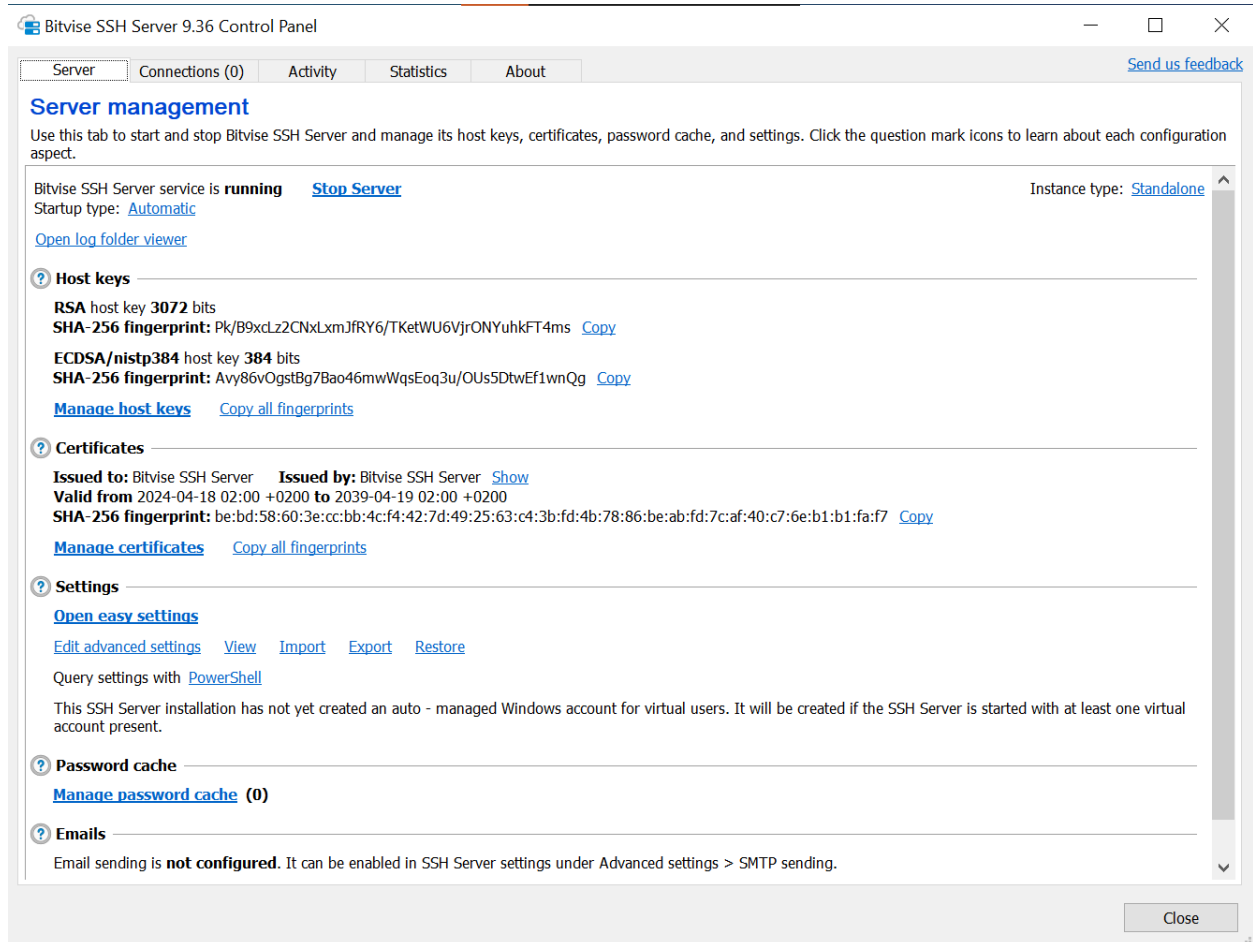
 Administrador: C:\Windows\system32\conhost.exe

```
Microsoft Windows [Versión 10.0.19045.3803]
(c) Microsoft Corporation. Todos los derechos reservados.

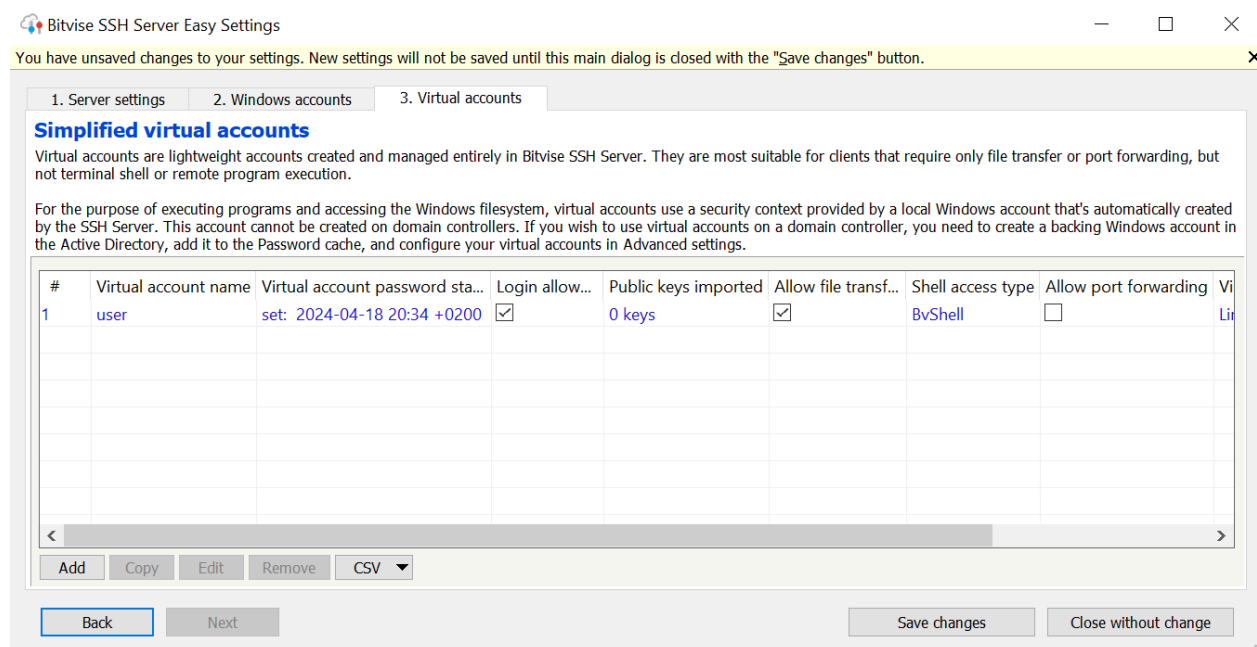
user@DESKTOP-SDFU56M C:\Users\user>
```

# Bitvise

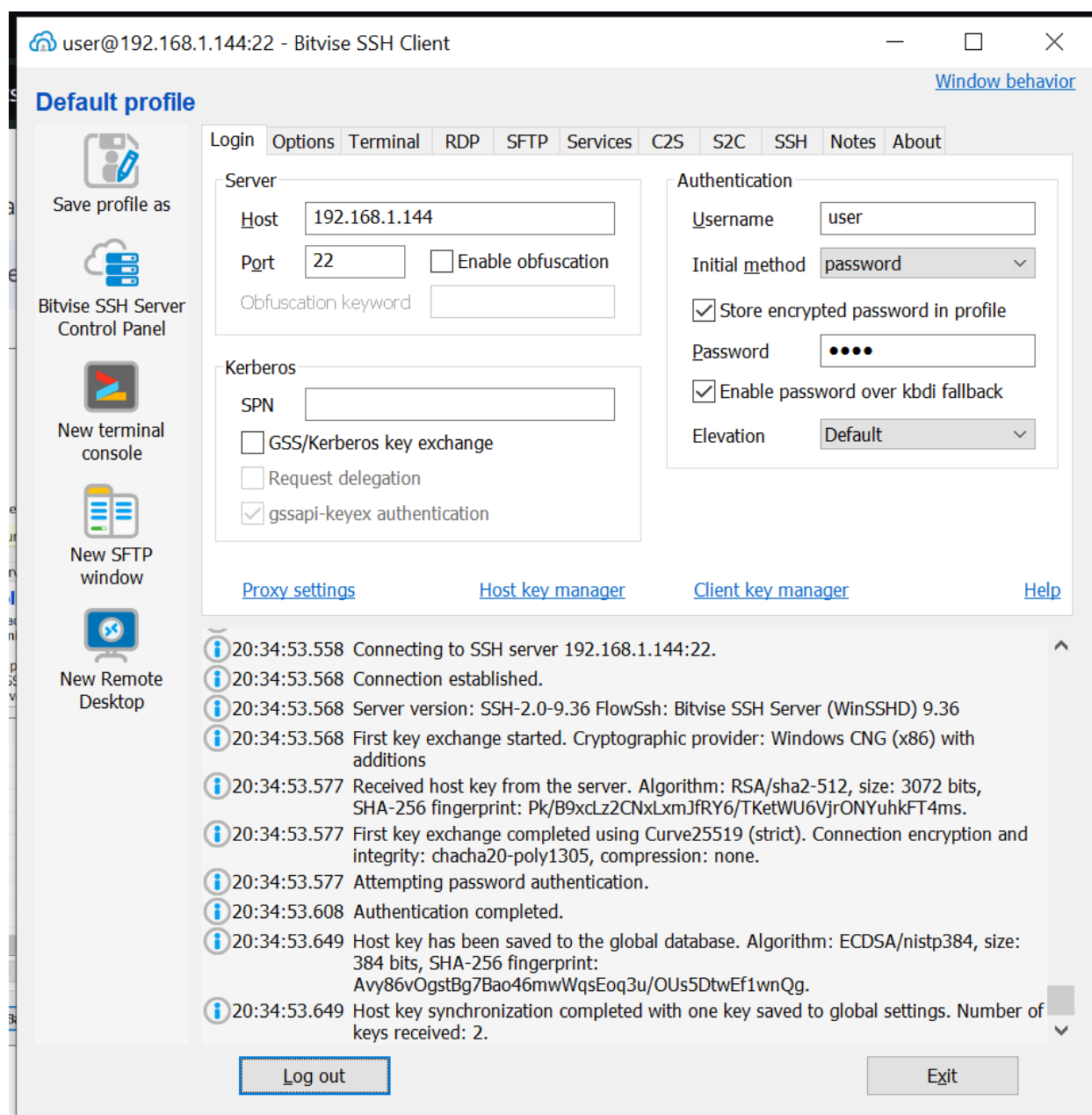
En el caso de Bitvise, contamos con versiones tanto de servidor como de cliente. Para configurar el servidor, una vez instalado, nos dirigimos a 'Settings'.



Una vez aquí, navegamos hasta 'Virtual Account', ya que este servicio requiere la creación de una cuenta de usuario virtual.

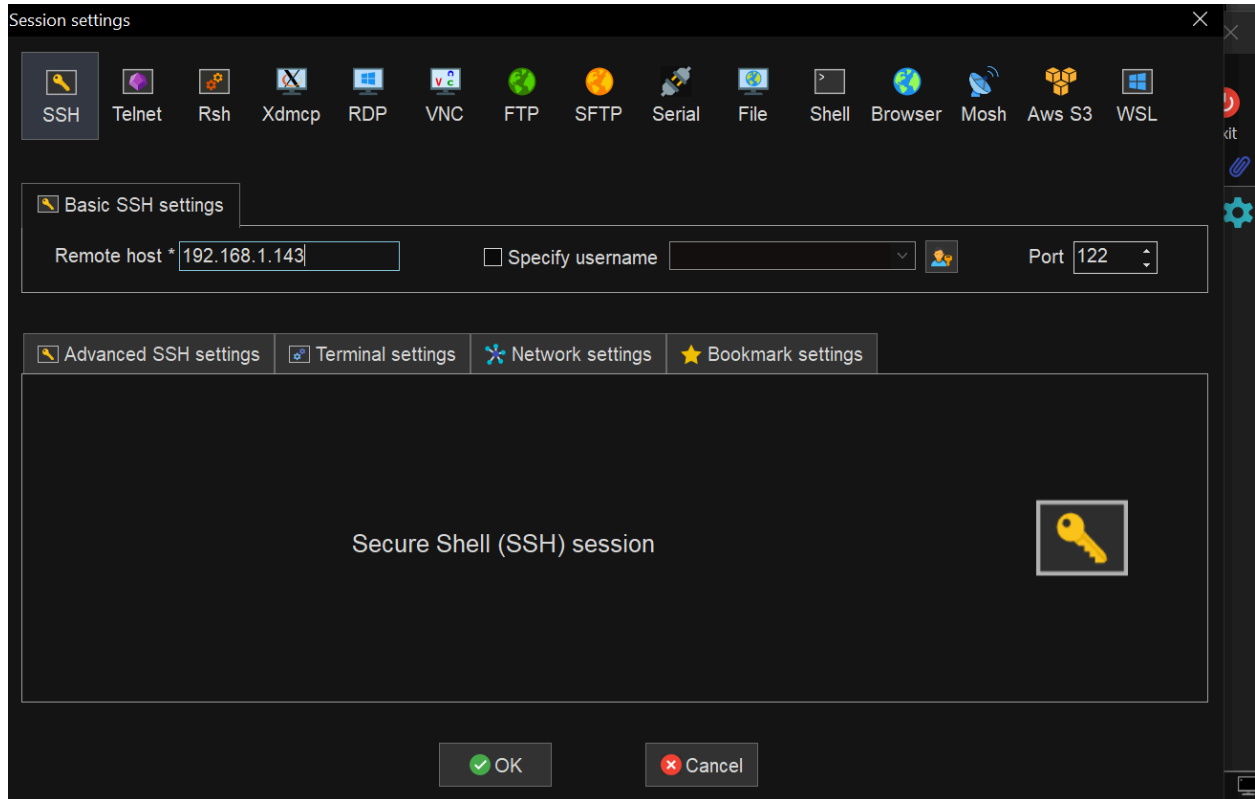


Una vez configurada la cuenta de usuario virtual, podemos dirigirnos al cliente y especificar el host, el puerto y las credenciales del usuario que acabamos de crear. Una vez establecida la conexión, tendremos la opción de guardar la sesión y acceder a varias funciones, incluida la ejecución de una terminal



# Mobaxterm

El servicio MobaXterm es una alternativa bastante robusta a PuTTY. Muestra las sesiones guardadas en un panel lateral. Para probar este cliente, voy a utilizar el servidor Linux. Por lo tanto, seleccionó 'Session' para crear una nueva sesión SSH, donde especifico la dirección IP de Linux y el puerto 122, ya que lo habíamos modificado previamente."



Seleccionamos la sesión a la que queremos conectarnos y luego hacemos clic en 'Execute'. Esto nos mostrará una terminal donde ingresamos nuestras credenciales de identificación. Una vez hecho esto, obtendremos acceso al servidor, similar a lo que estábamos haciendo anteriormente en PuTTY.

