13-2-2024

Open SSH

Escritorio Remoto

Marcos García Rodríguez

Álvaro Baquero Herrero

ÍNDICE

INTRODUCCIÓN

¿Qué es un servidor SSH?

Un servidor SSH es un sistema que utiliza el puerto 21 y que permite a los usuarios **acceder** y **gestionar** de forma **remota** otros dispositivos a través de una **conexión segura**.

Utiliza un protocolo de seguridad que proporciona **confidencialidad**, **integridad** y **autenticación** de los datos mediante el uso de **cifrado** simétrico y asimétrico.

¿Qué papel tiene el cifrado simétrico y asimétrico?

El cifrado asimétrico se utiliza en la **negociación** y **conexión** para establecer el cifrado simétrico y **validar** la identidad del cliente en el servidor.

El cifrado **asimétrico** es fundamental en SSH para la **autenticación** y la **creación de claves compartidas** de forma segura. Permite **validar** la identidad del cliente en el servidor y garantiza que la clave pública pueda compartirse **sin riesgo**, ya que solo la clave privada puede **descifrar** los mensajes encriptados con la clave pública.

Como **fuente principal**, hemos utilizado el siguiente vídeo que explica detalladamente el proceso de **instalación y configuración** del servidor OpenSSH, así como su conexión con un cliente.

Además, nos hemos guiado con la siguiente **página oficial de Microsoft** para el acceso en el servidor como administrador desde un equipo remoto.

Instalación

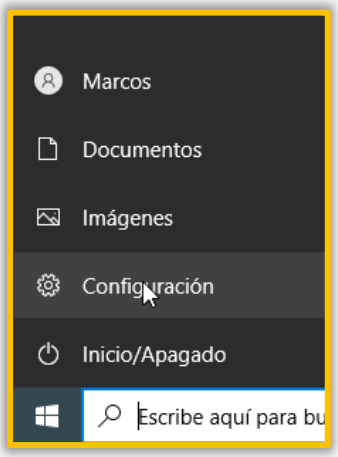
Por defecto, Windows 10 trae **preinstalado** el **Cliente** OpenSSH, por lo que únicamente será necesario instalar el Servidor OpenSSH en el servidor.

La **instalación** de SSH en Windows 10 es **sencilla** en principio. No debemos omitir ningún paso además de hacerlos todos de la misma forma en la que se explica a continuación.

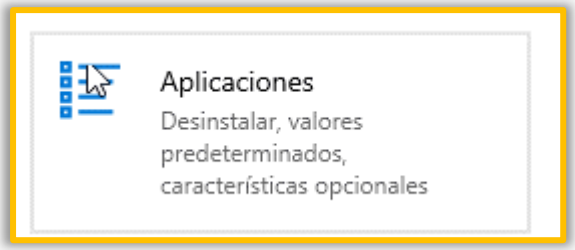
La instalación de este servicio requiere de **conexión a Internet** y que las 2 máquinas estés en la misma red. Se recomienda **desactivar el firewall de Windows** también, aunque no es necesario al 100%.

En primer lugar, debemos agregar la característica de Servidor OpenSSH en el servidor.

Para ello, entraremos en la Configuración del equipo.

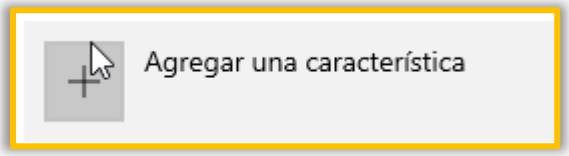


Accedemos al apartado de Aplicaciones situado en la pantalla principal de Configuración

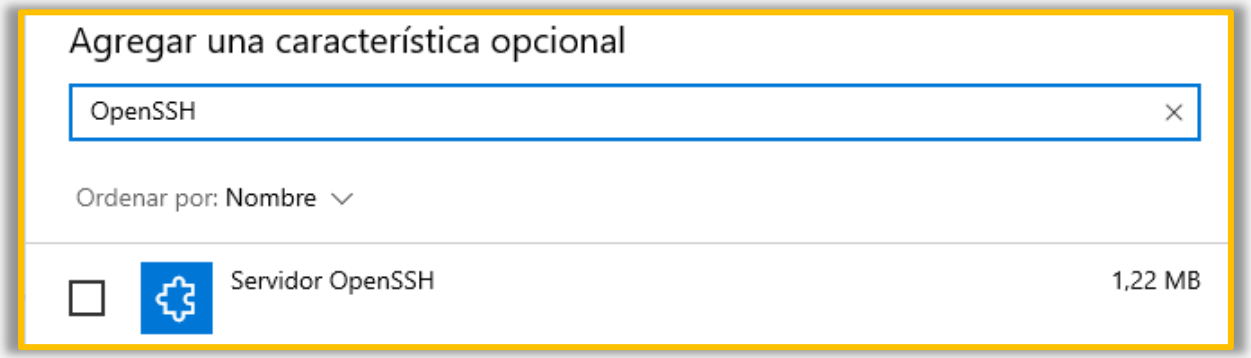


Entramos en características opcionales situado en Aplicaciones y Características.

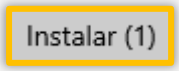
Pinchamos en agregar una característica ya dentro de características opcionales.



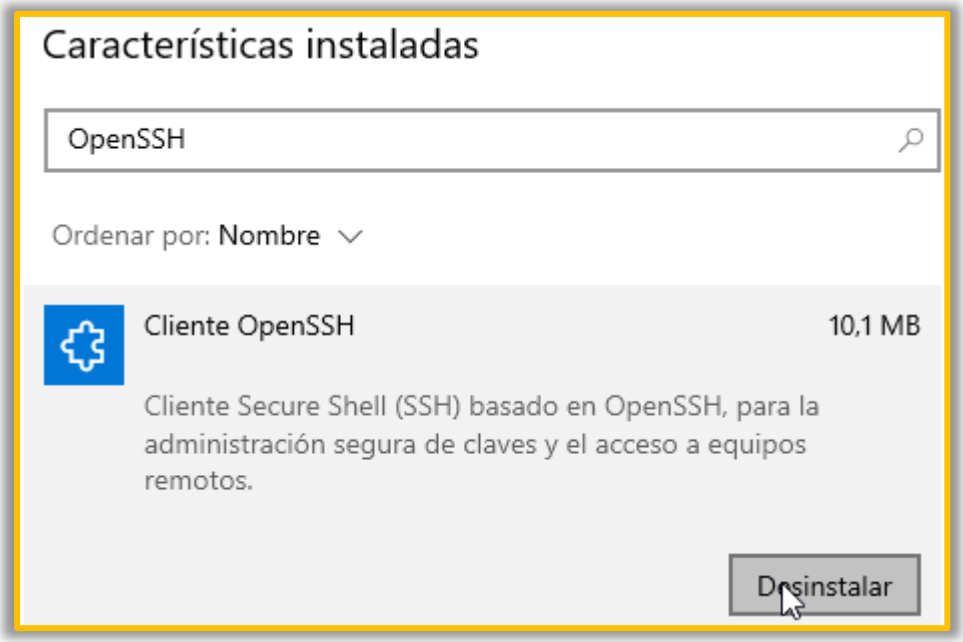
Seleccionamos Servidor OpenSSH. Este nos permite permitir la conexión de Clientes OpenSSH



Clicamos Instalar.



Mientras se instala el Servidor, quitaremos el servicio cliente para no tener problemas. Para ello buscaremos OpenSSH en las características instaladas, y lo desinstalaremos.

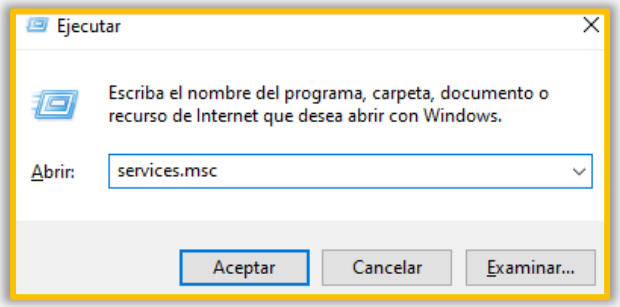


Ejecutar el Servicio

Para ejecutar el servidor, primero hemos de habilitarlo, para ello utilizaremos el panel de servicios, al que accederemos mediante la siguiente combinación de teclas:

- **Windows + R**

Y escribiremos el comando que aparece a continuación



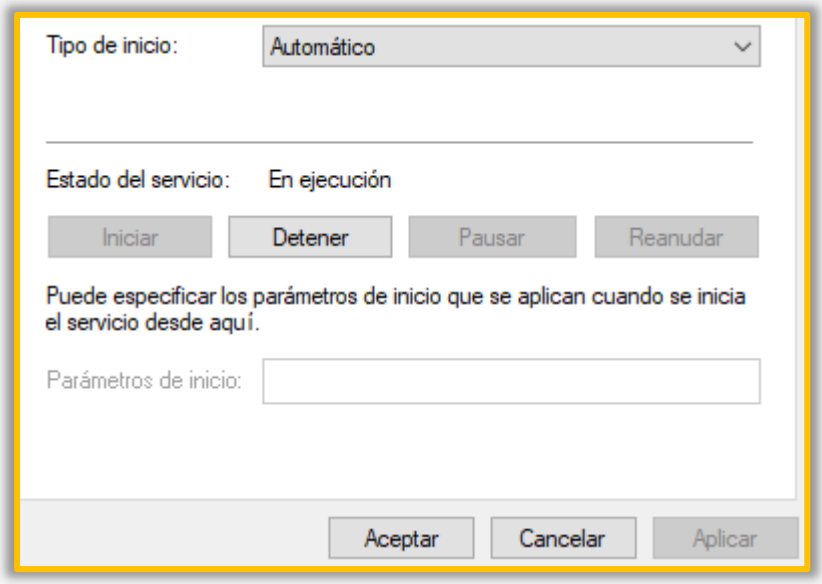
Una vez aquí, buscaremos los servicios:

- **OpenSSH Authentication Agent**
- **OpenSSH SSH Server**

Para iniciarlo, haremos doble clic sobre uno de ellos, y seleccionaremos

- **“Tipo de inicio: Automático”**

Aplicaremos y deberemos iniciar el servicio, ya que seguirá detenido.



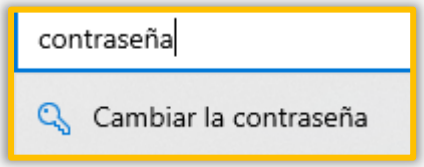
Haremos este proceso en ambos servicios.

Una vez iniciados, **volveremos al cliente**, en el que haremos el mismo proceso con el servicio “**OpenSSH Authentication Agent**”.

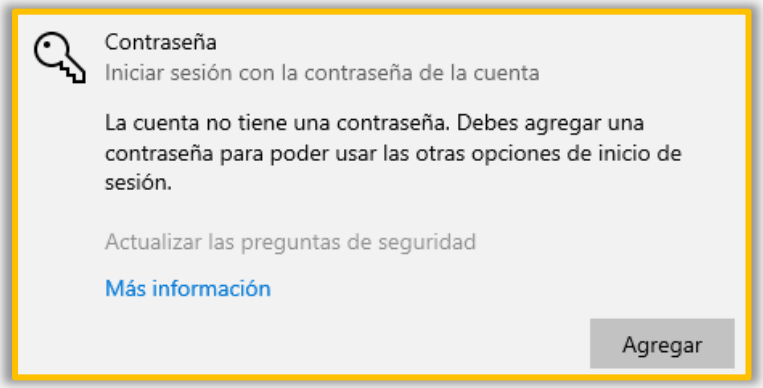
Creación de una contraseña Previa

Para poder acceder al Servidor SSH, **este deberá tener una contraseña**. De lo contrario, no podremos iniciar sesión desde un cliente a menos que cambiemos la configuración del archivo de configuración.

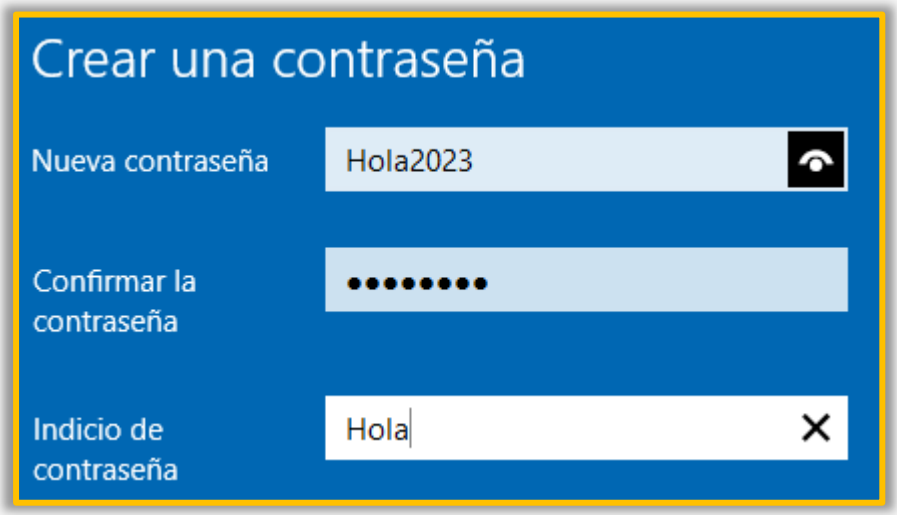
Buscamos **contraseña** en la barra de búsqueda de la configuración.



Clicaremos en **agregar**.



Escribimos la contraseña con la que queramos que los administradores accedan al servidor SSH.



Configuración

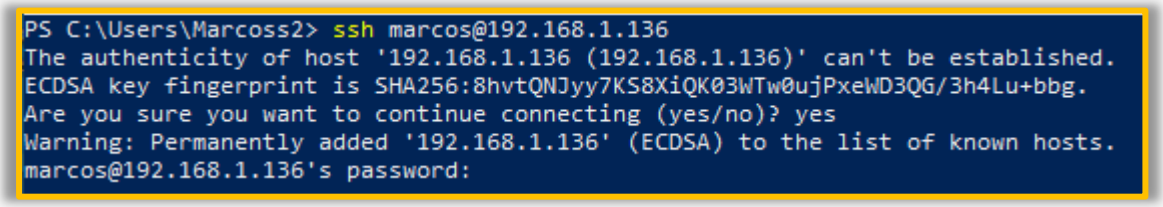
De nuevo en el cliente, iniciaremos la consola de Windows (PowerShell) mediante la cual conectaremos al cliente con el servidor., haremos esto buscándolo en la barra de búsqueda de Microsoft.

Para que ambos equipos se conecten entre sí, deben estar en la **misma red**. Nosotros, como estamos utilizando VirtualBox, utilizaremos una **red interna** entre ambos.

Para conectar el servidor al cliente, desde el PowerShell, escribiremos **"ssh (nombre)@IP"** del servidor, quedando de la siguiente forma:

- **ssh marcos@10.0.3.2**

Nos aparecerá un mensaje de que no se puede verificar la autenticidad del servidor, y nos preguntará si queremos continuar con el proceso, a lo que diremos que si (yes).



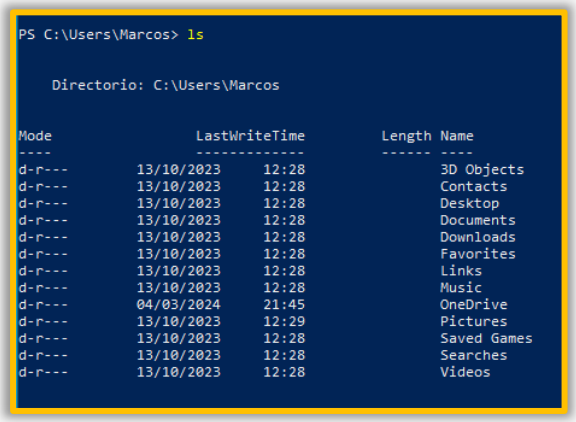
Escribimos la contraseña y accederemos al servidor. Podremos identificar esto por el nombre de usuario que aparece.

Una vez estamos en el servidor, abriremos un PowerShell en el que trabajaremos.

Guardaremos las claves de los diferentes usuarios en una carpeta específica para ello. En primer lugar, veremos las carpetas del directorio, para ello utilizaremos el comando "**ls**".

- **ls**

Este nos dará una lista de las carpetas existentes del usuario que estemos utilizando.



Una vez comprobado que no existe la carpeta destinada a ello, la crearemos nosotros mediante el comando siguiente:

- **mkdir .ssh**

```
PS C:\Users\Marcos> mkdir .ssh


    Directorio: C:\Users\Marcos


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         04/03/2024     22:02                .ssh
```

De esta forma, identificaremos rápidamente en qué carpeta se guardarán las claves.

Volveremos a hacer “ls” para ver que se ha creado en el lugar correcto.

```
PS C:\Users\Marcos> ls


    Directorio: C:\Users\Marcos


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         04/03/2024     22:02                .ssh
d-r---         13/10/2023     12:28                3D Objects
d-r---         13/10/2023     12:28                Contacts
d-r---         13/10/2023     12:28                Desktop
d-r---         13/10/2023     12:28                Documents
d-r---         13/10/2023     12:28                Downloads
d-r---         13/10/2023     12:28                Favorites
d-r---         13/10/2023     12:28                Links
d-r---         13/10/2023     12:28                Music
d-r---         04/03/2024     21:45                OneDrive
d-r---         13/10/2023     12:29                Pictures
d-r---         13/10/2023     12:28                Saved Games
d-r---         13/10/2023     12:28                Searches
d-r---         13/10/2023     12:28                Videos
```

Una vez hecho esto, saldremos del PowerShell del servidor.

- **Exit: Para salir del PowerShell.**
- **Exit: Para salir del servidor.**

```
PS C:\Users\Marcos> exit

marcos@DESKTOP-QPAQF0N C:\Users\Marcos>exit
Connection to 192.168.1.136 closed.
```

Y se habrá cerrado la sesión como cliente del servidor.

Creación de claves

Para crear la clave, en primer lugar, utilizaremos el comando "**cd**" para acceder a la carpeta en la que se guardará la clave.

- **cd .ssh**

Si utilizamos el comando **"ls"** podremos ver el archivo que contiene los datos de la conexión anterior.

```
PS C:\Users\Marcoss2\.ssh> ls


    Directorio: C:\Users\Marcoss2\.ssh


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         04/03/2024     21:59            176 known_hosts
```

Y ahora sí, generaremos las claves pública y privada.

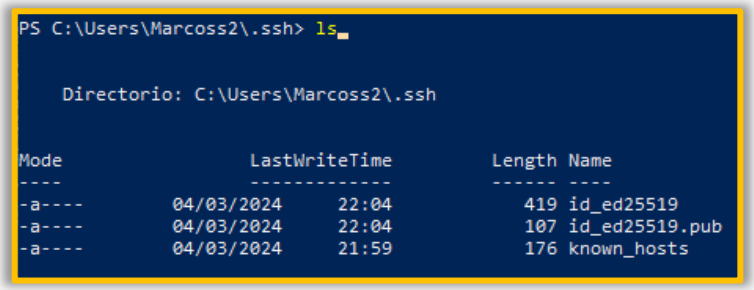
- **ssh-keygen -t ed25519**

Como por defecto, el protocolo SSH utiliza el protocolo RSA, vamos a emplear este comando para usar el protocolo ed25519
Nos pedirá un nombre que asignarle a nuestro archivo, nosotros lo dejaremos por defecto, por lo que no escribiremos nada y haremos **"Intro"** y nos pedirá una contraseña con la que el cliente accederá al servidor. Como vamos a acceder mediante cifrado de clave pública y privada, el servidor utilizará estas para saber que realmente somos nosotros. Por lo que no ingresaremos **ninguna contraseña**.
Y se creará de la siguiente forma.

```
PS C:\Users\Marcoss2\.ssh> ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\Marcoss2/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\Marcoss2/.ssh/id_ed25519.
Your public key has been saved in C:\Users\Marcoss2/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:RKTBTsfFaxrmt0t/3gYGc25ucL2/mA0WVrEkoKmFIbo marcoss2@DESKTOP-9S6HH4R
The key's randomart image is:
+--[ED25519 256]--+
|     o.+oo.... o |
|    . ++= +   o o|
|   . o.o.+ .   o |
|    . ..= o o o  |
|   E   +S+   B . |
|        o . o B .|
|         ... O ..|
|         .... Oo.|
|          ...*.+=|
+----[SHA256]-----+
```

Para comprobarlo, podemos hacer **“ls”** y veremos cómo aparecen tres archivos:

1. Clave privada
2. Clave pública
3. Datos de conexión



Traspaso de clave pública

Para el acceso del cliente en el servidor mediante un **cifrado asimétrico**, serán necesarias dos claves, la **clave privada** la tendrá únicamente el cliente, pero la **pública** deberemos agregarla al servidor, a continuación, veremos cómo se hace.

Mediante el uso del comando **SCP** podremos copiar un archivo desde un equipo hacia otro.

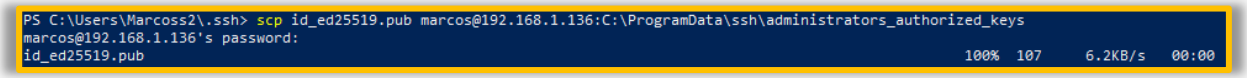
Este comando estará compuesto por:

1. El tipo de comando (**SCP**)
2. Lugar donde está la clave
3. Lugar a donde queremos llevar el archivo.
4. Nombre con el que se guardará el archivo.

De esta forma, el comando quedaría así:

- **scp id_ed25519.pub** marcos@192.168.1.136:C:\ProgramData\ssh\administrators_authorized_keys

Una vez ejecutemos el comando, insertaremos la contraseña y se habrá copiado.

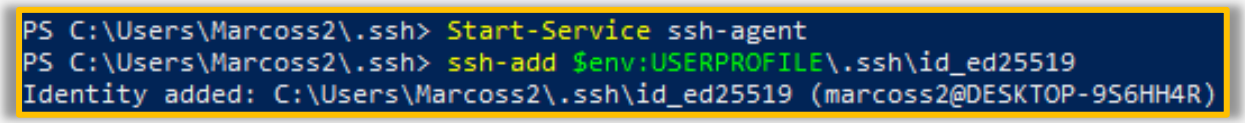


Con la contraseña pública ya en el servidor, será necesario utilizar un servicio que nos ofrece el cliente SSH para **relacionar la clave pública** que hemos mandado al servidor **con la clave privada** existente en el cliente. Además, permite verificar que realmente pertenece a este usuario cada vez que accedamos al servidor. Para ello, usaremos el comando **Start-Service**.

- **Start-Service ssh-agent**

Y posteriormente emplearemos el comando que verifica al usuario actual con la clave privada.

- **ssh-add $env:USERPROFILE\.ssh\id_ed25519**



Finalmente, vamos a utilizar el siguiente comando, del cual explicaremos algunas funciones importantes.

- En primer lugar, "**$remotePowershell**" nos permitirá hacer ver al servidor que estamos accediendo con un PowerShell remoto.
- También utilizaremos "**Add-Content -Force -Path**" para añadir una regla en el servidor que indica el forzado de apertura de la ruta indicada cuando intentemos acceder desde el cliente.
- Además, agregaremos el valor "**authorizedKey**" para indicar que la ruta indicada es una clave válida.
- Y finalmente, el comando "**inheritance/r"** que nos permite agregar una clase con herencia de las ya existentes. En nuestro caso, lo agregaremos al grupo de administradores. Como el idioma del sistema es uno diferente del inglés, escribiremos el código del grupo para evitar problemas. Podemos verlo con el siguiente comando:

- **Get-LocalGroup | Select-Object Name, SID**

```
PS C:\Users\Marcoss2> Get-LocalGroup | Select-Object Name, SID

Name                                SID
----                                ---
Administradores                     S-1-5-32-544
Administradores de Hyper-V          S-1-5-32-578
Device Owners                       S-1-5-32-583
Duplicadores                        S-1-5-32-552
IIS_IUSRS                           S-1-5-32-568
Invitados                           S-1-5-32-546
```

- **$remotePowershell = "powershell Add-Content -Force -Path $env:C:\ProgramData\ssh\administrators_authorized_keys -Value '$authorizedKey';icacls.exe ""$env:C:\ProgramData\ssh\administrators_authorized_keys"" /inheritance:r /grant ""*S-1-5-32-544:F"" /grant ""SYSTEM:F"""**

```
PS C:\Users\Marcoss2\.ssh> $remotePowershell = "powershell Add-Content -Force -Path $env:C:\ProgramData\ssh\administrators_authorized_keys -Val
ue '$authorizedKey';icacls.exe ""$env:C:\ProgramData\ssh\administrators_authorized_keys"" /inheritance:r /grant ""*S-1-5-32-544:F"" /grant ""SY
STEM:F""
```

Comprobación de Clave en Servidor

Para comprobar que las claves funcionan correctamente, intentaremos acceder al otro equipo (al servidor) desde el cliente. En primer lugar, utilizaremos la función **$remotePowershell** para que el servidor reconozca que vamos a entrar con una clave, y permita leer y contrastar el archivo almacenado de la clave pública, con la clave del cliente.

- **ssh marcos@192.168.1.136 $remotePowershell**

```
PS C:\Users\Marcoss2\.ssh> ssh marcos@192.168.1.136 $remotePowershell
marcos@192.168.1.136's password:
archivo procesado: \ProgramData\ssh\administrators_authorized_keys
Se procesaron correctamente 1 archivos; error al procesar 0 archivos
```

Y como vemos, se ha procesado correctamente el archivo de la clave.

Y, para terminar, nos conectaremos al servidor y no nos pedirá ninguna clave ni contraseña.

- **ssh marcos@192.168.1.136**

```
PS C:\Users\Marcoss2\.ssh> ssh marcos@192.168.1.136
```

```
marcos@DESKTOP-QPAQF0N C:\Users\Marcos>
```