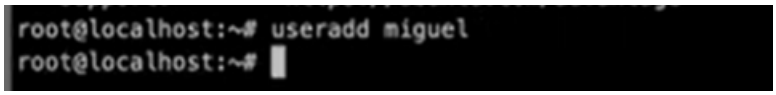
## CREAR USUARIOS ADICIONALES EN LAS MÁQUINAS LINUX

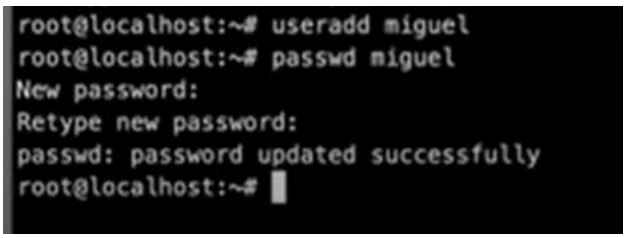
En principio puede parecer una tarea que podría ser no muy necesaria pero realmente sí que es interesante porque es el primer paso a implementar para conseguir una mejora de seguridad fundamental que es desactivar el usuario root y no permitir login con el usuario root. Para eso tenemos que crear antes nuevos usuarios en el sistema y podemos crear en el sistema cualquier número de usuarios que necesitemos.

Lógicamente para este usuario que voy a crear voy a tener que utilizar el usuario root porque realmente no hay otro usuario creado en el sistema.

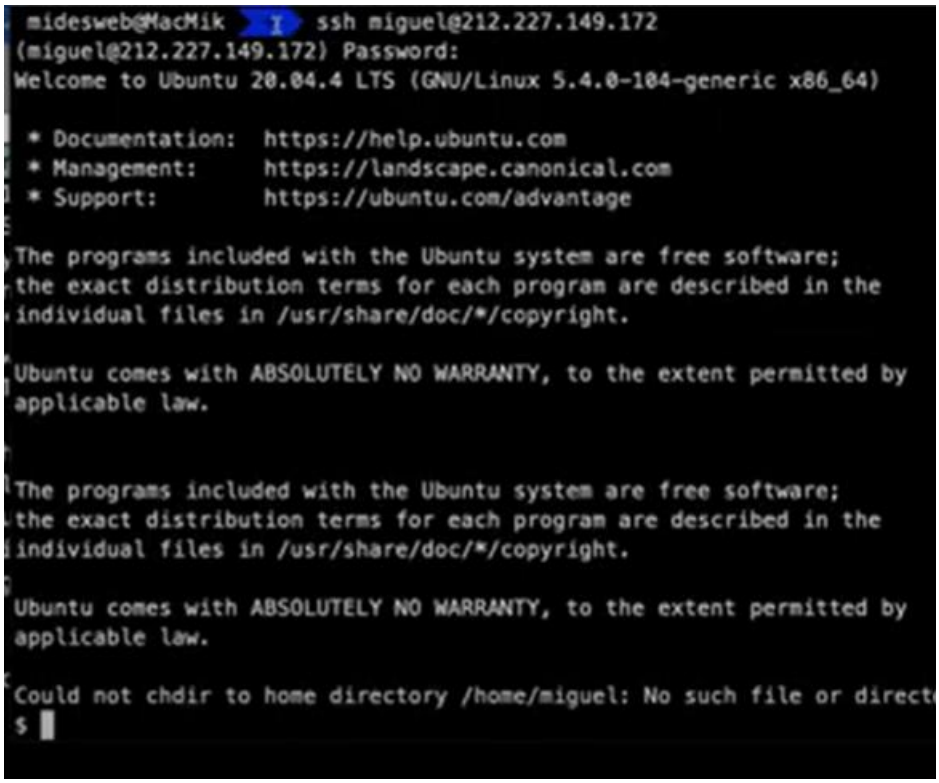
Copiamos la ip para poder hacer la conexión por ssh y ahora me pide la contraseña y ya estamos dentro del servidor. Ahora ya puedo crear ese usuario para ello hay un comando que se llama **useradd**, a continuación, tengo que especificar el nombre del usuario que voy a crear y le colocamos cualquier nombre que queramos y ya podemos dar enter.




Le proporcionamos una clave a este usuario para que pueda conectarme utilizando esta clave, para eso está el comando paswd y a continuación le tengo que indicar el nombre del usuario al que quiero aplicarle esta clave. me pregunta cuál es la clave y la tengo que repetir y ya está, ya tengo un usuario ya tengo su clave. Sobra decir que es importante que utilicemos claves bien seguras.

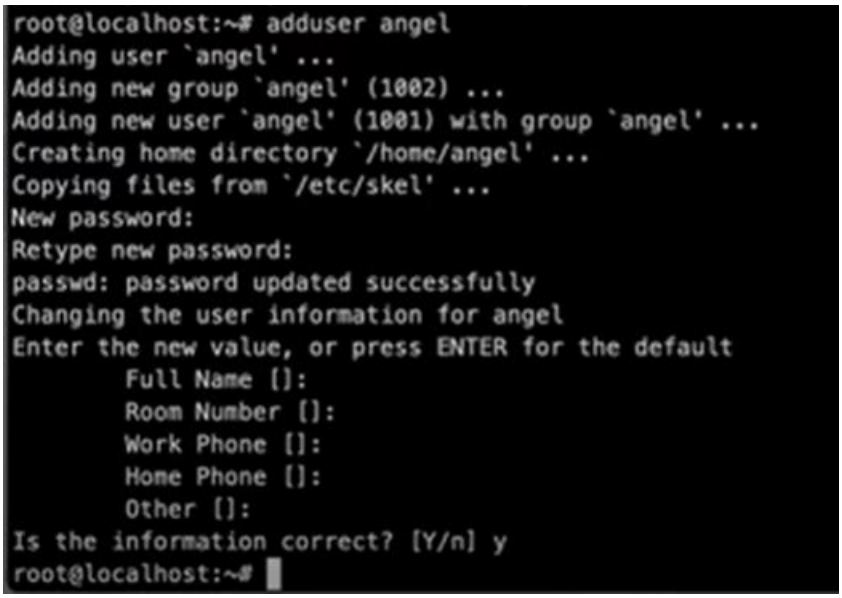



Ahora vamos a hacer una comprobación con otra ventana del terminal para hacer una conexión con este usuario, nos conectamos a través de ssh con el usuario miguel, al cargar ya no pone root si no, miguel y a continuación pide la clave y hemos podido conectar el servidor perfectamente con el usuario miguel.




Si observáis aquí nos aparece un error que no se ha creado un directorio y que no se puede dirigir al directorio /miguel.

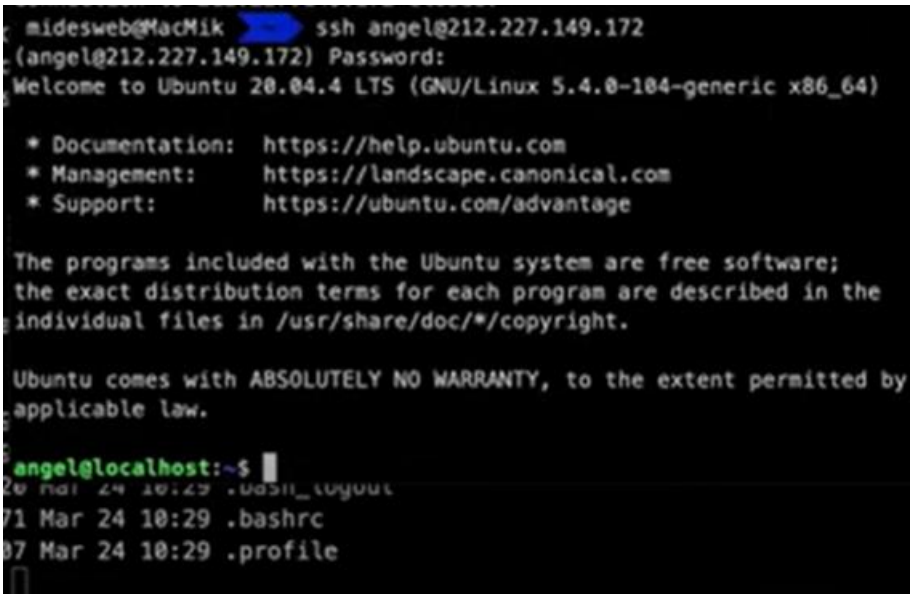
Existe otro comando para poder crear el usuario, antes hemos utilizado y **useradd** y vamos a utilizar **adduser**. Vamos a llamar ángel a este segundo usuario y si os fijáis lo que se está haciendo es crear el directorio de este usuario y está configurando diversos archivos ya para este usuario incluso me está pidiendo la clave con lo cual me asegura que este usuario se vaya a crear con clave desde el principio, colocamos una que sea suficientemente segura y a continuación me solicita información sobre cómo debe ser este usuario, el nombre y algún tipo de información del usuario aunque no hace falta que coloques nada y una vez que lo tienes todo pues ya le das a ok.




En este caso ha creado un directorio home que es el directorio típico donde este usuario va a tener todas sus cosas.

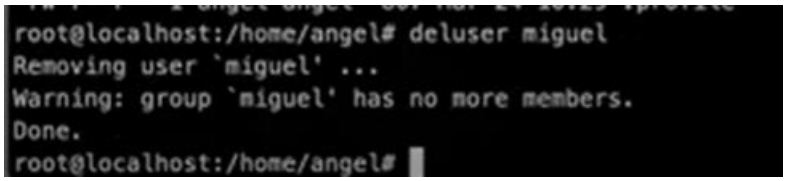
Hacemos aquí un cd /home/angel y aquí encontramos todo una serie de configuraciones de este usuario en particular aquí podríamos tener otras configuraciones como por ejemplo la carpeta .ssh para colocar las llaves de seguridad. Es la manera más útil y más interesante de crear usuarios.

Salimos con exit y nos vamos a intentar conectar con el usuario ángel la ruta para la conexión sería la misma simplemente le ponemos ángel y a continuación nos pide la contraseña y ya no habrá ningún problema para poder conectarnos con este usuario.



## ELIMINAR USUARIOS

Ahora vamos a aprender a eliminar los usuarios, teníamos un usuario que era miguel que realmente no ha salido del todo bien y lo voy a eliminar para ello simplemente voy a colocar el comando **deluser** y a continuación el nombre del usuario y con esto ya estaría eliminado este usuario




deluser tiene algunas configuraciones interesantes que podemos utilizar, como por ejemplo, **deluser –help**, --remove-home que nos permite eliminar el directorio home de este usuario o --remove-all-files que nos permite eliminar no solo el directorio home sino además todos los archivos que pertenezca a este usuario que haya podido crear a lo largo de todo el sistema.

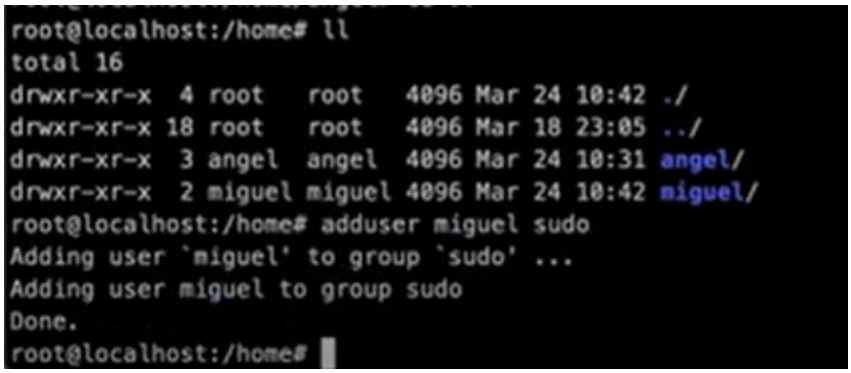
## CREAR USUARIOS CON PERMISOS DE SUPERUSUARIO

A continuación, vamos a explicar cómo crear usuarios que además tengan permisos administrativos es decir el permiso de superusuario para poder realizar cualquier tipo de acciones.

Vamos a ver cuáles son estos permisos y porque los necesitamos. Algunas operaciones que se hacen con el servidor como por ejemplo **apt update** que necesitan permisos de administrador.

Lo que vamos a hacer es crear un usuario nuevo en esta máquina al que luego le daremos permisos de superusuario, para eso tengo que empezar creando el usuario y luego le añadiremos los permisos de superusuario entonces lanzamos el comando **adduser** y a continuación el nombre del usuario que en este caso voy a utilizar de nuevo el nombre de usuario miguel, me pide la clave que la repita y a continuación toda la información de miguel que no hace falta que coloque.

Si nos vamos a la carpeta anterior y hacemos un ls vemos que ya tenemos el home directorio de ángel y el home directorio de miguel, ahora lo que voy a hacer es darle permisos administrativos o de súper administrador para este usuario miguel, para eso utilizo el mismo comando **adduser** con el nombre de usuario al que quiero añadirle esos permisos y el grupo de los usuarios que van a tener permisos de administrador, que se llama **sudo**. Con este comando lo que estoy haciendo es al usuario miguel darle permisos de superusuario y hemos añadido a miguel al grupo de los usuarios sudo o sea de los usuarios super administradores.




Ahora nos desconectamos con el usuario ángel y a continuación hacemos la conexión con el usuario miguel, me pide la contraseña ya estoy dentro con el usuario miguel, estamos en la carpeta personal de miguel y como todo parece estar funcionando perfectamente vamos a probar a hacer el comando apt update, si os fijáis está dando exactamente el mismo error que antes y si le hemos dado permisos de administrador ¿porque ahora no me deja realizar este comando y me dice que el permiso está denegado?

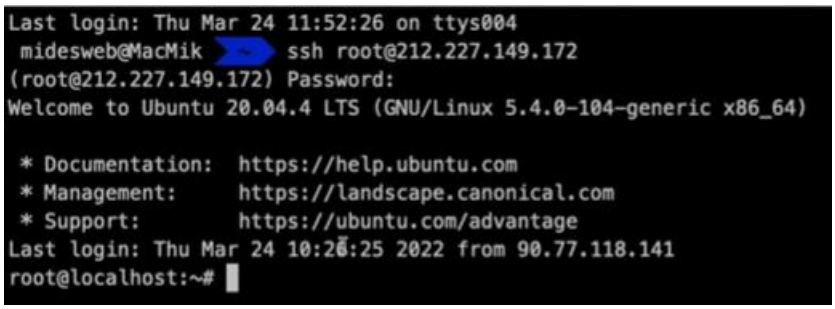
Cuando nosotros tenemos usuarios que son del grupo de súper administradores y queremos hacer acciones que realmente trabajen mediante el rol de superusuario entonces lo que vamos a hacer es colocar delante del comando la palabra **sudo** lo tenemos aquí **sudo apt update** y ahora este comando ya me dejara, antes me preguntará la clave y esta es la seguridad que nos ofrece el tener un usuario del grupo sudo. Cada vez que necesite hacer alguna tarea de administración tendré que indicarle la clave de este usuario

## DESACTIVAR EL ACCESO DEL USUARIO ROOT

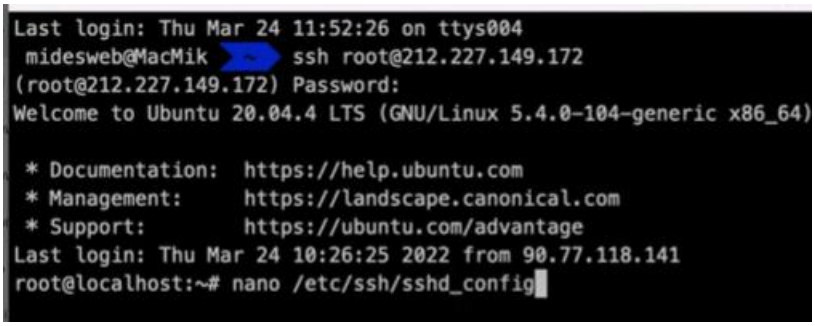
Es una tarea que se hace muy muy rápido, pero ojo solo lo podemos hacer cuando ya tenemos otro usuario que tenga características o posibilidad de realizar tareas de superusuario.

Ahora que ya tenemos otro usuario que es capaz también de hacer las tareas de superusuario es mejor que el usuario root esté desactivado porque es el típico usuario que en los ataques de fuerzabruta van a intentar encontrar una clave para poder acceder al servidor y realizar tareas administrativas de todo tipo.

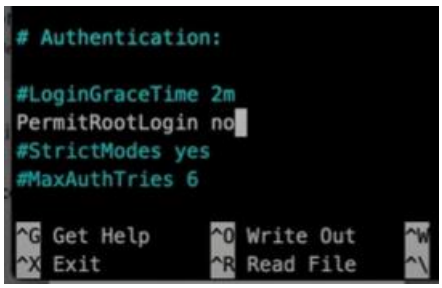
Otra cosa que queremos hacer es darle acceso al usuario que hemos hecho de manera adicional por medio una llave ssh para mejorar la seguridad. Para realizar esa tarea como todavía está activo el usuario root lo vamos a hacer con él y vamos a ver cómo se hace, hacemos ssh contra este servidor



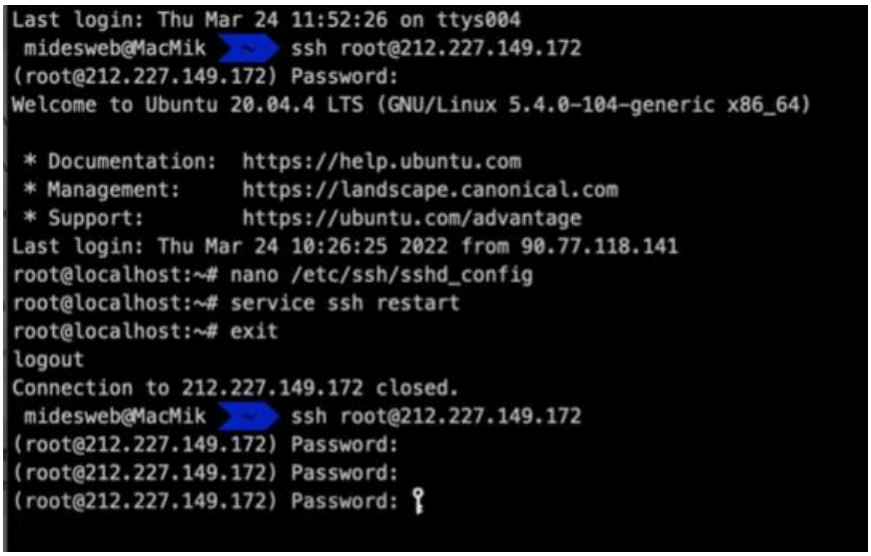

Una vez autenticado hay un archivo de configuración que es el que tengo que editar y para editarlo vamos a utilizar un editor de textos.




El archivo que quiero editar está en la carpeta **/etc/ssh/sshd_config** editamos este archivo y ahora nos movemos por el archivo buscando la configuración que nos va a servir para desactivar el usuario root está aquí **PermitRootLogin** y lo que vamos a decirle es que no nos permita loguearmos.




Ahora tenemos que guardar el archivo y ya estaría hecha esta configuración, nos falta solamente reiniciar el servicio de ssh, para eso utilizamos el comando **service ssh restart** y ahora vamos a probar a ver si nos dejan loguearnos con root a continuación hacemos exit y nos volvemos a conectar con el servidor, vuelvo a poner la contraseña.




Vamos a intentar conectarnos con el usuario miguel que habíamos creado anteriormente. Ahora indicamos la clave de miguel y con miguel sí que nos deben dejar pasar.

Otro de los objetivos es conseguir que el usuario miguel pueda acceder a través de una llave ssh, para poder asociar llaves ssh para el acceso a determinados usuarios, hay un archivo que se llama authorized_keys donde podemos colocar todas las llaves ssh que queríamos que permitiese el acceso con ese usuario, ese archivo estaba una carpeta .ssh y esa carpeta a su vez está dentro de la carpeta personal del usuario al que queremos autorizar con la llave ssh, bien, ahora miramos en la carpeta personal de este usuario, vamos a hacer un **ls -la** y quiero mostraros que en la carpeta personal de este usuario no está la carpeta .ssh.

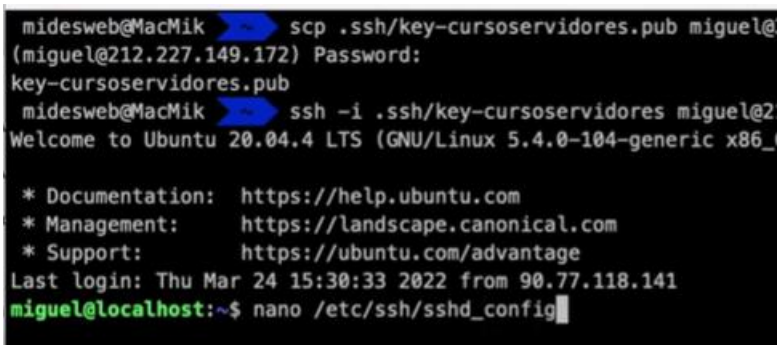
En un primer paso vamos a tener que crear esa carpeta con el comando **mkdir,** y creamos la carpeta .ssh, una vez que ya la tenemos podemos hacer un ls -la y ya está la carpeta donde tengo que colocar el archivo authorized keys con la llave que quiero que se permita el acceso por ssh esa llave la tengo en mi ordenador local o sea que voy a hacer un exit para irme a mi ordenador local y ahora vamos a hacer la subida del archivo con la llave pública, acordaros siempre que **la llave privada la quedamos en local y la pública es la que dejamos en el servidor** el comando para subir es **scp** a continuación le digo la ruta del archivo que quiero subir, la llave pública y luego le tengo que decir el usuario miguel, arroba, pego la ip del servidor, a continuación tengo que decirle dónde vamos a colocar esta llave y sería en la carpeta personal de miguel esto lo consiguió con el rabito de la ñ y a continuación una barra .ssh y a continuación authorized_keys


Es un archivo de texto donde podríamos tener varias llaves, en el caso de que quieras que haya varias llaves ssh que permitan el acceso a este mismo servidor

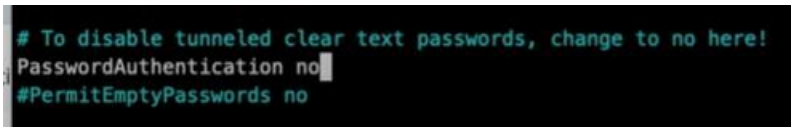
con este mismo usuario en este caso pues tendrías que editar el archivo dentro del servidor y poner las distintas llaves en el mismo fichero de texto.

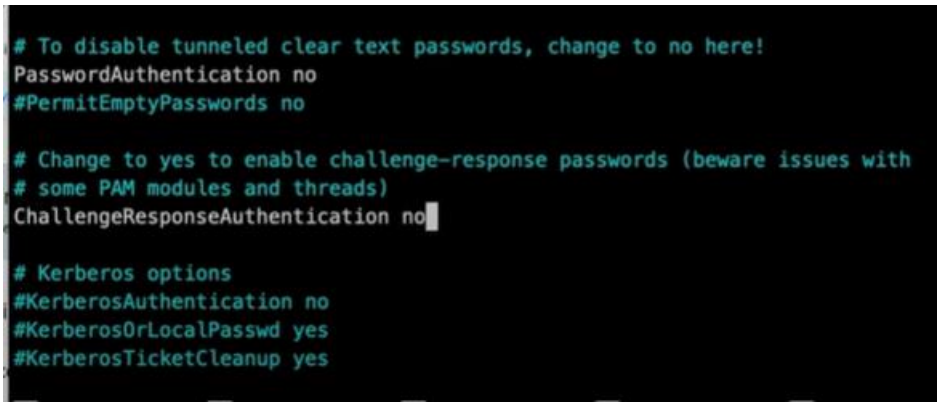

Ahora ya podría conectar utilizando esa llave con este servidor el comando ssh acordaros - i era para introducirle una llave y a continuación ponemos miguel arroba y la ip y ya estamos dentro del servidor con la llave.

El objetivo sería ya desactivar por completo cualquier acceso con una clave a este servidor que no se pueda conectar directamente con ninguna clave y que solo se puede conectar a través de la llave ssh y eso lo vamos a tener que hacer de nuevo en el archivo de configuración que hemos abierto antes utilizamos el programa nano y ahora editamos el archivo de antes que era el /etc/ssh/sshd_config.




La configuración que queremos encontrar ahora sea password authentication y lo encontramos comentado, todas las líneas que aparecen con una almohadilla adelante son código que está comentado entonces simplemente lo vamos a descomentar y le vamos a decir que no y con esto estamos haciendo que no se puede autenticar por password en este servidor.




Ahora hay que guardar el archivo y reiniciar el servicio de csh guardamos con el mismo nombre archivo.

Vamos a reiniciar el servidor de ssh para eso utilizamos el comando **sudo services ssh restart** como vimos antes y con esto ya hemos reiniciado el servidor de ssh con lo cual podemos probar si nos deja o no conectarnos con nuestro usuario por clave.


Nos está diciendo ya directamente que no está permitido la conexión con este usuario por clave y también si intentásemos a utilizar el usuario root esto también lo ha afectado.

## CREAR LLAVES SSH

Creación de llaves SSH en Linux, Mac y Windows. Explicamos el proceso para crear una llave SSH en local y usarlas al hacer login en servidores remotos, de modo que podamos aumentar la seguridad de las conexiones.

El proceso de creación de llaves es bastante sencillo en Linux y un poco más farragoso en Windows, pues se tiene que usar un software en particular. De todos modos, es bastante rápido. Vamos a resumirlo con algunas notas extra para poder aclarar posibles dudas.

Para conocimiento general, la llave SSH está compuesta por dos ficheros, uno con la llave pública y otro con la llave privada. La llave pública es la que tenemos que usar cuando sea necesario, colocándola en el lugar adecuado cuando se solicite. Sin embargo, la llave privada esalgo que debe quedar a buen recaudo y tener cuidado de que no sea accesible por otras personas. Llave pública y privada tienen el mismo nombre de archivo, aunque la llave pública tendrá extensión ".pub", por lo que la podremos distinguir fácilmente.

Una llave SSH se puede usar varias veces, de modo que podrías configurarlas para el acceso adiversos servidores. Tampoco hay problema por crear varias llaves SSH en tu sistema si fuera necesario. Cada una tendría ese juego de par "llave pública / llave privada".

## CREAR LA LLAVE SSH EN LINUX / MAC

Para hacer esta tarea usamos un comando de sistema: ssh-keygen. Este comando se encarga del proceso de creación de los archivos de la llave. Al ejecutarlo sin parámetros crea una llave de tipo "RSA".

**ssh-keygen**

Podemos especificar el tipo de la llave con el parámetro -t. De modo que aquí

```
❯ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/midesweb/.ssh/id_rsa):
```

sh-keygen -t rsa

estaríamos creando también una llave RSA.

Podemos especificar el número de bits en la clave que se va a generar. Para llaves "rsa" el valor predeterminado es de 2048 bit, que es suficiente en la mayoría de los casos. El valor mínimo sería 1024 bits. Por medio de la opción -b lo podemos configurar. Esto creará una llave de 4096 bit:

Como puedes ver en las anteriores imágenes, una vez iniciado este comando nos va a solicitardos cosas:

- La localización donde almacenar la clave y el nombre del archivo.
- La clave para poder usar esta llave ssh.

ssh-keygen -t rsa -b 4096

```
❯ ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/midesweb/.ssh/id_rsa):
```

Sobre estos puntos es importante conocer:

## RUTA Y NOMBRE DEL ARCHIVO PARA ALMACENAR LA LLAVE

Esta parte la puedes configurar a tu gusto. Realmente, cuando se arranca el comando ssh-keygen ya te sugiere una carpeta donde se suelen almacenar las llaves ssh y un nombre de archivo. En la mayoría de los casos es solamente aceptar la propuesta.

Nota: La carpeta donde se guardan las llaves se llama ".ssh" y por tanto es una carpeta oculta, por comenzar por ".". Como puedes observar la carpeta .ssh la sitúan de manerapredeterminada en la carpeta de tu usuario.

Te sugiero mantener todas las llaves ssh en la carpeta que nos indican. Sin embargo, el nombredel archivo lo puedes variar si lo deseas y de hecho será necesario en la mayoría de los casos, porque generalmente tendrás una llave para el acceso a cada servidor. Solamente acuérdate del nombre de tu llave a la hora de configurar los sistemas, o a la hora de usarla para loguearte en el servidor remoto.

El nombre del archivo o nombre de la ssh key es indiferente y no necesita una extensión enparticular. Por ejemplo, en mi caso podría guardarlas en esta ruta y nombre de archivo.

```
/Users/midesweb/.ssh/llave-ssh-servidor-preproduccion
```

## LA CLAVE ASOCIADA A LA LLAVE

La llave ssh se puede asociar a una clave creada por ti mismo, de modo que cuando alguien use la llave ssh tendrá que indicar además la clave asociada.

La creación del password para la llave es meramente opcional. Es interesante porque le agrega un nivel extra de seguridad a tu llave SSH, de modo que, si alguien la consigue, seguiría sin poder usar la llave si desconoce tu clave. Si no pones clave, la dejas en blanco, simplemente note preguntará nada cuando se vaya a usar la llave ssh.

Obviamente, si le ponemos una clave será todavía más segura, porque aparte del archivo de la llave ssh un posible intruso necesitaría también la clave que haya asignado en el momento de la creación.

Sin embargo, he de reconocer que tener que escribir la clave cada vez que accedes al servidor usando la llave es un poco tedioso, por lo que muchas veces no se usa. Por supuesto, tienes queevaluar también los riesgos de perder u olvidar la clave

que acabas de generar para la llave, puesto que, si no tienes la clave correcta, la llave ssh será inservible.

En resumen, poner clave a esta llave es una decisión tuya. Solo que si te decides por crear esa clave, guárdala bien para que no se te pierda, porque entonces no podrías usar la llave SSH.

## USAR LA LLAVE SSH

Para usar la llave simplemente tienes que localizar la carpeta donde se generó la llave y encontrar el archivo con el nombre de la llave que finaliza por.pub.

A la hora de crear ciertos servicios de alojamiento, o algunos servicios de Git, te solicitarán tu llave. Tendrás que abrir el .pub, copiar su contenido y pegarlo donde se te solicite.

Para el ejemplo de SSH, una vez configurada tu clave desde el panel de control de tu proveedorde alojamiento o proveedor cloud, tendrás que indicar el nombre de tu clave para poder hacer el login. Lo consigues con un comando como este:

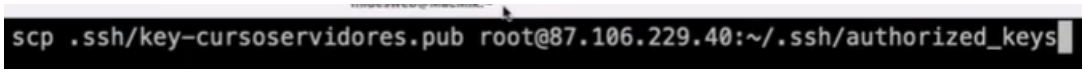
```
ssh -i ~/.ssh/nombre_mi_clave user@255.255.0.1
```

Con el anterior comando usarás tu llave SSH llamada "nombre_mi_clave", que está en la carpeta "~/.ssh/", para conectarte a tu servidor con IP 255.255.0.1 y con el nombre de usuario"user".

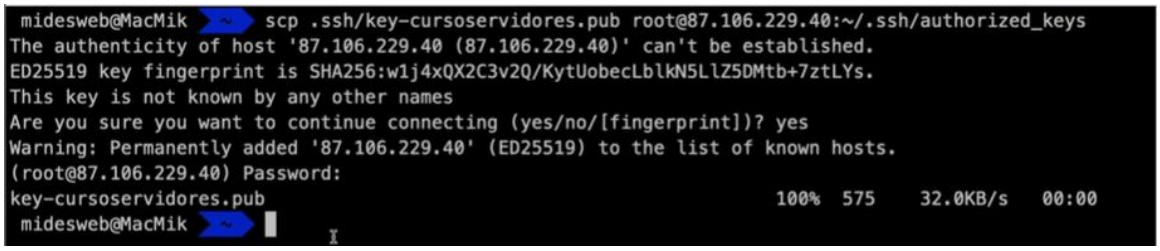
Si configuraste una clave para esta llave, el terminal te solicitará que la introduzcas antes deproceder a realizar el login.


Si, por ejemplo, un servidor lo quiero configurar para que podamos utilizar una de las llaves ssh que tenemos creadas, vamos a utilizar la llave pública que se llama key-cursoservidores. Lo que tengo hacer es subir ese archivo a este servidor y para eso utilizamos un comando específico que es el comando scp. El comando sirve justamente para eso, para subir archivos que tengo en local y dejarlos en el servidor y se utiliza de esta manera: scp a continuación le tengo que decir el nombre del archivo que quiero copiar en el servidor en este caso va a ser la llave pública porque es la llave que tiene está en el servidor, tengo que decir dónde está esa llave para eso utilizamos la ruta .ssh y dentro de esa carpeta está key-cursoservidores.pub y fijaros que es la llave pública y a continuación le tengo que decirle dónde lo vamos a colocar y será con el usuario root, nos conectaremos al servidor que tiene este ip y a continuación le tengo que decir dónde está la ruta de

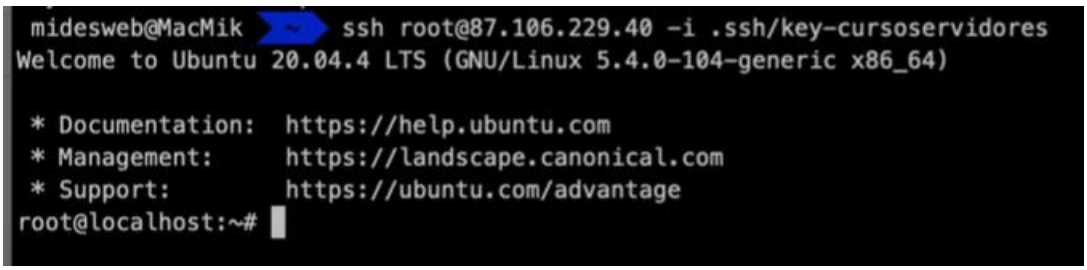
el destino de este archivo, para ello colocó dos puntos y ahora la ruta concreta que está dentro de la carpeta personal del usuario root para ello voy a utilizar el simbolito este de aquí que va encima de la ñ lo he conseguido colocar con alternativa y luego la letra eñe y a continuación barra .ssh que es donde están todas las llaves ssh, barra authorized_keys.




Fijaros de escribir bien esta ruta porque si no colocáis la llave en la carpeta correcta no va a funcionar. Una vez ya lo tenemos le damos a enter. Si no nos hemos conectado antes a este servidor le digo que sí y ahora a continuación me pide la clave del root, ya que para hacer este proceso necesito una clave de root para dejar la llave en el lugar correcto. Ahora ya se ha transferido el archivo de la llave pública, aquí está toda la información de que se ha podido transferir



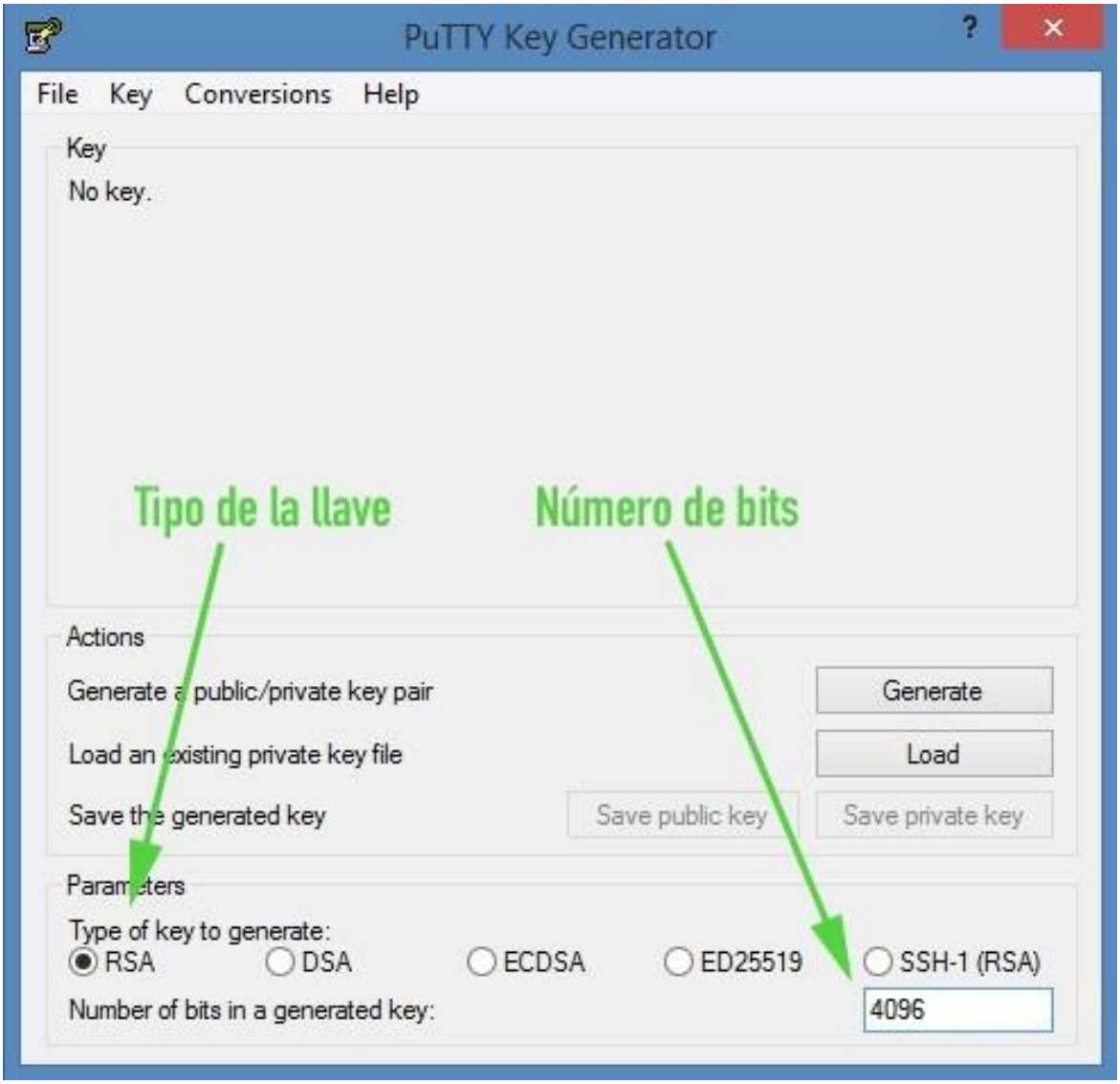

A continuación ya puedo utilizar la llave privada para conectarme a este servidor, para eso utilizamos el comando ssh igual que habíamos utilizado antes, tenemos que decirle que vamos a utilizar el usuario root a continuación la ip del servidor y además que no se nos olvide tenemos que decirle que la llave está en la carpeta punto ssh y a continuación le tengo que decir el nombre de la llave privada en este caso hacemos enter y ya estamos en el servidor sin necesidad de haber escrito la contraseña:



## CREAR LA LLAVE SSH DESDE WINDOWS

Un programa popular para hacer SSH en Windows es Putty. Cuando lo instalas viene otro software que nos permite crear claves, llamado Putty Key Generator, bastante sencillo de usar.

Lo encuentras desde el botón de Wondows, buscando luego por Putty, como "PuTTYgen".

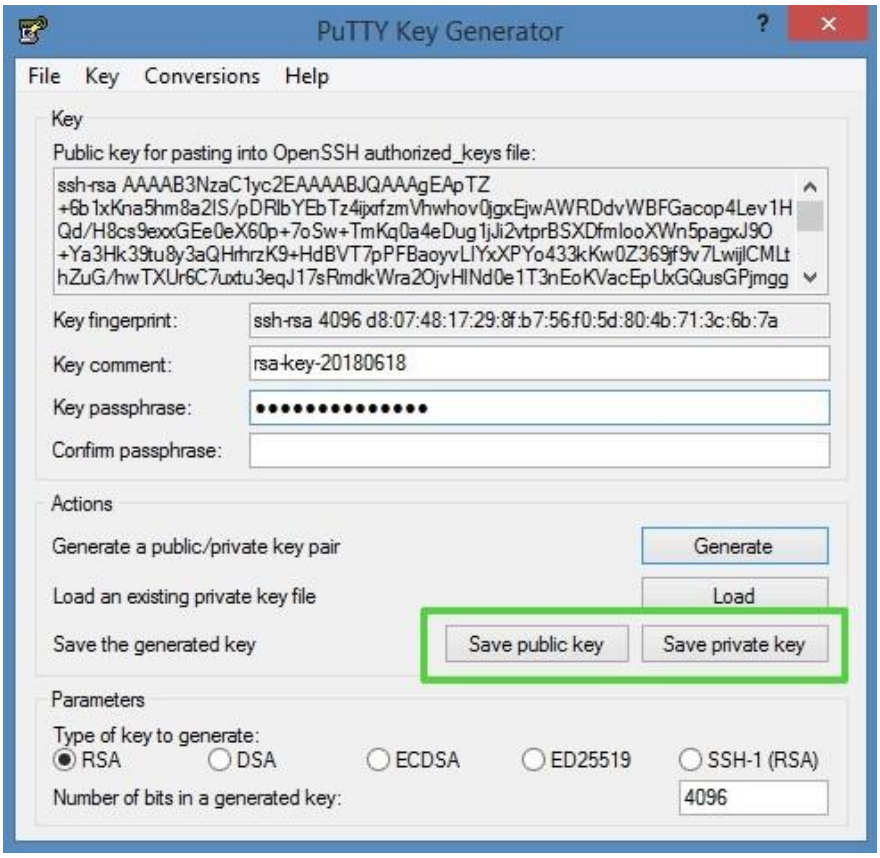
Ya dentro de PuTTYgen, para generar tu llave puedes configurar el tipo y seleccionar el número de bits en la parte de abajo. Generalmente será "RSA" y el valor predeterminado debits es de 2048.

Luego tienes que pulsar el botón "Generate" para comenzar la generación de la clave.

En este momento te pedirá que muevas el puntero del ratón sobre la ventana de Puttygen parair creando una secuencia aleatoria.

Una vez generada la clave te aparecerá la llave pública en la pantalla. Puedes copiar y pegar su contenido y llevártelo a cualquier sitio donde necesites configurar esta llave SSH. También puedes encontrar un botón para salvar tu llave pública en un archivo, en cualquier lugar de tudisco duro.

También tendrás que guardar tu llave privada. Hay otro botón justamente para ello. Simplemente acuérdate donde has dejado cada archivo. Con esto has terminado todo tutrabajo con PuttyGen.



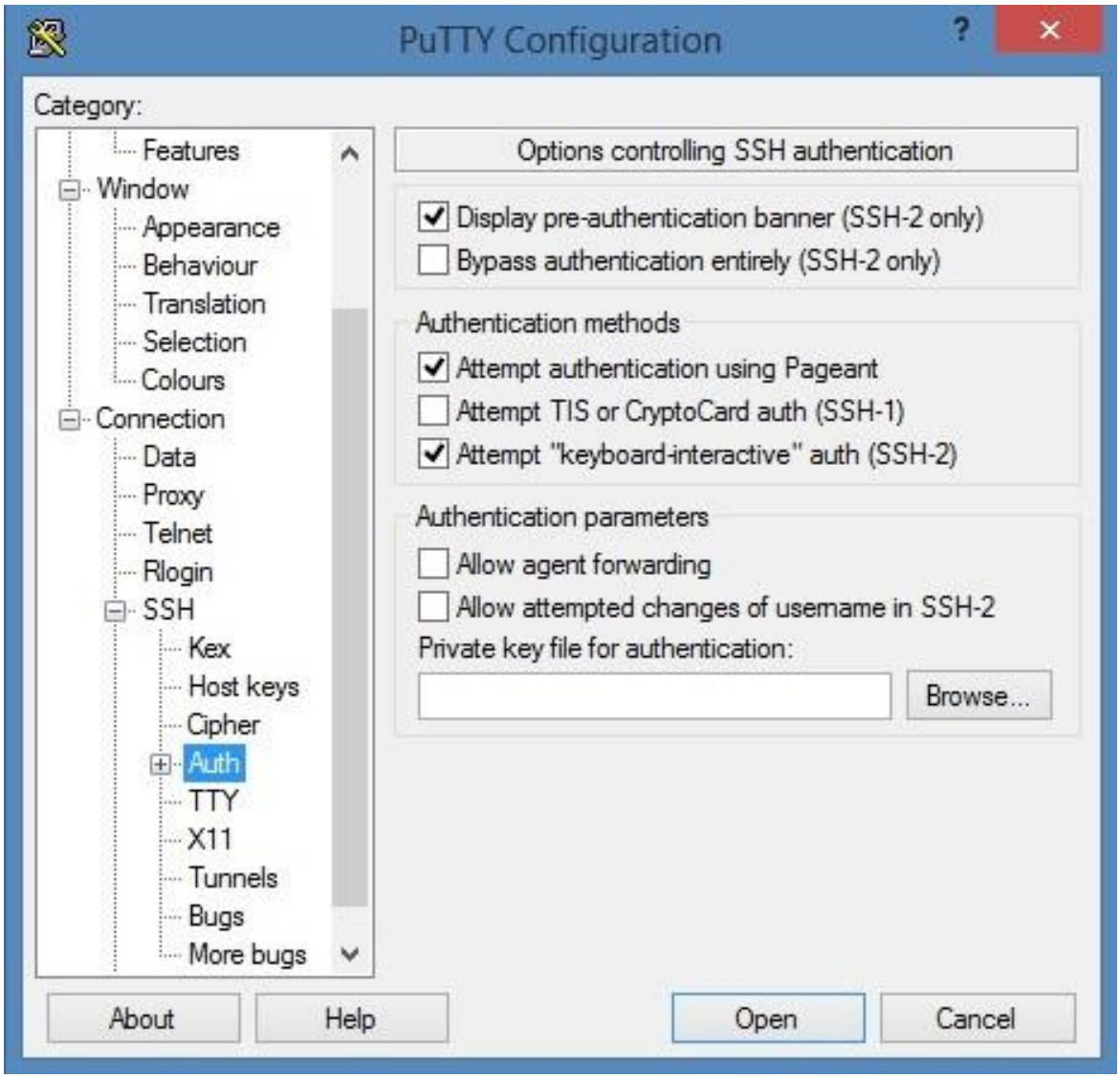
## USAR LA LLAVE PRIVADA

Para acceder a los sitios donde tengas configurada la llave pública generada en el paso anteriortienes que usar tu llave privada. Esto lo hacemos dentro de Putty, el programa que conecta porSSH.

En Putty, en la ventana de configuración es donde debes indicar la llave privada que vas a usarpara conectar por SSH.

Nota: La ventana de configuración es la que te aparece al iniciar Putty. Inicialmente te permite colocar la IP del servidor al que te quieres conectar, o el nombre del dominio del servidor.

Para configurar nuestra clave tenemos que acceder a la configuración "Connection / SSH / Auth". Allí verás un botón que te permite buscar la llave privada del disco duro, para agregarlaa tu inicio de sesión SSH.



## ARCHIVO CONFIG PARA CONFIGURACIÓN ÁGIL DE ACCESOS SSH

Vamos a usar el archivo config que permite la configuración de accesos SSH para un loginrápido al administrar servidores.

En el día a día de la administración de servidores realizamos muchos logueos a diferentes direcciones IP, cada uno de ellos con un usuario distinto, su propia llave SSH, etc. Si manejasmás de un servidor observarás que es imposible memorizar todos estos datos, por lo que generalmente tendrás que ir tirando de notas para acordarte y quizás copia-pegas para poderlanzar más rápido los comandos de consola para la conexión SSH.

Vamos a ver cómo puedes mejorar sensiblemente el flujo de conexión SSHcon los servidores, por medio de un archivo llamado "config", que nos permite tener una especie de "alias" de cada máquina, ayudando en el acceso de una manera muy ágil.


## QUÉ ES EL ARCHIVO CONFIG

El archivo config (config file se le suele llamar en inglés) es un archivo de texto plano, que guardamos en una ruta determinada en nuestro ordenador. Permite especificar tantos hosts de conexión como queramos, asignando un nombre fácil de recordar por nosotros. De este modo, para conectar con los servidores escribiremos el nombre del host en vez de toda la cadena de conexión SSH.

Por ejemplo, este es el comando de conexión habitual que haces en tu terminal para conectartepor SSH a un servidor, usando una llave SSH.

```
ssh -i ~/.ssh/llave_privada el_usuario@0.255.0.1
```

Gracias al archivo config, podemos cambiar este comando de conexión por simplemente algocomo:

```
ssh mi_servidor
```

Obviamente, resulta mucho más fácil de recordar y te ahorrará probablemente copiar y pegarde cualquier lugar donde tengas escrito el comando, equivocarte al escribirlo y, en fin, agilizartu día a dia.

## DÓNDE SE ALMACENA EL ARCHIVO CONFIG

El archivo config lo encuentras en la carpeta .ssh, donde se almacenan todas las llaves ssh que has creado en tu sistema, generalmente en la carpeta de tu usuario y dentro del directorio .ssh. Por ejemplo, la carpeta de mi usuario se llama midesweb. El archivo config lo tengo en Users/midesweb/.ssh

Probablemente, si no lo has creado tú anteriormente, ese fichero "config" no estará en esa ruta.Así que lo tendremos que crear. Es importante saber que el archivo config no lleva ninguna extensión.

Puedes crear el archivo con cualquier editor de tu preferencia. Por supuesto, debe de ser un programa que permita trabajar con archivos de texto plano, como cualquier editor para programadores.

Nota: tener especial cuidado los usuarios de Windows a la hora de crear el archivo, ya que este sistema suele colocar extensiones a los archivos. Generalmente tu editor de programación permite crear archivos con cualquier nombre, por lo que no deberías tener problema. Además, en la configuración del explorador de archivos es recomendable indicar que el sistema no oculte las extensiones de archivos conocidos de manera predeterminada, para poder saber a ciencia cierta que el archivo no tiene tal extensión. Obviamente, si tienesun terminal de línea de comandos avanzado, también puedes crear el archivo con un comando como "touch config", desde la ruta indicada (carpeta ".ssh" en tu usuario).

Los usuarios de Linux y Mac tienen que tener en cuenta que el directorio .ssh es un directorio oculto, por lo que no lo encontraremos si navegamos con el explorador de archivos. Lo más cómodo es llegar a él mediante el terminal.

## CONTENIDO DEL ARCHIVO CONFIG

Ahora veamos el código que tenemos que escribir dentro del archivo config para crear nuestras diferentes conexiones con los servidores que administramos. Básicamente se trata de unas cuantas líneas para cada servidor, que indican los parámetros de conexión que se deben de usar.

```
Host nombre_conexion HostName 1.2.3.4 User root

IdentityFile /Users/midesweb/.ssh/nombre_del_archivo_de_la_clave_privada
```

Como puedes ver, estamos indicando varios parámetros:

- El nombre de la conexión configurada (nombre_conexion)La IP del servidor, en HostName
- El usuario (root, o el que sea en tu caso)
- La ruta para la llave ssh privada que hayas configurado, si es que usas llave ssh, lo quees bastante recomendable y en muchos proveedores incluso obligatorio

Podemos tener varias de estas entradas, con la cantidad de servidores que fuera necesario para nosotros.

## USAR LAS CONEXIONES SSH CONFIGURADAS

Ahora, para el acceso por SSH al servidor, simplemente tenemos que usar las conexiones,desde el terminal de línea de comandos, colocando "ssh" seguido del nombre que hemos indicado para esta conexión en particular.

```
ssh nombre_conexion
```

Listo, con este sencillo comando habremos conseguido conectar por SSH con el servidor, usando nuestra llave, el usuario correcto y la IP. Ya no necesitamos recordar todos esos datosde conexión y podremos acceder al servidor de una manera mucho más ágil.