

Comando scp

Comando scp

Contenido

1. [Introducción](#)
2. [Sintaxis](#)
3. [Ejemplos](#)
4. [Trucos](#)
5. [Windows](#)

Introducción

scp significa *copiado seguro* del inglés *secure copy*, esto quiere decir que se copiará archivos de un ordenador a otro a través de una conexión segura y encriptada.

El comando `scp` utiliza por defecto el puerto 22, y se conecta mediante un enlace encriptado ssh

Se puede utilizar `scp` para copiar archivos de un ordenador local a otro remoto, también se puede copiar del remoto al local y también se puede copiar entre dos remotos, mientras estas conectado a un tercer ordenador, y el tráfico no pasará por el ordenador en que estás.

Se puede usar `scp` en Linux, Mac y Windows.

Sintaxis

Auto (CSS) ▾



```
scp [-12346BCEpqr] [-c cipher] [-F ssh_config] [-i identity_file] [-l limit] [-o ssh_option] [-P port] [-S program] [[user@]host1:]file1 ... [[user@]host2:]file2
```

Explicar la sintaxis completa y cada opción del comando `scp` esta fuera del alcance de este artículo, pero puedes obtener más ayuda usando el siguiente comando, desde cualquier sistema .nix

Auto ▾



```
man scp
```

O, lo puedes leer en línea [aquí](#) (en inglés)

Pero vamos, veamos lo básico al menos.

Auto ▾



```
scp [[user@]host1:]file1 ... [[user@]host2:]file2
```

scp

Es el comando en sí, le indica al sistema operativo que copie uno más archivos por medio de una conexión segura, mejor conocida como conexión ssh

[[user@]host1:]file1

El origen, donde se especifica el archivo o archivos que serán copiados, puede o no contener la información del ordenador remoto, y también puede contener la información del usuario que es dueño del archivo o archivos en el ordenador remoto. Si el usuario no está especificado, entonces por defecto usará los datos del usuario actual en el ordenador donde estás escribiendo el comando. Y si ningún ordenador remoto es especificado, entonces buscará el archivo localmente

[[user@]host2:]file2

El destino, donde se especifica la vía donde los archivos serán copiados, es decir el destino de los mismos, y de nuevo, puede o no contener la información sobre el ordenador remoto y el usuario en ese ordenador. Al igual que fuera explicado arriba, si el usuario no se especifica pero si el nombre o el IP de un ordenador, en ese caso tratará de obtener acceso a ese ordenador remoto usando las credenciales del usuario actual, y si no se especifica ordenador remoto, se asume que el destino de los archivos es local.

Solo para dejar todo claro, se puede no especificar ni ordenador ni usuario remotos, tanto en el origen como en el destino, y en este caso el comando `scp` actuará tal como el comando `cp` y copiará los archivos localmente.

Ejemplos

**Copiar un archivo local a un destino remoto*

Auto (Ruby) ▾



```
scp /ruta/al/archivo-origen usuario@ordenador:/ruta/al/directorio-destino/
```

Así que si queremos copiar el archivo `/home/user/table.csv` a un ordenador remoto llamado `ordenador.ejemplo.com` y archivo a copiar esta en el directorio de jane, debemos usar un comando así.

Auto ▾



```
scp /home/user/table.csv jane@ordenador.ejemplo.com:/home/jane/
```

Copiar un archivo de un ordenador remoto al ordenador local

Auto (Ruby) ▾



```
scp usuario@ordenador:/ruta/al/archivo-origen /ruta/al/directorio-destino
```

Digamos que se quiere copiar el mismo archivo del directorio de jane en `ordenador.ejemplo.com` a tu directorio en el ordenador local

Auto (Swift) ▾



```
scp jane@ordenador.ejemplo.com:/home/jane/table.csv /home/usuario/
```

Copiar un archivo de un ordenador remoto a otro ordenador remoto

Con el comando `scp` se puede copiar archivos entre dos servidores remotos sin necesidad de establecer una conexión permanente con ninguno de ellos, todo el trabajo pesado será realizado por `scp`

Auto ▾



```
scp user1@server1:/path/to/file user2@server2:/path/to/folder/
```

Así, si queremos copiar un archivo del directorio de jane a otro de pete, siendo que ambos archivos están en servidores a los cuales no estamos conectados.

Auto (Swift) ▾



```
scp jane@host.example.com:/home/jane/table.csv  
pete@host2.example.com:/home/pete/
```

Copiar un único archivo de un servidor remoto al mismo servidor remoto, pero en otro directorio

Auto (Swift) ▾



```
scp jane@host.example.com:/home/jane/table.csv  
pete@host.example.com:/home/pete/
```

En esta ocasión se copiará de un ordenador al mismo ordenador sin establecer una conexión permanente con él, es equivalente a establecer una conexión ssh con el servidor, y luego usar el comando `cp` para copiar el archivo.

Copiar todos los archivos de un tipo específico

Auto ▾



```
scp /ruta/al/folder/*.ext usuario@servidor:/ruta/al/folder/
```

Este comando copiará todos los archivos con la extensión `.ext`. Así por ejemplo si queremos copiar todos los archivos de texto `.txt` a un nuevo folder

Auto ▾



```
scp /home/usuario/*.txt jane@host.example.com:/home/jane/
```

Puedes usar comodines en estos comandos tal cual en otros que admiten su uso.

Copiar todos los archivos de un directorio a un servidor remoto

Auto ▾



```
scp /ruta/al/directorio/* usuario@servidor:/ruta/al/directorio/
```

Esto copiará todos los archivos dentro del directorio local al directorio en el servidor remoto. Veamos un ejemplo.

Auto ▾



```
scp /home/user/html/* jane@host.example.com:/home/jane/backup/
```

Todos los archivos .html serán copiados al directorio *backup* en el servidor *host.example.com*

Copiar todos los archivos de un directorio recursivamente al servidor remoto

Auto (Swift) ▾



```
scp -r /home/user/html/* jane@host.example.com:/home/jane/backup/
```

Tal como en el ejemplo anterior, pero en esta ocasión copiará todos los contenidos recursivamente.

Copiar un directorio y todo su contenido a un directorio en el servidor remoto

Auto (Ruby) ▾



```
scp -r /ruta/al/directorio-origen usuario@servidor:/ruta/al/directorio-destino/
```

En esta ocasión el directorio en sí es copiado, incluyendo claro está todo su contenido, y no solo el contenido, como en el caso anterior. Veamos un ejemplo.

Auto ▾



```
scp -r /home/user/html jane@host.example.com:/home/jane/backup/
```

Lo que va a suceder es que el directorio local html, será copiado con su contenido recursivamente al servidor remoto en el directorio `/home/jane/backup/`

Copiar multiples archivos

Se puede copiar multiples archivos sin tener que copiarlos todos, solo basta ponerlos en una lista separada por espacios

Auto (Swift) ▾



```
scp archivo1 archivo2 archivo3  
usuario@servidor.ejemplo.com:/directorio/destino/
```

Si los archivos están en diferentes directorios, solo debes especificar la ruta completa.

Auto (Swift) ▾



```
scp /ruta/archivo1 /ruta/archivo2 /ruta/archivo3  
usuario@servidor.ejemplo.com:/directorio/destino/
```

Trucos

Hemos visto hasta ahora las formas básicas de `scp`, ahora vamos a ver algunos trucos que nos pueden ayudar en alguna oportunidad.

Incrementa la velocidad

`scp` usa AES-128 para encriptar los datos, esto lo hace bastante seguro, pero también algo lento. Si queremos darle un poco más de velocidad y aún tener seguridad, podemos usar Blowfish o RC4.

Para incrementar la velocidad de `scp` se puede cambiar el algoritmo de cifrado del AES-128 que usa por defecto a Blowfish

Auto ▾



```
scp -c blowfish user@server:/home/user/file .
```

O usar RC4 que parece ser el más rápido.

Auto ▾



```
scp -c arcfour user@server:/home/user/file .
```

Esta última no es muy segura, y no debe ser utilizada si la seguridad realmente es algo que le importe.

Incrementa la seguridad

Si la seguridad es lo más importante, se puede incrementar a costa de algo de velocidad.

Auto ▾



```
scp -c 3des user@server:/home/user/file .
```

Limita el ancho de banda

Se puede limitar el ancho de banda a ser utilizado por el comando `scp`

Auto ▾



```
scp -l limit username@server:/home/username/* .
```

El límite esta especificado en Kbits/s, por lo que por ejemplo si se desea limitar a 50 Kbps

Auto ▾



```
scp -l50 user@server:/path/to/file /path/to/folder
```

Ahorra datos

Igualmente es posible ahorrar en consumo de datos, esto se consigue comprimiendo los archivos a ser enviados.

Auto ▾



```
scp -C user@server:/path/to/file /path/to/folder
```

Definir IPv4 o IPv6

Si se desea forzar el uso de IPv4 o de IPv6 también es posible, veamos estos ejemplos

Auto ▾



```
scp -4 user@server:/path/to/file /path/to/folder
```

El ejemplo de arriba usa IPv4 y el de abajo IPv6

Auto ▾



```
scp -6 user@server:/path/to/file /path/to/folder
```

Especifica el puerto

En caso que el servidor remoto no este escuchando en el puerto que usa ssh por defecto que es el 22, se puede hacer que `scp` use otro puerto para establecer la conexión con el ordenador remoto.

Auto ▾



```
scp -P [port] [user]@[server]:[path/to/]file [/path/to/]file
```

Usando la P mayúscula se puede especificar el puerto que `scp` usará para conectarse, en el ejemplo de abajo podemos ver que se esta indicando a `scp` a usar el puerto 2222

Auto ▾



```
scp -P 2222 user@server:/home/jane/file /home/jane/
```

Mostrando Detalles

Si quieres ver lo que pasa debajo de la superficie puedes usar el parámetro `-v`

Auto ▾



```
scp -v user@server:/home/jane/file /home/jane/
```

Windows

En caso de estar trabajando en un ordenador que estuviera usando Windows en lugar de Linux o algún otro derivado de Unix como ser algún BSD, aún puedes disfrutar de `scp` en varias formas, línea de comandos o herramientas gráficas.

pscp

pscp es un comando de terminal que funciona en Windows, casi de la misma manera que el comando **scp** de los sistemas operativos tipo Unix. Lo primero que se debe hacer es bajar el software desde [aquí](#), o usando este link [directo](#)

Una vez descargado puede ser invocado desde la terminal de comandos de Windows, para hacer eso, ir a menú inicio, luego *ejecutar* y escribir

Auto ▾



```
cmd
```

Luego presionar ENTER, para Windows 8.x, presionar la tecla Windows y hacer click en la lupa, escribir **cmd** y presionar ENTER.

Una vez en la línea de comandos, asegurarse de estar en el directorio donde **pscp** fuera descargados, o agregar dicho directorio a la variable PATH, supongamos que esta en el folder Downloads, se debe ejecutar este comando.

Auto ▾



```
SET PATH=C:\Users\Guillermo\Downloads;%PATH%
```

Debe cambiarse el usuario de acuerdo a la configuración particular de cada PC.

Este comando deberá ser ejecutado cada vez que se reinicie el ordenador, o configurarlo de manera permanente, el como hacerlo esta fuera del alcance de este artículo.

Abajo, se pueden ver las opciones del comando.

Auto (Visual Basic .NET) ▾



```
PuTTY Secure Copy client
```

```
Release 0.63
```

```
Usage: pscp [options] [user@]host:source target  
       pscp [options] source [source ... ] [user@]host:target  
       pscp [options] -ls [user@]host:filespec
```

Options:

```

-V          print version information and exit
-pgpfp     print PGP key fingerprints and exit
-p         preserve file attributes
-q         quiet, don't show statistics
-r         copy directories recursively
-v         show verbose messages
-load sessname Load settings from saved session
-P port    connect to specified port
-l user    connect with specified username
-pw passw  login with specified password
-1 -2     force use of particular SSH protocol version
-4 -6     force use of IPv4 or IPv6
-C        enable compression
-i key     private key file for authentication
-noagent   disable use of Pageant
-agent     enable use of Pageant
-batch     disable all interactive prompts
-unsafe    allow server-side wildcards (DANGEROUS)
-sftp     force use of SFTP protocol
-scp      force use of SCP protocol

```

Copiar archivos de Windows a Linux

Se puede utilizar `scp` para copiar archivos de Windows a Linux

Auto ▾



```
pscp c:\path\to\file user@remote-server:/path/to/remote-folder
```

Copiar archivos de Linux a Windows

Igualmente se puede copiar archivos de Linux (FreeBSD o Max OS X) a Windows.

Auto (Lua) ▾



```
pscp user@remote-server:/path/to/remote-file c:\path\to\local-folder\
```

Especificar un protocolo

Se puede especificar el protocolo que scp va a utilizar en cada conexión.

-scp

Forzará a pscp a utilizar el protocolo scp

-sftp

Esto forzará a pscp a utilizar el protocolo sftp

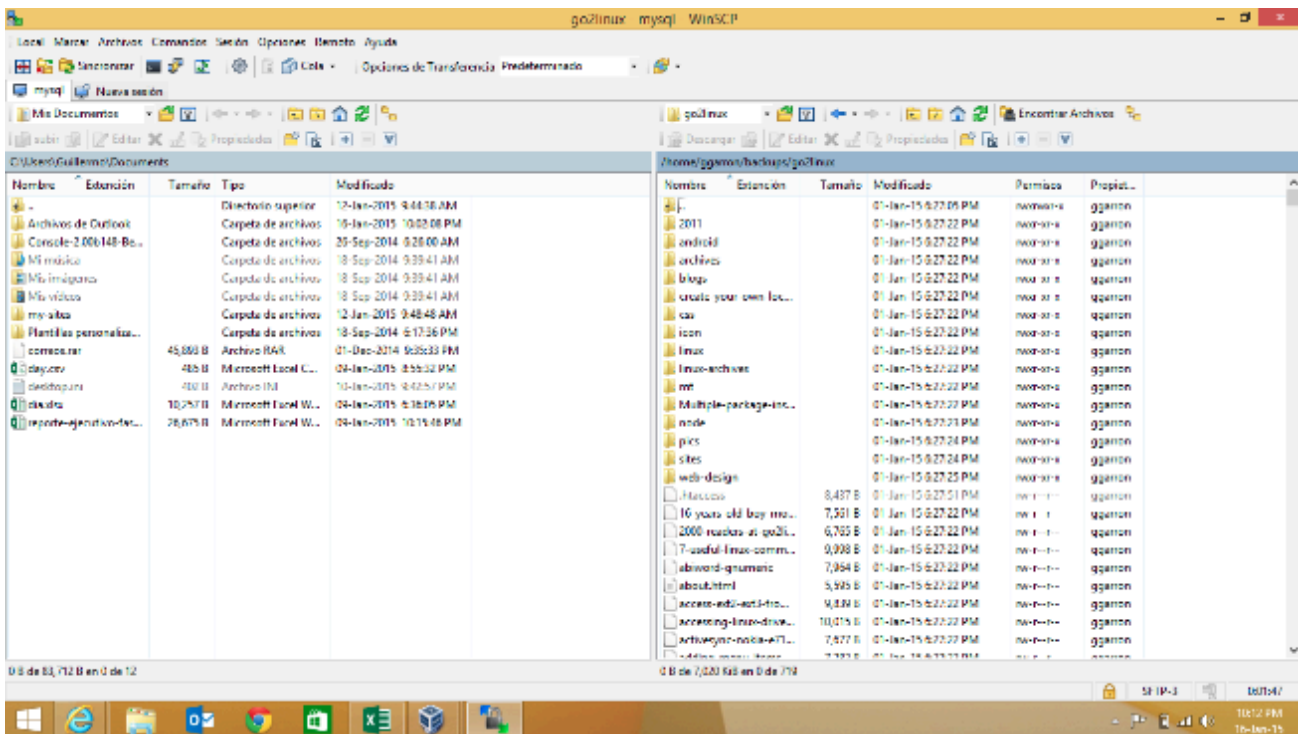
WinSCP

WinSCP es la versión gráfica de scp, principalmente utilizada en Windows, su desarrollo comenzó en el 2.000 por **Martin Prikryl** y está licenciado bajo la licencia GNU, por lo tanto es software libre

WinSCP no es solo un cliente SCP, también es un cliente SFTP

Capturas de Pantalla

Abajo hay unas capturas de pantalla



Principales Características

Algunas de sus principales características son:

Sincronismo total

Esta característica mantendrá ambos lados de la conexión en sincronía, por lo que cada cambio realizado en cualquiera de los lados será actualizado en el otro lado. La aplicación debe mantenerse activa, y conectada para mantener esta funcionalidad activa

Sincronismo remoto

En este caso se mantiene en sincronía el lado remoto con el local, es decir que cualquier cambio en el lado local se reflejará en el lado remoto, pero no al revés

Se puede decidir si los archivos del lado remoto serán borrados cuando se borren en el lado local al iniciar la conexión. Si se activa dicha funcionalidad, cualquier archivo borrado en el lado local será borrado en el remoto, pero si no, solo los archivos agregados en el lado local se crearán en el remoto y los cambios en los archivos existentes se reflejarán en el remoto

Arrastrar y Soltar

Es posible arrastrar y soltar del lado local al remoto y viceversa, para copiar archivos de uno de los lados al otro

Como se puede ver, se obtiene la misma funcionalidad básica para Windows que los usuarios de Linux o Mac OS X ya disfrutaban, y puede que esta sea la forma más fácil de copiar archivos de Windows a Linux y de Linux a Windows.

Si tienes sugerencias o dudas, puedes hacerlas [aquí](#)

Editado por última vez el: 9 de mayo de 2015

Por: Guillermo Garron