

rsync, copias de seguridad remotas



Todos sabemos que si queremos mantener nuestros datos seguros, necesitamos crear una copia de seguridad de ellos periódicamente. Si esta tarea se puede hacer automáticamente pues todavía mejor. Voy a mostraros cómo hacer esto, con dos ordenadores con conexión remota mediante **ssh**, utilizando **rsync** para sincronizar carpetas y **cron** para automatizar la tarea. Para este caso vamos a utilizar dos equipos con **Ubuntu**, pero esto es indiferente, ya que funciona exactamente igual en cualquier Sistema Operativo **GNU/Linux**.

Empezamos instalando todo lo necesario.

```
# apt-get install ssh rsync cron
```

Primero comprobamos que podemos conectarnos desde nuestro ordenador cliente al servidor: tan sencillo como abrir el terminal y escribir:

```
ssh usuario@IP_servidor
```

En mi caso,

```
ssh hispalis70@192.168.1.4
```

En el caso de que algo falle, puedes comprobar los errores con:

```
ssh -v usuario@IPservidor
```

Os recuerdo que la IP interna la podemos conocer con el [comando ifconfig](#).

Una vez que sabemos que todo está correcto, salimos de la sesión con la orden **exit**.

Como vamos a hacer que la tarea se ejecute automáticamente, debemos evitar que al conectar por **ssh** nos pida la contraseña. Para ello, seguimos estos pasos:

Desde nuestro equipo cliente, generamos una clave pública:

```
ssh-keygen -t rsa
```

Te hará las siguientes preguntas:

Enter file in which to save the key (/home/hispalis70/.ssh/id_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Sólo pulsa **Enter** en todas y obtendrás un resultado parecido a éste.

Your identification has been saved in /home/hispalis70/.ssh/id_rsa.

Your public key has been saved in /home/hispalis70/.ssh/id_rsa.pub.

The key fingerprint is:

61:c9:a6:db:82:84:5f:16:fa:73:c9:1e:d3:5f:72:e3 hispalis70@hispalis70-pc

Ahora tendremos que copiar dicha clave a nuestro ordenador servidor:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub usuario@IP_servidor
```

Quiero aclarar que por no tener que escribir una contraseña tu red no es menos segura, todo lo contrario, ya que la contraseña generada aleatoriamente es muy complicada y sólo sirve para el usuario de ese ordenador. Otra cosa a tener en cuenta es que si los pasos anteriores los hacemos como administrador (root), todavía tendremos más seguridad, pero será necesario abrir la sesión como tal, para la sincronización.

Una vez hecho esto, ya podemos empezar con la [sincronización](#) de carpetas.

Tomando como ejemplo mi propio /home, voy a explicar qué hacer para tener una copia

de seguridad de dicha carpeta y que se actualice automáticamente cuando se le indique. Este método sólo actualiza las novedades en la carpeta, evitando el engorroso trabajo de copiarlo todo de nuevo.

A continuación, vamos a utilizar **cron** y [rsync](#) para que la sincronización se realice automáticamente. La sintaxis de **cron** es bastante sencilla:

```
.----- minuto (0 - 59)
| .----- hora (0 - 23)
| | .----- día del mes (1 - 31)
| | | .----- mes (1 - 12)
| | | | .-- día de la semana (0 - 6) (Domingo=0 o 7)
| | | | |
* * * * * tarea a ejecutar
```

Donde 'tarea a ejecutar', será el comando [rsync](#), que utilizamos para sincronizar carpetas, siendo ésta su sintaxis más básica:

rsync [dir_origen] [dir_destino]

Una vez visto esto, vamos a empezar...

Lo primero que tenemos que hacer es editar el fichero de **cron**. En la consola escribimos:

```
crontab -e
```

En el archivo editado agregamos la siguiente línea (es sólo una línea):

```
* 2 * * 0 rsync -r -delete
/home/hispalis70/hispalis70@192.168.1.4:/home/hispalis70/home2
```

Nota: delante de delete son dos guiones seguidos

Con esto estamos diciendo que la copia de /home se ejecute todos los domingos a las 2 de la mañana.

Con la variable **-r** la sincronización se realizará recursivamente, haciendo un recorrido del origen y creando los subdirectorios necesarios en el destino.

Con **-delete** se eliminarán en el equipo remoto los archivos que se borren en el origen.

Como podéis ver, estoy copiando mi [/home](#) al ordenador servidor, en el directorio /home2. Dicho directorio no hace falta crearlo previamente, ya que se crea automáticamente la primera vez.

Estos comandos pueden dar mucho más juego, y podéis encontrar información de todos ellos con el comando **man**.

Si tenéis alguna duda al respecto, podéis consultarnos en nuestro [foro de ayuda](#).