



Backup en Linux

■ **Duración:** ⌚ 20 minutos ■ **Dificultad:** 😊 Media

Objetivo. Hacer backup mediante comandos.

Material. Servidor Ubuntu o máquina Lubuntu

Para Linux vamos a probar la herramienta rsync. Es una herramienta sencilla pero muy potente. Básicamente nos permite sincronizar directorios en una misma máquina o entre dos máquinas.

1. Entramos en nuestro Ubuntu Server y nos ponemos en modo super-usuario con `sudo -i`.
2. Nos vamos al `/tmp` y creamos un directorio llamado `original` con un fichero `hola.txt`.
3. Ahora creamos una copia de `original` mediante el comando:

```
# rsync -av /tmp/original /tmp/copia
```

Como se ve en la Figura 4.81, la herramienta nos avisa de que va a crear el directorio `/tmp/copia` (no lo habíamos creado) y muestra los ficheros que ha traspasado y un resumen de bytes transferidos. El directorio `copia` reproduce la estructura de carpetas de `original`, no solo los ficheros.

```
root@ubuntu12:/tmp# rsync -av /tmp/original /tmp/copia
sending incremental file list
created directory /tmp/copia
original/
original/hola.txt

sent 115 bytes received 35 bytes 300.00 bytes/sec
total size is 5 speedup is 0.03
root@ubuntu12:/tmp# find original
original
original/hola.txt
root@ubuntu12:/tmp# find copia
copia
copia/original
copia/original/hola.txt
root@ubuntu12:/tmp#
```

Fig. 4.81. *rsync de directorios.*

4. ¿En qué se diferencia de hacer una copia normal mediante `cp`? Pues en que `rsync` no copia todo, sino solo los ficheros nuevos o los que han cambiado. Por ejemplo, creamos un fichero nuevo y sincronizamos. Solo se traspasa ese fichero (Fig. 4.82).

```
root@ubuntu12:/tmp# date > original/adios.txt
root@ubuntu12:/tmp# rsync -av /tmp/original /tmp/copia
sending incremental file list
original/
original/adios.txt

sent 159 bytes received 35 bytes 388.00 bytes/sec
total size is 35 speedup is 0.18
root@ubuntu12:/tmp# find /tmp/copia
/tmp/copia
/tmp/copia/original
/tmp/copia/original/adios.txt
/tmp/copia/original/hola.txt
root@ubuntu12:/tmp#
```

Fig. 4.82. *Solo copiamos los cambios.*

5. Si hemos borrado un fichero en el `original` y queremos que se actualice la copia, hay que incluir el parámetro `--delete`.

6. Con lo que hemos visto hasta ahora solo podemos hacer backups completos. El directorio `copia` lo podemos llevar a cualquier dispositivo extraíble o podría ser un disco en red. Para hacer backups incrementales ejecutaremos esto:

```
# rsync --avvb --delete --backup dir=/tmp/
backup1 /tmp/original /tmp/copia
```

Esta vez la sincronización deja en el directorio `/tmp/backup1` los ficheros que resultan modificados o eliminados; en `/tmp/copia` siempre está la versión actual. En nuestro ejemplo vamos a borrar el fichero `adios.txt` y al sincronizar vemos que ya no está en `original` ni en `copia`, pero sí en `backup1` (Fig. 4.83).

```
root@ubuntu12:/tmp# rm original/adios.txt
root@ubuntu12:/tmp# rsync --avvb --delete --backup-dir=/tmp/backup1 --delete /tmp/original
/tmp/copia
backup-dir is /tmp/backup1/
sending incremental file list
delta transmission disabled for local transfer or --whole-file
original/
backed up original/adios.txt to /tmp/backup1/original/adios.txt
selecting original/adios.txt
total: matches=0 hash=0 false alarms=0 data=0
sent 44 bytes received 16 bytes 120.00 bytes/sec
total size is 0 speedup is 0.00
root@ubuntu12:/tmp# find original copia backup1
original
copia
copia/original
backup1
backup1/original
backup1/original/adios.txt
root@ubuntu12:/tmp#
```

Fig. 4.83. *Backup incremental.*

7. Lo mismo ocurre para ficheros modificados. Vamos a crear un fichero en `original`, sincronizamos, modificamos ese fichero y al volver a sincronizar, además de actualizar `copia`, se guarda la versión anterior en `backup1`.
8. Finalmente, como es imprescindible que el backup se ejecute con regularidad, vamos a probar a meterlo en el cron. Le pondremos que se ejecute cada minuto (lo normal sería una vez al día), y en ese tiempo haremos cambios para comprobar el funcionamiento.

Como se ve en la Figura 4.84, hemos creado un script `mibackup.sh` que se invoca desde el cron. Para distinguir las distintas copias incrementales, el script utiliza la fecha en que se ejecuta. Dejamos un log para comprobar diariamente que todo ha ido bien.

```
# cat mibackup.sh
FECHA=`date +%y%m%d%H%M`
rsync --avvb --backup-dir=/tmp/backup_
$FECHA --delete
/tmp/original /tmp/copia >> /tmp/log_
$FECHA
# crontab -l
# m h dom mon dow command
* * * * * /tmp/mibackup.sh
```

```
root@ubuntu12:/tmp# cat mibackup.sh
FECHA=`date +%y%m%d%H%M`
rsync --avvb --backup-dir=/tmp/backup_$FECHA --delete /tmp/original /tmp/copia >>
/tmp/log_$FECHA
root@ubuntu12:/tmp# crontab -l
* * * * * /tmp/mibackup.sh
root@ubuntu12:/tmp# ls -l
total 32
drwxr-xr-x 3 root root 4096 2012-06-05 13:04 backup_1206051304
drwxr-xr-x 3 root root 4096 2012-06-05 13:52 copia
-rw-r--r-- 1 root root 297 2012-06-05 13:02 log_1206051302
-rw-r--r-- 1 root root 324 2012-06-05 13:03 log_1206051303
-rw-r--r-- 1 root root 397 2012-06-05 13:04 log_1206051304
-rw-r--r-- 1 root root 297 2012-06-05 13:05 log_1206051305
-rwxr-xr-x 1 root root 122 2012-06-05 13:00 mibackup.sh
drwxr-xr-x 2 root root 4096 2012-06-05 13:03 original
root@ubuntu12:/tmp#
```

Fig. 4.84. *Backup incremental programado.*