Rsync es una potente herramienta de Linux para transferir y sincronizar archivos y directorios en el mismo ordenador o entre varios a través de la red. Usa un algoritmo delta que compara los ficheros en el origen y el destino, y en caso de que se haya modificado, en lugar de copiar el fichero entero, copia solo los cambios que tenga respecto al destino. Veamos ejemplos de uso típicos:

Contenidos [hide]

- 1 Copia normal
- 2 Reducir tamaño de datos durante la copia
- 3 Otros parámetros interesantes
- 4 Copias incrementales
- 5 Copias en red
- 6 Interfaz gráfica

Copia normal

El siguiente ejemplo sería la forma más básica de usar rsync:

```
rsync -avh /origen /destino
```

Este comando copiará la carpeta origen y todo su contenido en la carpeta destino. Si quisieramos copiar solo el contenido y no la carpeta padre, pondriamos la barra al final **/origen/**.

Los parámetros usados en el ejemplo son:

- **-a**: modo archivo, imprescindible para backups, hace copia recursiva, copia enlaces simbólicos y conserva en el destino los permisos, propietario y fecha de modificación de los ficheros. Equivale a las opciones -rlptgoD
- **-v**: modo verbose, muestra información del proceso durante la copia. Cuantas más **v** pongamos, más información, aunque con una sola suele ser suficiente.
- **-h**: human-readable, para mostrar la información del proceso en formato más entendible (GB y MB en lugar de bytes).

Si ejecutamos el comando una segunda vez sobre las mismas carpetas, solo copiará los ficheros que hayan sido modificados.

Reducir tamaño de datos durante la copia

Rsync dispone de un par de parámetros interesantes para reducir el tamaño de los ficheros que se copian:

- **-z**: comprime el fichero durante la transferencia.

- **-W:** –whole-file: Con esta opción, al realizar las copias se enviará el fichero completo en lugar de solo sus modificaciones. Esta es la opción por defecto cuando al hacer copias el origen y el destino están especificados como discos locales (por ejemplo, copias entre el mismo ordenador, o copiar a una carpeta en la que montamos una carpeta compartida de otro equipo). Con esta opción la transferencia puede ser más rápida si el ancho de banda entre las máquinas de origen y destino es mayor que el ancho de banda del disco.
- **–no-whole-file**: En lugar de copiar un fichero modificado entero, copiará solo las diferencias entre el origen y el destino. Es la opción por defecto al hacer copias en red con la opción -e (ver más adelante).

Hay que tener en cuenta que estos parámetros no reducen necesariamente el tiempo de la copia ya que, aunque en algunos casos copiemos menos cantidad de MBs, también se tarda tiempo en comprimir o calcular las diferencias entre archivos.

Sí pueden servirnos para ahorrar ancho de banda en copias en red.

Otros parámetros interesantes

- **-u**: modo update, evita que se sobrescriban los ficheros del destino que son más recientes que los del origen. Útil si hemos trabajado directamente sobre las copias de seguridad.
- **-c**: por defecto rsync comprueba si un fichero se ha actualizado comparando su tamaño y fecha de modificación. Con este parámetro en lugar de eso comprobará el checksum de los ficheros.
- **-n**: modo de prueba, simula la copia pero no hace ningún cambio en el sistema de ficheros.
- **–delete**: borra en el destino los ficheros o carpetas que ya no están en el origen. Recomendable probar primero con el modo de prueba. Podemos decidir si los ficheros se borrarán antes, durante, o después de la transferencia. Esto es interesante por si se cancelase la transferencia a la mitad:
    - **–delete-before**: Es la opción por defecto. rsync borra los archivos antes del proceso de transferencia
    - **–delete-after**: rsync borra los archivos después del proceso de transferencia
    - **–delete-during**: rsync borra los archivos durante el proceso de transferencia
    - **–delete-delay**: rsync localizará los ficheros a borrar durante la transferencia, pero los borrará cuando ésta haya terminado.
- **-A**: Preservar ACLs.
- **-X**: Preservar atributos extendidos.
- También hay algunos parámetros de exclusión de ficheros que no detallaré en este artículo por no haberlos probado:
    - **–exclude=**PATTERN exclude files matching PATTERN
    - **–exclude-from=**FILE read exclude patterns from FILE
    - **–include=**PATTERN don’t exclude files matching PATTERN
    - **–include-from=**FILE read include patterns from FILE
    - **–files-from=**FILE read list of source-file names from FILE

Copias incrementales

- **-b**: modo backup. Con esta opción, en lugar de sobrescribir directamente en el destino los ficheros modificados, guarda primero una copia de su versión anterior en la carpeta de destino, del mismo nombre pero terminada en ~.
- **–suffix**: indica el sufijo que queremos que aparezca en las versiones anteriores guardadas en modo backup, en lugar del ~ por defecto. Podemos indicar, por ejemplo, la fecha actual:
  ```
  rsync -avhb --delete –suffix=.$(date +%d%m%Y) origen/ destino/
  ```
- **–backup-dir**: indicamos el directorio donde nos guardará los ficheros modificados. Importante indicarlo fuera de la carpeta de destino porque sino se borrará si usamos **–delete**.
  ```
  rsync -avhb --delete --backup-dir=/destino/back_$(date
  +%d%m%Y%H%M) origen/ destino/
  ```

Al guardarlos en otra carpeta, no añadirá ningún sufijo a los archivos.

Con el comando anterior tendremos una copia espejo de origen en destino, y, aparte, carpetas diarias que contendran la versión anterior de los ficheros modificados ese día.

Otros parámetros interesantes a la hora de hacer backups:

- **–compare-dest**=DIR also compare received files relative to DIR
- **–copy-dest**=DIR … and include copies of unchanged files
- **–link-dest**=DIR hardlink to files in DIR when unchanged

Con **-b** y **–backup-dir=DIR**, tienes una copia de la última versión en destino, y las versiones anteriores en DIR. Con **–compare-dest=DIR** en destino solo tendrás los ficheros que han cambiado respecto a DIR, que sería el último backup.

Como variación de la **–compare-dest**, está la **–copy-dest=DIR**, que hace lo mismo que la anterior pero además copia los ficheros que no hayan cambiado. ¡Pero ojo!, la ventaja de esta opción es que la copia de los ficheros que no han cambiado es local, de forma que en un rsync remoto, todos los ficheros que ya estaban en DIR no se tienen que transferir. La desventaja es que necesita mucho espacio en disco.

Finalmente, tenemos **–link-dest=DIR**, que hace lo mismo que **–copy-dest** pero usando hard links a los ficheros que ya existen, de manera que no consume apenas espacio en disco.

Copias en red

Con rsync también podemos hacer copias en red a través de SSH. Las copias mediante SSH van cifradas, por lo que son seguras. Veamos un ejemplo:

```
rsync -e ssh -avhP /origen usuario@192.168.1.50:/destino
```

El formato del comando es igual, teniendo en cuenta que:

- **-e**: permite seleccionar un shell diferente para copias remotas, por defecto es SSH.
- **-P**: Útil para copias en red. Si por ejemplo se corta la copia de un fichero muy grande a la mitad, al reanudarla solo copiará lo que resta del fichero, no el fichero completo.
- El destino viene indicado en el formato **usuario@IP:/ruta**. Debemos tener en cuenta que nos pedirá la contraseña del usuario del equipo de destino. Si queremos que no la pida debemos usar la Autentificación trasparente por clave pública/privada con OpenSSH.

También podemos hacer copias entre servidores usando el propio protocolo de rsync, para lo que necesitaremos en el servidor remoto tener instalado rsync como daemon y abierto el puerto 873:

```
rsync -avhP /origen rsync://usuario@servidor_remoto:/carpeta_destino
```

Ojo: las copias usando este protocolo no son seguras porque no van cifradas. Podemos usar el protocolo rsync para copias a través de red local y SSH para copias a través de internet.

Interfaz gráfica

Grsync proporciona una interfaz gráfica para rsync y podemos usarlo tanto en linux como en Windows y MacOS