

# Sincronizar carpetas remotas con rsync

Rsync viene incluida en casi todas las distribuciones Linux. SE puede usar como herramienta avanzada de copiado, permitiéndonos copiar archivos localmente o remotamente. También puede ser utilizada como herramienta de backups, soporta la creación de backups incrementales.

Rsync utiliza el algoritmo delta-transfer que nos permite transferir archivos nuevos, así como los cambios recientes a archivos existentes, mientras ignora archivos que no han sido modificados. El uso de rsync puede ser personalizado, ayudándonos a automatizar los respaldos, así mismo puede ser utilizado como un servicio para convertir su ordenador en un servidor y permitir que otros clientes de rsync se conecten a él.

Además del copiado de archivos y carpetas locales, rsync nos permite copiar mediante SSH (Secure Shell), RSH (Remote Shell) y puede ejecutarse como servicio en un ordenador y permitir a otros conectarse, cuando rsync es usado como servicio escucha el puerto 873.

Cuando utilizamos rsync como servicio o cuando utilizamos RSH, los datos enviados entre pc's no viajan cifrados, por consiguiente, si estás transfiriendo archivos entre pc's en la misma red local, esto es útil, pero esto no debe ser utilizado para transferir archivos en redes inseguras, como lo es Internet. Para este propósito debes usarlo con SSH.

## Ejemplos de uso:

Copiar contenidos de una carpeta en otra, reemplazando los archivos de la carpeta de destino.

```
rsync -rtv carpeta_origen/ carpeta_destino/
```

El parámetro -r significa que hará el copiado recursivo de carpetas, es decir, creará toda la estructura de las carpetas que hay dentro de carpeta\_origen en carpeta\_destino, y copiará todos los archivos que se encuentren dentro de estas.

El parámetro -t hace que rsync preserve los tiempos de modificación de los archivos que copia desde la carpeta origen.

El parámetro -v significa que la información imprimida durante la ejecución del programa será mucho más detallada, podemos utilizar esto para ver el progreso del comando.

El parámetro -a también hace la copia recursiva y preserva los tiempos de modificación, pero adicionalmente copia los vínculos simbólicos que encuentra como vínculos simbólicos, preserva los permisos, preserva la información del dueño y el grupo del archivo, y preserva los archivos de dispositivo y los archivos especiales. Esto es útil si se está copiando completo el directorio personal de un usuario, o si se están copiando carpetas del sistema a otro lugar.

```
rsync -av origen/ destino/
```

Para actualizar el contenido de una carpeta agregamos el parámetro -u a rsync, esto actualizará los contenidos de la carpeta destino en base a la carpeta de origen, aquí es donde el algoritmo delta-transfer actúa.

```
rsync -rtvu carpeta_origen/ carpeta_destino/
```

Para mantener dos carpetas sincronizadas, no solo necesitamos agregar los nuevos archivos en la carpeta de origen a la carpeta de destino, también necesitamos remover los archivos que fueron borrados de la carpeta origen en la carpeta destino. rsync nos permite hacer esto con el parámetro --delete, esto usado en conjunto con el parámetro previo -u que actualiza los archivos modificados nos permite mantener dos carpetas sincronizadas.

```
rsync -rtvu --delete carpeta_origen/ carpeta_destino/
```

El proceso de borrado se puede realizar antes, durante y después de la transferencia, podemos configurar cuando lo realizamos utilizando alguno de los parámetros siguientes:

- rsync puede buscar los archivos faltantes y borrarlos antes de del proceso de transferencia, este es el comportamiento por default y puede ser ajustado con --delete-before
- rsync puede buscar los archivos faltantes y borrarlos después que la transferencia se ha completado, con el parámetro --delete-after
- rsync puede borrar los archivos durante la transferencia, cuando se encuentra un archivo faltante, se borra en ese momento, activamos este rsync puede hacer la transferencia y encontrar los archivos faltantes durante este proceso, esperar hasta que ha terminado y borrar los archivos que encontró después, esto puede lograrse con --delete-delay

Ejemplo:

```
rsync -rtvu --delete-delay carpeta_origen/ carpeta_destino/
```

Para comprimir la información, podemos agregar el parámetro -z a rsync.

```
rsync -rvz carpeta_origen/ carpeta_destino/
```

### [Transferir archivos entre dos sistemas remotos](#)

Para transferir archivos entre el local y un server remoto, necesitamos especificar la dirección del pc remoto, puede ser un nombre de dominio, una dirección IP, o el nombre de un servidor que hayamos guardado previamente en nuestro archivo de configuración de SSH, seguido de dos puntos, y la carpeta que queremos usar para la transferencia. Rsync no puede transferir archivos entre dos sistemas remotos, solo una carpeta local o una carpeta remota pueden ser utilizadas en conjunto con una carpeta local. Para hacer esto utilizamos:

Carpeta local a carpeta remota, utilizando un dominio, una dirección de IP o un servidor definido en el archivo de configuración de SSH:

```
rsync -rtvz carpeta_origen/ usuario@dominio:/ruta/a/carpeta_destino/
```

```
rsync -rtvz carpeta_origen/ usuario@xxx.xxx.xxx.xxx/ruta/a/carpeta_destino/  
rsync -rtvz carpeta_origen/ nombre_servidor:/ruta/a/carpeta_destino/
```

Carpeta remota a carpeta local, utilizando un dominio, una dirección de IP o un servidor definido en el archivo de configuración de SSH:

```
rsync -rtvz usuario@dominio:/ruta/a/carpeta_origen/ carpeta_destino/
```

```
rsync -rtvz usuario@xxx.xxx.xxx.xxx:/ruta/a/carpeta_origen/ carpeta_destino/  
rsync -rtvz nombre_servidor:/ruta/a/carpeta_origen/ carpeta_destino/
```

### [Excluyendo archivos y directorios](#)

Hay casos en los que necesitamos excluir ciertos archivos y directorios de rsync, un caso común es cuando sincronizamos un proyecto local con un repositorio remoto o incluso con un sitio web, en este caso podríamos querer excluir algunos directorios de desarrollo y algunos archivos ocultos de ser transferidos al sitio vivo. Excluir archivos puede ser hecho con `--exclude` seguido del directorio o el archivo que queremos excluir.

```
rsync -rtv --exclude 'directorio' carpeta_origen/ carpeta_destino/  
rsync -rtv --exclude 'archivo.txt' carpeta_origen/ carpeta_destino/  
rsync -rtv --exclude 'direccion/a/directorio' carpeta_origen/ carpeta_destino/  
rsync -rtv --exclude 'direccion/a/archivo.txt' carpeta_origen/ carpeta_destino/
```

### [Parámetros adicionales de rsync](#)

- `-t` Preserva los tiempos de modificación de los archivos que están siendo transferidos.
- `-q` Suprime todos los mensajes que no sean de error, este parámetro es contrario al parámetro `-v` que muestra mucha más información de la transferencia.
- `-d` Copia los archivos de un directorio sin utilizar recursividad para copiar los directorios internos, en decir, solo los archivos son transferidos.
- `-l` Copia los symlinks como symlinks
- `-L` Copia los archivos a los que un symlink está apuntando cuando encuentre un symlink.
- `-W` Copia archivos enteros, ya que cuando utilizamos el algoritmo de delta-transfer solo se copia la parte de un archivo que fue actualizada, algunas veces esto no es deseado.
- `--progress` Muestra el progreso de los archivos que están siendo transferidos.

- -h Muestra la información que provee rsync en un formato más legible, las cantidades son dadas en K's, M's, G's, y así sucesivamente.

Rsync con ssh, o sea para que no pida contraseñas, es decir, con certificados

```
$ rsync -e ssh -avzn usuario\_remoto@192.168.0.100:/carpeta\_origen carpeta_destino
```

-a para copiar recursivamente manteniendo privilegios, fecha de creación, permisos, etc.; -v para incrementar el nivel de detalle de la operación; -z para comprimir los datos, así la transferencia es más rápida; -e ssh para usar ssh para copiar los archivos de un servidor a otro.

Puedo usar también --delete-after para borrar los archivos que hayan sido borrados antes de realizar el backup en el servidor