**RSYNC**

***Rsync*** es una aplicación para sistemas *GNU/Linux* que permite sincronizar carpetas de forma incremental y permite trabajar con datos comprimidos y cifrados, además, permite sincronizar carpetas a través de la red mediante ***openSSH***

NOTA No es necesario que exista el directorio destino, el propio comando ***rsync*** lo crea automáticamente siempre y cuando tenga permisos para ello

NOTA Tener especial cuidado con la barra al final del nombre de los directorios

*rsync -a dirA/ dirB/* | *rsync -a dirA/ dirB* sincroniza los dos directorios para que sean exactamente iguales

*rsync -a dirA dirB/* | *rsync dirA dirB* copiaremos el directorio dirA dentro del directorio dirB

La sintaxis del comando ***rsync*** es *rsync [opciones] directorio\_origen [directorio\_origen2] … directorio\_destino*

|  |  |  |
| --- | --- | --- |
| **OPCIONES GENERALES COMANDO RSYNC** | | |
| **OPCIÓN** | | **DESCRIPCIÓN** |
| -a | --archive | Realiza una copia exacta de una jerarquía de archivos y directorios recursivamente |
| -v | | Muestra por pantalla información referente a la sincronización |
| -W | --whole-file | Sincroniza archivos y directorios modificados sin aplicar el algoritmo ***rsync***  Opción por defecto |
| --no-whole-file | | Sincroniza archivos y directorios modificados aplicando el algoritmo ***rsync*** |
| -z | | Comprime la transferencia de archivos y directorios |
| -n | --dry-run | Realiza una prueba de sincronización sin llevarla a cabo |
| --delete | | Elimina archivos y/o directorios en el directorio destino que no se encuentran en el directorio origen |
| --prune-empty-dirs | | La sincronización no tiene en cuenta los directorios vacíos |
| -u | | No se sobre-escriben archivos y/o directorios modificados en el directorio destino  *NOTA Para realizar sincronizaciones bidireccionales ver* ***UNISON*** |

**Sincronizando directorios de forma local**

Ejemplo. Sincronizar/copiar el contenido del directorio dirA al directorio dirB

*rsync -av dirA/ dirB*

A partir de ahora, si modificamos algún archivo y volvemos a ejecutar el comando anterior, sólo se copiará el archivo modificado y no todo el directorio

Ejemplo. Sincronizar los directorios dirA y dirB utilizando el algoritmo ***rsync*** y comprimiendo la transferencia

*rsync -av -z --no-whole-file dirA/ dirB*

Ejemplo. Eliminar en el directorio destino dirB los archivos eliminados en el directorio origen dirA

*rsync -av --delete dirA/ dirB*

Ejemplo. Mantener en el directorio destino los archivos modificados en dicho directorio

*rsync -av -u dirA/ dirB*

**Backups incrementales**

Los backups incrementales permiten guardar versiones previas de los archivos modificados

*-b* | *--backup* Indica a ***rsync*** que haga una copia incremental

*--backup-dir=ruta* Indica donde se almacenarán los archivos originales antes de modificarlos en el directorio destino. Se tiene que indicar la ruta absoluta

*-vv* Muestra información sobre la sincronización de directorios y sobre las copias incrementales

Ejemplo. Realizar un backup incremental del directorio dirA en el directorio dirB guardando las copias originales en un directorio con el año mes día y hora

*rsync -avvb --backup-dir=$PWD/copias/$(date +%y%m%d%H) dirA/ dirB*

**Exclusión archivos/directorios**

*--exclude=PATTERN* Excluir archivos/directorios que coincidan con la expresión PATTERN

*--exclude-from=FILE* Excluir archivos/directorios que coincidan con una lista de PATTERNS desde un fichero FILE

*--include=PATTERN*  Incluir archivos/directorios que coincidan con la expresión PATTERN

*--include-from=FILE* Incluir archivos/directorios que coincidan con una lista de PATTERNS desde un fichero FILE

*--files-from=FILE* Incluir archivos/directorios especificados en el archivo FILE

|  |  |
| --- | --- |
| **EXPRESIONES REGULAES RSYNC** | |
| **EXPRESIÓN** | **DESCRIPCIÓN** |
| + | Incluye el archivo o directorio indicado |
| - | Excluye el archivo o directorio indicado |
| \* | Indica todos los archivos contenidos dentro de un directorio o todos los elementos si no se indica un directorio |
| \*\* | Indica todos los archivos y directorios dentro de un directorio |
| \*\*\* | Indica todos los archivos y todos los directorios incluyendo el propio directorio padre |

NOTA En todos los casos se debe indicar el directorio origen, el cual se tomará como directorio base, y todas las búsquedas serán relativas a dicha ubicación

Ejemplo. Sincronizar los archivos archA1.txt y el archA3.txt del directorio dirA y un archivo del subdirectorio dirAA en el directorio dirB

*rsync -av --files-from=listas/lista.txt dirA/ dirB*

Y el archivo listas/listas.txt

*#LISTAFICHEROS A INCLUIR*

*archA1.txt*

*archA3.txt*

*dirAA*

*dirAA/archAA1.txt*

Ejemplo. Sincronizar el directorio dirA con el directorio dirB excluyendo los archivos .doc

*rsync -av --exclude=”\*.doc” dirA/ dirB*

Ejemplo. Sincronizar el directorio dirA con el directorio dirB excluyendo los archivos .doc y los archivos y/o directorios que tengan 1 en su nombre

*rsync -av --exclude=”\*.doc” --exclude=”\*1\*” dirA/ dirB*

Ejemplo. Sincronizar solo los archivos .txt entre el directorio dirA y el directorio dirB

*rsync -av --include=”\*.txt” --exclude=”\*” dirA/ dirB*

Ejemplo. Sincronizar el directorio */var/log* y el directorio */var/games* con el directo dirB

*rsync -av --include-from=listas/listas.txt / dirB*

Y el archivo listas/listas.txt

*#LISTA DIRECTORIOS A INCLUIR*

*+ /var/*

*+ /var/log/\*\*\**

*+/var/games/\*\*\**

*-\**

**Sincronización remota a través del protocolo *openSSH***

***Rsync*** está preparado para trabajar conjuntamente con el protocolo ***openSSH***, simplemente hay que indicar el usuario y el ordenador con el que nos conectaremos de forma *usuario@ordenador* en el directorio origen o en el directorio destino, no en ambos

Ejemplo. Sincronizar remotamente el directorio dirA con el directorio dirB que se encuentra en el ordenador 192.168.1.XX

*rsync -av dirA/* [*usuario@192.168.1.XX:dirB*](mailto:usuario@192.168.1.XX:dirB)

NOTA Si el servicio ***openSSL*** está configurado para escuchar peticiones desde un puerto distinto al puerto por defecto 22 se tendrá que especificar de la forma *-e “ssh -p número\_puerto”*

**UNISON**

***Unison*** es una herramienta de sincronización bidireccional, por tanto, los directorios son origen y destino a la vez. Permite sincronizaciones remotas mediante ***openSSH***

Instalación ***unison*** y su interfaz gráfica desde los repositorios

*sudo apt-get install unison* | *sudo apt-get install unison-gtk*

La sintaxis para sincronizaciones locales es *unison [opciones] directorioA directorioB*

La sintaxis para sincronizaciones remotas es *unison [opciones] directorioA ssh://usuario@ordenador\_remoto/directorio2*

Ejemplo. Sincronizar los contenidos del directorio dirA y el directorio dirB

*unison dirA dirB*

Si el contenido de las dos carpetas difiere, ***unison*** dará a conocer la dirección de sincronización y si el fichero es nuevo o ha sido modificado

Para aceptar la sincronización tendremos que pulsar ***f*** o ***/*** para no aceptarla, en caso de conflicto, un archivo modificado en ambos directorios, se nos dará a elegir la dirección de sincronización mediante las teclas ***>*** y ***<***

Si quieres ver los ficheros log *unison.log* con las sincronizaciones realizadas, estos se encuentran en el directorio *.unison* dentro del *HOME* del usuario que ha realizado la sincronización

**AUTOMATIZAR PROCESOS CON CRON**

Para programar tareas o trabajos recurrentes, como puede ser crear copias de seguridad, todas las distribuciones *GNU/Linux* disponen de un programa ***cron*** que se encarga comprobar las posibles tareas programadas en el sistema y ejecutarlas

Instalación ***cron***y su interfaz gráfica desde los repositorios

*sudo apt-get install cron* | *sudo apt-get install gnome-schedule*

El comando que permite crear tareas programadas o *cronjobs* es ***crontab*** y su sintaxis es *crontab [opciones]*

|  |  |
| --- | --- |
| **OPCIONES BÁSICAS CRONTAB** | |
| **OPCIÓN** | **DESCRIPCIÓN** |
| -e | Edita el *crontab* del usuario |
| -l | Lista los *crontab* del usuario |
| -r | Elimina los *crontab* del usuario |
| -u | Permite al administrador del sistema *root* editar los *crontab* del usuario especificado  Puede utilizarse con cualquier opción anterior |

**Creando *crontabs***

Cada línea de ***crontab*** contiene 6 campos con la siguiente información y en este orden

Minuto Valores desde 0 a 59, así como el comodín \*

Hora Valores entre 0 y 23 o \*

Día del mes De 1 a 31 o \*

Mes De 1 a 12 o \*

Día de la semana De 0 a 6, donde 0 indica domingo, o \*

Comando Indica el comando junto con sus opciones a ejecutar. Alternativamente puede ser el nombre de un script con más comandos

Ejemplo. Ejecutar el comando “*comando*” a las 10:30 todos los días

*30 10 \* \* \* comando*

En los campos individuales los valores pueden estar separados por una coma

Ejemplo. Ejecutar el comando “*comando*” a las 10:30 de lunes a viernes

*30 10 \* \* 1,2,3,4,5 comando*

También es posible indicar un rango usando guiones

Ejemplo. Ejecutar el comando “*comando*” a las 10:30 de lunes a viernes

*30 10 \* \* 1-5 comando*

Ejemplo. Ejecutar el comando “*comando*” a las 10:30 de lunes a miércoles en enero, febrero, marzo, abril, julio, octubre, noviembre y diciembre

*30 10 \* 1-4,7,10-12 1-3 comando*

Una barra seguida de un número también define periodos de tiempo regulares

Ejemplo. Ejecutar el comando “*comando*” cada dos horas

*\* \*/2 \* \* \* comando*

Las tablas de ***cron*** para los usuarios se guardan en */var/spool/cron/crontabs* y se clasifican por el nombre de usuario

También existe una tabla global en el directorio */etc/crontab* con los trabajos que maneja ***cron***. A diferencia de las tablas *crontab* normales la *crontab* global tiene un séptimo campo que indica el usuario que ejecutará el comando

**Autorizar o denegar el uso de *crontab***

NOTA En *Debian/Ubuntu* no existen los archivos *cron.deny* y *cron.allow*, habrá que crearlos según nuestras necesidades

Se puede autorizar o denegar el uso del comando ***crontab*** mediante los archivos *cron.allow* y *cron.deny* dentro del directorio */etc* con los usuarios que podrán crear o no respectivamente *cronjobs*

NOTA Se tiene que tener en cuenta que la existencia o no de estos ficheros afectan directamente a las restricciones, así, si el archivo *cron.allow* existe la restricción es total a todos los usuarios menos a los listados dentro de él, si existe pero está vacío nadie podrá crear *cronjobs*

**Ejercicios**

1. Crea una copia de seguridad del directorio */home/usuario/Dodumentos* los viernes a las 18:00 en el directorio */etc/copias\_sec*
2. Crea una copia de seguridad del directorio */home/usuario\_compañero* en el directorio */home/usuario/copia* comprimiendo la transferencia y excluyendo los archivos y directorios ocultos
3. Crea una copia de seguridad del directorio */var/log* en el directorio */home/usuario\_compañero/log\_remoto* sin necesidad de introducir contraseña en la transferencia
4. Crea una copia de seguridad incremental del directorio */home/usuario\_compañero/Escritorio* en el directorio */home/usuario/copia\_escritorio*
5. Crea una copia de seguridad de los directorios */var/log/news* y del directorio y del directorio */etc/samba* en el directorio */home/usuario/ejercicio* cada 4 meses los lunes y jueves a las 17:00