

# Seguridad y alta disponibilidad ASIR2

## Tema 2. Almacenamiento y recuperación de datos

Copias de seguridad en Linux:

# Rsync

Práctica		Tema:	2	Número:	2
Grupo:	Componentes:				
ASIR2__					
Fecha de comienzo			Fecha de entrega		

**ÍNDICE****Contenido**

ÍNDICE .....	2
TÍTULO.....	3
OBJETIVOS .....	3
ENUNCIADO .....	3
DESARROLLO .....	3
Manejo básico Rsync .....	3
Copia remota.....	4
Automatización de la copia.....	6
Servidor SAMBA .....	7
HERRAMIENTAS .....	7
COMENTARIOS TÉCNICOS / DIFICULTADES ENCONTRADAS .....	7
CONCLUSIÓN.....	8
RECURSOS .....	8

**TÍTULO**

Backup Linux: rsync.

**OBJETIVOS**

- Realizar copias de seguridad con el comando de Linux rsync.

**ENUNCIADO**

Rsync o Remote Sync (Sincronización Remota) es una herramienta que permite copiar y sincronizar datos de forma local y remota. Se suele usar para realizar copias de seguridad o migraciones de datos rápidas entre dos máquinas Linux.

A lo largo de la práctica vamos a trabajar los comandos más sencillos para familiarizarnos con el comando.

Veremos también como podemos hacer copias y guardarlas en un equipo remoto y, finalmente, como automatizar las copias.

Aquí tenéis varios ejemplos de su funcionamiento:

<https://www.librebyte.net/comandos/rsync-ejemplos-practicos/>

<https://www.tecmint.com/rsync-local-remote-file-synchronization-commands/>

<https://gigastur.es/copias-seguridad-linux-rsync>

Copias totales, incrementales, diferenciales

<https://www.linuxadictos.com/rsync-como-crear-copia-seguridad-incremental.html>

**DESARROLLO**

Realiza las siguientes acciones (**pega pantallazos de los pasos que das** y contesta a las preguntas).

**Manejo básico Rsync**

1. En Kali crea un directorio llamado `original_tunombre`.  
Genera 10 archivos que empiecen con tu nombre y guárdalos dentro del directorio recién creado:  

```
touch original_tunombre/tunombre{1..10}
```

  
Crea otros dos directorios, llamados `copia_tunombre` y `copia2_tunombre`.
2. La sintaxis general del comando rsync es:  

```
rsync [opciones] origen destino
```

  
*Ejecuta rsync -av original\_tunombre/ copia\_tunombre*  
*Ejecuta rsync -av original\_tunombre copia2\_tunombre*  
¿Qué diferencia hay entre las dos copias?  
Que en la primera copia no se ha creado el directorio `original_miguel` dentro del directorio de copia. **(0.1 p)**  
  
¿Qué hace el argumento `-a` (explicándolo bien)? ¿y el argumento `v`? **(0.15 p)**  
*Copia los archivos*
3. Crea 5 archivos nuevos llamados `hola1..hola5` en la carpeta `original_tunombre` y vuelve a ejecutar el segundo comando. Observa la información ofrecida al ejecutar el comando y el contenido de la carpeta destino.

Modifica uno de los archivos de la carpeta original\_tunombre y crea otros 5 archivos nuevos en dicha carpeta. Vuelve a ejecutar el segundo comando.

Observa la información ofrecida al ejecutar el comando y el contenido de la carpeta destino.

Justifica qué tipo de copia hace por defecto rsync.

Ha realizado una copia incremental ya que muchos de los archivos copiados ya existían y no han sido modificados **(0.1 p)**

4. Borra uno de los ficheros de la carpeta original\_tunombre y ejecuta el segundo comando. El archivo eliminado en la carpeta origen no se borra en la de destino.

Modifica la orden para que, si se borra un archivo del origen del que ya hemos hecho copia, al ejecutar el comando se borre también en el destino. **(0.15 p)**

Tendrías que agregar la opción `--delete`

### Copia remota

5. En el servidor crea un directorio llamado copiasseguridad\_tunombre.

Guarda una copia de original\_tunombre (Kali), comprimiéndola en el propio comando rsync, en el directorio recién creado en el servidor Ubuntu.

Para ello tendrás que tener ssh activo (rsync usa ssh por debajo).

No te conectes al servidor con el usuario root (ssh no permite conectarse con root, a no ser que lo modifiquemos en el archivo de configuración, y no es una buena práctica de seguridad). **(0.2 p)**

```
(miguel@kalimiguel)-[~]
$ rsync -avz original_miguel/ miguel@172.16.0.254:copiasseguridad_miguel
miguel@172.16.0.254's password:
sending incremental file list
rsync: [generator] chgrp "/home/miguel/copiasseguridad_miguel/." failed: Operation not permitted (1)
./
hola1
hola2
hola3
hola4
hola5
miguel1
miguel10
miguel3
miguel4
miguel5
miguel6
miguel7
miguel8
miguel9

sent 853 bytes  received 285 bytes  455,20 bytes/sec
```

6. Sin mirar en el servidor el contenido de lo copiado desde Kali, comprueba que lo que se ha copiado es, efectivamente, exactamente lo que has mandado copiar. **(0.1 p)**

7. En el servidor, crea un script llamado copia\_tunombre.sh.

El script hará lo siguiente:

- Dentro del directorio destino, cada vez que se ejecute el script, se creará un directorio llamado copia\_tunombre\_fecha (donde fecha será la fecha del día en que se ejecuta el script y se hace la copia) en el que se guardará la copia. **(0,25 p)**

```
root@serverquique:/home/quique# ls destino/  
copia_202011022015 copia_202011022016
```

- Sólo copiará, de forma comprimida, los archivos que empiecen por tu nombre (los "hola" no). **(0,1 p)**
- Se tendrá que mostrar el progreso de la transferencia. **(0,1 p)**
- Se creará un log llamado log\_copia\_tunombre en el que quedará registro de los archivos copiados y la fecha en la que se ejecutó el script. El formato será el siguiente: **(0,25 p)**

```
==== COPIA 28/10/21 – 15:09 =====  
Quique1.txt  
Quique2.txt  
...  
  
==== COPIA 29/10/21 – 10:25 =====  
NombreArchivo.txt
```

Verifica que funciona el script.

```
#!/bin/bash  
FECHA=$(date +%d%m%Y)  
mkdir copia_miguel_$FECHA  
echo "==== COPIA $FECHA ====" > log_copia_miguel.txt  
rsync -avz miguel@192.168.1.6:original_miguel/miguel* copia_miguel_$FECHA >> log_copia_miguel.txt
```

```
miguel@servermiguel:~$ . copia_miguel.sh
mkdir: cannot create directory 'copia_miguel_19102022': File exists
miguel@192.168.1.6's password:
Permission denied, please try again.
miguel@192.168.1.6's password:
receiving incremental file list
miguel1
miguel10
miguel13
miguel14
miguel15
miguel16
miguel17
miguel18
miguel19

sent 195 bytes  received 539 bytes  54.37 bytes/sec
total size is 5  speedup is 0.01
miguel@servermiguel:~$
```

### Automatización de la copia

8. Explica para que sirven y las diferencias entre el cron y crontab. (0.1 p)  
<https://blog.desdelinux.net/cron-crontab-explicados/>  
*Cron es un administrador de procesos en segundo plano y crontab es un fichero con comandos para ejecutar en background*
9. En el servidor, usando el script creado en el punto 6 y el cron, haz que se haga una copia todos los sábados a las once de la noche y otra que se ejecute 2 minutos después de cuando termines de configurar el cron.

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
00 23 * * 6 /home/miguel/copia_seguridad.sh
45 12 19 10 * /home/miguel/copiaseguridad.sh
```

PISTA: Puede ser que el servidor no tenga nuestra “misma hora”.  
 Verifica si se ejecuta la tarea automática.

(0.25 p)

## 10. Observa el log a ver si ha funcionado el script.

Antes, al lanzar el script manualmente ha funcionado y, ahora, al automatizarlo no ha funcionado.

¿Se te ocurre por qué puede ser?

Por no darle al script permiso de ejecución

**(0,15 p)**

### Servidor SAMBA

<https://noviello.it/es/como-instalar-y-configurar-el-servidor-samba-en-ubuntu-20-04-lts/>

A través del protocolo smb, podremos compartir recursos en una red Linux o entre equipos Windows y Linux.

Vamos a hacer que el directorio del servidor donde guardábamos las copias sea un recurso compartido con otros equipos de la red.

a) Instala samba.

b) Abre el archivo de configuración (/etc/samba/smb.conf)

Al final del todo, añade lo necesario para compartir el directorio del servidor.

Se podrá acceder a él sin autenticación y copiar y borrar archivos.

```
: write list = root, @lpadmin
[private]
comment = needs username and password to access
path = /srv/private/browseable = yes
guest ok = no
writable = yes
valid users = @samba
```

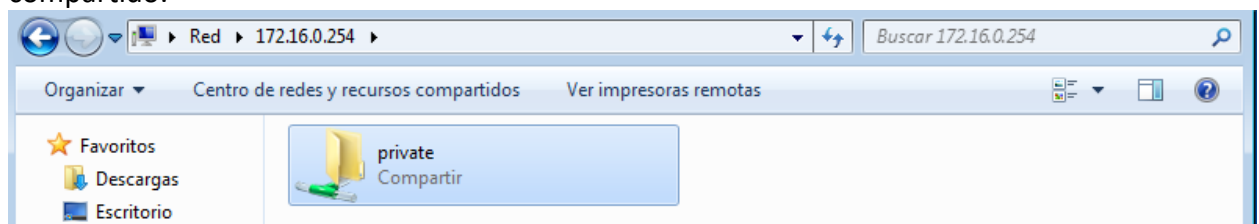
Reinicia samba:

c) Tendremos que tener permiso sobre el directorio para poder escribir en él.

(sudo setfacl -R -m "u:nobody:rw" /prueba\_samba)

Como todavía no hemos visto acls (el comando anterior), lo haremos a lo bruto (y mal).

d) Prueba desde el entorno de red de Windows a copiar un archivo en el directorio compartido.



## HERRAMIENTAS

**(-0.1 p)**

## COMENTARIOS TÉCNICOS / DIFICULTADES ENCONTRADAS

**(-0.1 p)**

**CONCLUSIÓN***(-0.2 p)***RECURSOS***(-0.1 p)*