

# Práctica Seguridad Informática: Backup

## ESCENARIO

### Máquinas virtuais ou físicas:

RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado

Rede: 192.168.120.0/24

BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

### Máquina A:

Rede Interna(eth0) e NAT(eth1)

IP/MS: 192.168.120.100/24

Servidor SSH

SO: Debian amd64 xfce instalado

sda: SO instalado

User: usuario

Passwd: abc123.

Groups usuario: usuario sudo

usuario → sudo su - → root

tar || rsync || rdiff-backup || cron

### Máquina B:

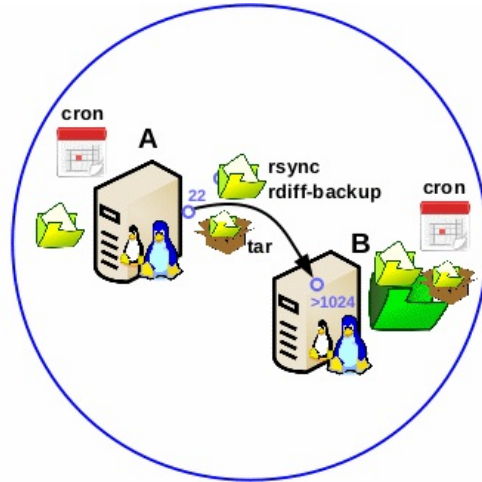
Rede Interna(eth0)

IP/MS: 192.168.120.101/24

Cliente SSH

SO: Live GNU/Linux

tar || rsync || rdiff-backup || cron



**LIMITACIÓN DE RESPONSABILIDADE** O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

## NOTA:

### ■ Documentación de interese:

- **Respaldo (debian-handbook)**
- **Práctica SI RAID**
- **Amanda**
- **Bacula**
- **Comandos GNU/Linux e SHELL BASH (/bin/bash) (1)**
- **Cifrado asimétrico: Conexión Remota mediante SSH sen contrasinal**
- Paquete tar (# apt update && apt -y install tar).  
\$ man tar  
\$ tar cvfj ruta\_destino/file.tar.bz2 ruta\_orixe  
\$ ssh user@host tar cvfj - ruta\_orixe > ruta\_destino/file.tar.bz2

- tar cvf : Empaquetar
- tar tvf: Ver contido do empaquetado
- tar xvf: Desempaquetar
- tar cvfz : Comprimir en tar.gz
- tar tvfz: Ver contido do comprimido en tar.gz
- tar xvfz: Descomprimir en tar.gz
- tar cvfj : Comprimir en tar.bz2
- tar tvfj: Ver contido do comprimido en tar.bz2
- tar xvfj: Descomprimir en tar.bz2

- o Paquete rsync (# apt update && apt -y install rsync).

\$ man rsync

\$ rsync -avz --progress ruta\_orixe ruta\_destino

\$ rsync -avz --progress --rsh=ssh user@host:ruta\_orixe ruta\_destino

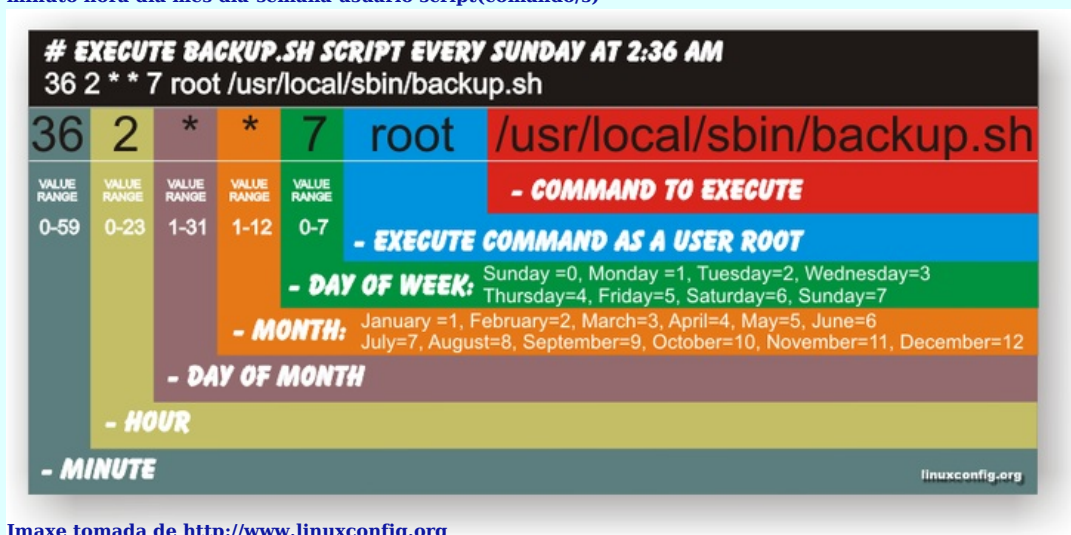
Copiar a estrutura arbórea do contido do cartafol ruta\_orixe dentro do cartafol ruta\_destino, coas seguintes opcións:

- -a → modo arquivo. Equivale ás opcións -rlptgoD, onde:
  - -r → modo recursivo
  - -l → copiar ligazóns simbólicas como ligazóns simbólicas
  - -p → preservar permisos
  - -t → preservar mtime (tempos de modificación)
  - -g → preservar grupo propietario
  - -o → preservar usuario propietario
  - -D → equivale a --devices --specials, onde
    - --devices → preservar arquivos de dispositivos
    - --specials → preservar ficheiros especiais
- -v → modo detallado
- -z → modo comprimir
- --progress → amosar o progreso da copia durante a transferencia

- o cron: /etc/crontab (global) , /var/spool/cron/crontabs/username

### /etc/crontab (global)

minuto hora dia mes dia-semana usuario script(comando/s)



Imaxe tomada de <http://www.linuxconfig.org>

- o Paquete **rdiff-backup** (# apt update && apt -y install rdiff-backup).

\$ man rdiff-backup

\$ rdiff-backup ruta\_orixe ruta\_destino #Xerar en ruta\_destino unha copia diferencial de ruta\_orixe. Se ruta\_destino non existe créase. Dentro de ruta\_destino crearase o directorio rdiff-backup-data o cal permitirá restaurar estados anteriores mediante rdiff-backup

\$ rdiff-backup user@host::ruta\_orixe ruta\_destino #Xerar na ruta local ruta\_destino unha copia diferencial de ruta\_orixe dun host remoto . Se ruta\_destino non existe créase. Dentro de ruta\_destino crearase o directorio rdiff-backup-data o cal permitirá restaurar estados anteriores mediante rdiff-backup. A conexión remota con ruta\_orixe establécese mediante unha conexión SSH (cifrada).

\$ rdiff-backup -l ruta\_destino #Amosar listado con todas as copias xeradas en ruta\_destino. Este comando ademais de ser útil para saber que copias temos realizadas danos as datas desas copias, polo que poderemos empregar esas datas no comando de restauración para volcar a copia en data que nos interese.

\$ rdiff-backup -r data ruta\_destino ruta\_orixe#Restaurar a copia según data que nos interese dentro de ruta\_destino en ruta\_orixe. Para indicar a data podemos empregar o comando anterior de rdiff-backup coa opción -l

#### doc-examples rdiff-backup

rdiff-backup-data

→ directorio que almacena a información necesaria para o correcto funcionamento de rdiff-backup.

rdiff-backup-data/increments

→ directorio que almacena as copias das versions anteriores á última copia realizada.

Tipos de Backup	Descripción
Completo	<div data-bbox="225 152 1549 495"></div>
	<p><b>Restauración:</b> Para restaurar un respaldo completo, simplemente teremos que volcar a última copia completa.</p> <p><b>Restauración ⇒ Copia día7</b></p>
Incremental	<div data-bbox="225 786 1549 1128"></div>
	<p><b>Restauración:</b> Para restaurar un respaldo incremental, temos que restaurar a última copia completa e logo ir volcando todas as copias incrementais dende esa copia completa ata chegar á última copia incremental.</p> <p><b>Restauración ⇒ Datos + Copia día1 + Copia día2 + Copia día3 + Copia día4 + Copia día5 + Copia día6 + Copia día7</b></p>
Diferencial	<div data-bbox="225 1545 1549 1888"></div>
	<p><b>Restauración:</b> Para restaurar un respaldo diferencial, temos que restaurar a última copia completa e logo volcar a última copia diferencial correspondente con esa copia completa.</p> <p><b>Restauración ⇒ Datos + Copia día7</b></p>

## Resumo Prácticas Exemplos

### Restauración completa

- No **Exemplo1. tar - estrutura arbórea: copia local** imos crear unha copia de respaldo completa de todo o que colga dun directorio. A copia farase no propio host.
- No **Exemplo2. tar - estrutura arbórea: copia remota** imos crear unha copia de respaldo completa de todo o que colga dun directorio. A copia farase remotamente de forma cifrada mediante conexión SSH.
- No **Exemplo3. cron tar - estrutura arbórea: copia local todos os días** mediante unha tarefa programada imos crear unha copia de respaldo completa cada día de todo o que colga dun directorio. A copia farase no propio host.
- No **Exemplo4. tar - estrutura arbórea: recuperación dunha copia local** imos recuperar todo o que colga dun directorio mediante o volcado da copia de respaldo completa realizada na tarefa programada do Exemplo3. A recuperación farase de forma local dende o propio host.

### Restauración incremental

- No **Exemplo5. rsync - estrutura arbórea: copia local** imos crear unha copia de respaldo de todo o que colga dun directorio. A copia farase no propio host.
- No **Exemplo6. rsync - estrutura arbórea: copia remota** imos crear unha copia de respaldo de todo o que colga dun directorio. A copia farase remotamente de forma cifrada mediante conexión SSH.
- No **Exemplo7. cron rsync - estrutura arbórea: copia local todos os días** mediante unha tarefa programada imos crear unha copia de respaldo incremental cada día de todo o que colga dun directorio. A copia farase no propio host.
- No **Exemplo8. rsync - estrutura arbórea: recuperación dunha copia local** imos recuperar todo o que colga dun directorio mediante o volcado da copia de respaldo completa e as sucesivas copias de respaldo incremental realizada na tarefa programada do Exemplo7. A recuperación farase de forma local dende o propio host.

### Restauración diferencial

- No **Exemplo9. rdiff-backup - estrutura arbórea: copia local** imos crear unha copia de respaldo de todo o que colga dun directorio. A copia farase no propio host.
- No **Exemplo10. rdiff-backup - estrutura arbórea: copia remota** imos crear unha copia de respaldo de todo o que colga dun directorio. A copia farase remotamente de forma cifrada mediante conexión SSH.
- No **Exemplo11. cron rdiff-backup - estrutura arbórea: copia local todos os días** mediante unha tarefa programada imos crear unha copia de respaldo diferencial cada día de todo o que colga dun directorio. A copia farase no propio host.
- No **Exemplo12. rdiff-backup - estrutura arbórea: recuperación dunha copia local** imos recuperar todo o que colga dun directorio mediante o volcado da copia de respaldo completa e a última copia diferencial realizada na tarefa programada do Exemplo11. A recuperación farase de forma local dende o propio host.



## Backup: Copias de respaldo

### Máquina virtual A: Debian amd64

1. Na contorna gráfica abrir un terminal e executar:

```
usuario@debian:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
```

```
usuario@debian:~$ passwd usuario #Cambiar o contrasinal do usuario usuario. Por como contrasinal abc123. (Olla que o contrasinal ten un caracter punto final).
```

2. Cambiar hostname da máquina virtual A. Por debianA como hostname:

```
usuario@debian:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debian:~# echo 'debianA' > /etc/hostname #Indicar ao sistema o valor do hostname.
```

```
root@debian:~# echo 'kernel.hostname=debianA' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
```

```
root@debian:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
```

```
root@debian:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.
```

```
usuario@debian:~$ exit #Pechar o terminal saíndo da consola local do usuario usuario.
```

3. Configurar a rede:

Na contorna gráfica abrir un terminal e executar:

```
usuario@debianA:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
```

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflito con este demo.
```

```
root@debianA:~# systemctl disable avahi-daemon #Impide que o servizo avahi-daemon sexa iniciado no arranque xerando os links K* nos runlevels (/etc/rcX.d)
```

```
root@debianA:~# /etc/init.d/network-manager stop #Parar o demo network-manager(xestor de rede) para poder configurar doutro xeito (co comando ip(ifconfig) de forma manual ou mediante networking (ficheiros /etc/init.d/networking, /etc/init.d/networking.d) a configuración de rede e non ter conflito con este xestor.
```

```
root@debianA:~# systemctl disable network-manager #Impide que o servizo network-manager sexa iniciado no arranque xerando os links K* nos runlevels (/etc/rcX.d)
```

```
root@debianA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), interna(enp0s3) e NAT(enp0s8).
```

```
$ man interfaces #Ver ás páxinas de manual referente ao ficheiro de configuración de rede /etc/network/interfaces
$ cat /etc/network/interfaces #Amosar o contido do ficheiro configuración de rede /etc/network/interfaces
$ ls -l /etc/network/interfaces.d #Listar de forma extendida o contido do directorio /etc/network/interfaces/setup
$ cat /etc/network/interfaces.d/setup #Amosar o contido do ficheiro configuración de rede /etc/network/interfaces/setup
```

```
root@debianA:~# cat > /etc/network/interfaces.d/setup <<EOF #Comezo do ficheiro a crear /etc/network/interfaces.d/setup
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto enp0s3
```

```
iface enp0s3 inet static
```

```
address 192.168.120.100/24
```

```
auto enp0s8
```

```
iface enp0s8 inet dhcp
```

```
EOF #Fin do ficheiro a crear /etc/network/interfaces.d/setup
```

```
root@debianA:~# /etc/init.d/networking status #Comprobar o estado do demo networking, é dicir, comprobar se está activa a configuración de rede en /etc/network/interfaces (/etc/network/interfaces.d).
```

```
root@debianA:~# /etc/init.d/networking start #Arrancar o demo networking, é dicir, activar a configuración de rede en /etc/network/interfaces (/etc/network/interfaces.d).
```

```
root@debianA:~# /etc/init.d/networking status #Comprobar o estado do demo networking, é dicir, comprobar se está activa a configuración de rede en /etc/network/interfaces (/etc/network/interfaces.d).
```

```
root@debianA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), interna(enp0s3) e NAT(enp0s8).
```

```
root@debianA:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local enp0s3
```

#### 4. Comprobar estado do Servidor SSH:

```
# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
# apt -y install netcat #Instalar o paquete netcat, é dicir, instalar o paquete que integra o comando nc. Co
parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
# dpkg -l net-tools ; [ $(echo $?) -eq '1' ] && apt update && apt -y install net-tools #Verificar se o paquete
net-tools está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase. O
paquete net-tools é necesario para poder empregar comandos coma: ifconfig, netstat, route e arp.
# dpkg -l openssh-server ; [ $(echo $?) -eq '1' ] && apt update && apt -y install openssh-server #Verificar se
o paquete openssh-server está instalado. Se non está instalado, actualízase a lista de paquetes dos
repositorios e instálase.
```

root@debianA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, por defecto non está arrancado.

root@debianA:~# nc -vz localhost 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.

root@debianA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.

root@debianA:~# netstat -natp | grep 22 #Mediante o comando netstat comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.

root@debianA:~# ss -natp | grep 22 #Mediante o comando ss comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.

root@debianA:~# /etc/init.d/ssh start #Arrancar o servidor SSH.

root@debianA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, agora debe estar arrancado.

root@debianA:~# find /etc/rc\* -name "\*ssh\*" #Busca polas links runlevels nos cartafoles /etc/rc\*

root@debianA:~# systemctl enable ssh #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos runlevels (/etc/rcX.d)

root@debianA:~# find /etc/rc\* -name "\*ssh\*" #Busca polas links runlevels nos cartafoles /etc/rc\*

root@debianA:~# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled

root@debianA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.

root@debianA:~# ssh -v usuario@localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende localhost co usuario **usuario** e o seu contrasinal. Se é a primeira vez que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter. A opción -v (modo verbose) aporta información máis detallada da conexión.

usuario@debianA:~\$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

usuario@debianA:~\$

## Máquina virtual B: Kali amd64

5. Configuración da rede. Na contorna gráfica abrir un terminal e executar:

```
kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
```

```
kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@kali:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflito con este demo.
```

```
root@kali:~# /etc/init.d/network-manager stop #Parar o demo network-manager(xestor de rede) para poder configurar de forma manual a configuración de rede e non ter conflito con este xestor.
```

```
root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
```

```
root@kali:~# ip addr add 192.168.120.101/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.101 e máscara de subrede: 255.255.255.0.
```

```
root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
```

```
root@kali:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
```

```
root@kali:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
```

```
root@kali:~# echo '192.168.120.100 debianA' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome debianA, para que atenda á IP 192.168.120.100
```

```
root@kali:~# ping -c4 debianA #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
```

6. Cambiar hostname da máquina virtual B. Por kaliB como hostname:

```
root@kali:~# echo 'kaliB' > /etc/hostname #Indicar ao sistema o valor do hostname.
```

```
root@kali:~# echo 'kernel.hostname=kaliB' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
```

```
root@kali:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
```

```
root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de kali.
```

```
kali@kali:~$ exit #Pegar o terminal saíndo da consola local do usuario kali.
```

7. <sup>SSH</sup> **B → A** Acceder mediante SSH dende a máquina virtual B á máquina virtual A. Dende agora executaremos sempre os comandos dende a máquina virtual B, a través da consola SSH:

Na contorna gráfica abrir un terminal e executar:

```
kali@kaliB:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
```

```
kali@kaliB:~$ nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
```

```
kali@kaliB:~$ nc -vz debianA 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
```

```
kali@kaliB:~$ ssh -v usuario@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el. Agora accedemos como o usuario usuario a través da conexión cifrada SSH.
```

```
usuario@debianA:~$
```

## 8. Exemplo1. tar - estrutura arbórea: copia local

Imos crear unha copia de respaldo completa de todo o que colga do directorio /home. A copia farase no propio host.

```
usuario@debianA:~$ ls -l /home #Listar de forma extendida o contido do directorio /home
```

```
usuario@debianA:~$ du -hs /home #Amosar en formato humano canto ocupa o cartafol /home
```

```
usuario@debianA:~$ ls -ld /var/backups/ #Listar soamente os permisos do cartafol /var/backups, é dicir, listar os permisos do propio cartafol pero non os do seu contido.
```

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# tar cvfj /var/backups/backupHOME.tar.bz2 /home #Crear un ficheiro comprimido mediante bzip2 do que foi empaketado mediante tar, o cal contén o contido do directorio /home eliminando as barras iniciais '/' dos nomes, é dicir, empaqueta home e non /home. Independientemente do número de veces que se execute este comando sempre farase unha copia de respaldo completa
```

```
root@debianA:~# tar cvfj /var/backups/backupHOME2.tar.bz2 /home/ #Equivalente ao comando tar executado anteriormente, cambiando soamente o nome do ficheiro destino da copia.
```

**IMPORTANTE:** Notar que non existe diferenca se empregamos como *ruta\_orixe* o cartafol **/home** ou **/home/**

```
root@debianA:~# md5sum /var/backups/backupHOME*.tar.bz2 #Crear hash MD5 dos ficheiros backupHOME.tar.bz2 e backupHOME2.tar.bz2. Podemos comprobar que os hash son iguais.
```

```
root@debianA:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/
```

```
root@debianA:~# du -hs /var/backups #Amosar en formato humano canto ocupa o cartafol /var/backups
```

```
root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.
```

```
usuario@debianA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de kali en kaliB.
```

```
kali@kaliB:~$
```



## 9. Exemplo2. tar - estrutura arbórea: copia remota

Imos crear unha copia de respaldo completa de todo o que colga do directorio /home. A copia farase remotamente de forma cifrada mediante conexión SSH.

```
kali@kaliB:~$ ssh usuario@debianA ls -l /home #Listar de forma extendida o contido do directorio /home do host
debianA. O comando é executado en debianA accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é
amosada en kaliB
```

```
kali@kaliB:~$ ssh usuario@debianA du -hs /home #Amosar en formato humano canto ocupa o cartafol /home do
host debianA. O comando é executado en debianA accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa
saída é amosada en kaliB
```

```
kali@kaliB:~$ ssh usuario@debianA ls -ld /var/backups/ #Listar soamente os permisos do cartafol /var/backups
do host debianA, é dicir, listar os permisos do propio cartafol pero non os do seu contido. O comando é executado en debianA
accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é amosada en kaliB
```

```
kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo
(/etc/sudoers, visudo)
```

```
root@kaliB:~# ssh usuario@debianA tar cvfj - /home > /var/backups/backupHOME.tar.bz2
#Crear un ficheiro comprimido mediante bzip2 en kaliB, que foi empaquetado mediante tar, o cal contén o contido do
directorio /home de debianA eliminando as barras iniciais '/' dos nomes, é dicir, empaqueta home e non /home. O comando
é executado en debianA accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é amosada en
kaliB. Independentemente do número de veces que se execute este comando sempre farase unha copia de
respaldo completa
```

```
root@kaliB:~# ssh usuario@debianA tar cvfj - /home/ > /var/backups/backupHOME2.tar.bz2
#Equivalente ao comando tar executado anteriormente, cambiando soamente o nome do ficheiro destino da copia.
```

**IMPORTANTE:** Notar que non existe diferenza se empregamos como *ruta\_orixe* o cartafol **/home** ou **/home/**

```
root@kaliB:~# md5sum /var/backups/backupHOME*.tar.bz2 #Crear hash MD5 dos ficheiros
backupHOME.tar.bz2 e backupHOME2.tar.bz2
```

```
root@kaliB:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/ do host kaliB
```

```
root@kaliB:~# du -hs /var/backups #Amosar en formato humano canto ocupa o cartafol /var/backups do host
kaliB
```

```
root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de kali.
```

```
kali@kaliB:~$
```

## 10. Exemplo3. cron tar - estrutura arbórea: copia local todos os días

Mediante unha tarefa programada imos crear unha copia de respaldo completa cada día de todo o que colga do directorio /home. A copia farase no propio host.

usuario@debianA:~\$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@debianA:~# echo '0 0 \* \* \* root tar cvfj /var/backups/backupHOME.tar.bz2 /home' >> /etc/crontab #Crear un ficheiro comprimido mediante bzip2, que foi empacutado mediante tar, o cal contén o contido do directorio /home eliminando as barras iniciais '/' dos nomes, é dicir, empacuta home e non /home. **Independentemente do número de veces que se execute este comando sempre farase unha copia de respaldo completa.** Esta tarefa programada execútase como o usuario root tódolos días ás 00:00. **Equivale a @daily ou @midnight**

```
root@debianA:~# echo '@daily root tar cvfj /var/backups/backupHOME.tar.bz2 /home' >> /etc/crontab
#Equivale ao comando anterior.
```

```
root@debianA:~# echo '@midnight root tar cvfj /var/backups/backupHOME.tar.bz2 /home' >> /etc/crontab
#Equivale ao comando anterior.
```

root@debianA:~# grep crontab /var/log/syslog Buscar o patrón crontab dentro do ficheiro de rexistros /var/log/syslog. Podemos observar que existe 1 nova entrada, a referente á recarga do ficheiro /etc/crontab.

```
Jan 11 19:26:01 debianA cron[358]: (*system*) RELOAD (/etc/crontab)
```

**IMPORTANTE:** Co realizado en /etc/crontab cada día sobreescríbese o ficheiro /var/backups/backupHOME.tar.bz2, polo que soamente poderíamos recuperar a copia de respaldo completa do último día. Pero que pasa se quixeramos recuperar de mais días atrás? Ou que pasa se a copia queda corrupta? Entón, estaría ben mellorar a xestión de copias de respaldo: creando máis tarefas programadas ou modificando esta tarefa programada para que non execute soamente o comando tar, senón que execute un script o cal xa permite a xestión que necesitamos.

root@debianA:~# for i in \$(seq 1 7); do echo "0 0 \* \* \${i} root tar cvfj /var/backups/backupHOME-\${i}.tar.bz2 /home" >> /etc/crontab;done #Engadir 7 liñas ao ficheiro /etc/crontab, onde cada unha delas permite crear un ficheiro comprimido mediante bzip2, que foi empacutado mediante tar, o cal posúe contén o contido do directorio /home eliminando as barras iniciais '/' dos nomes, é dicir, empacuta home e non /home. Esta tarefa programada execútase como o usuario root tódolos días ás 00:00. Cada día de semana o ficheiro posúe un nome distinto, así:

Día 1(luns)	→ backupHOME-1.tar.bz2
Día 2(martes)	→ backupHOME-2.tar.bz2
Día 3(mércores)	→ backupHOME-3.tar.bz2
Día 4(xoves)	→ backupHOME-4.tar.bz2
Día 5(venres)	→ backupHOME-5.tar.bz2
Día 6(sábado)	→ backupHOME-6.tar.bz2
Día 7(domingo)	→ backupHOME-7.tar.bz2

**Agora farase unha copia de respaldo completa por cada día da semana, de tal xeito que poderemos recuperar ata os últimos 7 días.**

root@debianA:~# grep crontab /var/log/syslog Buscar o patrón crontab dentro do ficheiro de rexistros /var/log/syslog. Podemos observar que existe 1 nova entrada, a referente á recarga do ficheiro /etc/crontab.

```
Jan 11 19:52:01 debianA cron[358]: (*system*) RELOAD (/etc/crontab)
```

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~\$

## 11. Exemplo4. tar - estrutura arbórea: recuperación dunha copia local

Imos recuperar todo o que colga do directorio /home mediante o volcado da copia de respaldo completa realizada na tarefa programada do Exemplo3. A recuperación farase de forma local dende o propio host. Para isto imos provocar o desastre en /home, é dicir, imos eliminar /home en debianA para logo recuperalo mediante a restauración da copia de respaldo completa.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# rm -rf /home #Eliminar o directorio /home e todo o que colga(-r) deste sen pedir confirmación(-f).
```

```
root@debianA:~# ls -l /home #Listar de forma extendida o contido do directorio /home. Neste caso obteremos un erro xa que acabamos de eliminar ese directorio.
```

```
root@debianA:~# du -hs /home #Amosar en formato humano canto ocupa o cartafol /home. Neste caso obteremos un erro xa que acabamos de eliminar ese directorio.
```

```
root@debianA:~# tar xvfj /var/backups/backupHOME-$(date +%u).tar.bz2 -C / #Descomprimir o contido do ficheiro no directorio raíz (/), é dicir, restaurar a copia de respaldo completa do directorio /home. O ficheiro a descomprimir será o último realizado na tarefa programada. Para determinalo empregamos o comando date +%u o cal determina o número do día da semana. Así se hoxe é martes, será o día 2.
```

**Agora restauramos mediante a última copia de respaldo completa según o día no que estamos, é dicir, se estamos en luns sendo as 12:20h realizamos a copia completa do propio luns as 00:00h. Estamos a considerar que esta última copia non está corrupta, senón deberíamos intentar o respaldo con copias de respaldo completas de días anteriores.**

```
root@debianA:~# ls -l /home #Listar de forma extendida o contido do directorio /home
```

```
root@debianA:~# du -hs /home #Amosar en formato humano canto ocupa o cartafol /home
```

```
root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.
```

```
usuario@debianA:~$
```

## 12. Exemplo5. rsync - estrutura arbórea: copia local.

Imos crear unha copia de respaldo de todo o que colga do directorio /home. A copia farase no propio host.

```
usuario@debianA:~$ ls -l /home #Listar de forma extendida o contido do directorio /home
```

```
usuario@debianA:~$ du -hs /home #Amosar en formato humano canto ocupa o cartafol /home
```

```
usuario@debianA:~$ ls -ld /var/backups/ #Listar soamente os permisos do cartafol /var/backups, é dicir, listar os permisos do propio cartafol pero non os do seu contido.
```

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# dpkg -l rsync ; [ $(echo $? ) -eq '1' ] && apt update && apt -y install rsync
#Verificar se o paquete rsync está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase.
```

```
root@debianA:~# rsync -avz --progress /home/ /var/backups #Copiar de forma incremental a estrutura arbórea do contido do cartafol /home dentro do cartafol /var/backups, en modo arquivo (-a)(Ver NOTAS), de forma detallada (-v), comprimida (-z) e amosando o proceso de copia (--progress). Como é a primeira que se empregan esas rutas (orixe e destino) faise unha copia completa da ruta orixe na ruta destino, xa que na ruta destino non existe ningunha copia da ruta orixe.
```

```
root@debianA:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/
```

```
root@debianA:~# du -hs /var/backups #Amosar en formato humano canto ocupa o cartafol /var/backups
```

```
root@debianA:~# rsync -avz --progress /home/ /var/backups #Copiar de forma incremental a estrutura arbórea do contido do cartafol /home dentro do cartafol /var/backups, en modo arquivo (-a)(Ver NOTAS), de forma detallada (-v), comprimida (-z) e amosando o proceso de copia (--progress). Como é a segunda vez que se empregan esas rutas (orixe e destino) faise unha copia incremental da ruta orixe na ruta destino, xa que na ruta destino existe unha copia completa da ruta orixe. Neste caso como non existen cambios non se produce copia de arquivos/directorios.
```

```
root@debianA:~# rsync -avz --progress /home /var/backups #Copiar de forma incremental a estrutura arbórea /home dentro do cartafol /var/backups, en modo arquivo (-a)(Ver NOTAS), de forma detallada (-v), comprimida (-z) e amosando o proceso de copia (--progress). Como é a primeira que se emprega esa ruta orixe (/home != /home/) con esa ruta destino (/var/backups) faise unha copia completa da ruta orixe na ruta destino, xa que na ruta destino non existe ningunha copia da ruta orixe.
```

**IMPORTANTE:** Notar a diferencia co comando anterior. Agora estamos a pór como *ruta\_orixe* o cartafol /home e non /home/. Así, como *ruta\_orixe*:

- /home → indica o cartafol /home e todo o seu contido
- /home/ → indica todo o contido do cartafol /home, pero non o propio cartafol /home

```
root@debianA:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/
```

```
root@debianA:~# du -hs /var/backups #Amosar en formato humano canto ocupa o cartafol /var/backups
```

```
root@debianA:~# rm -rf /var/backups/home #Eliminar o directorio /var/backups/home e todo o que colga (-r) deste sen pedir confirmación (-f).
```

```
root@debianA:~# rm -rf /var/backups/usuario #Eliminar o directorio /var/backups/usuario e todo o que colga (-r) deste sen pedir confirmación (-f).
```

```
root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.
```

```
usuario@debianA:~$
```

### 13. Exemplo6. rsync - estrutura arbórea: copia remota.

Imos crear unha copia de respaldo de todo o que colga do directorio /home. A copia farase remotamente de forma cifrada mediante conexión SSH.

```
kali@kaliB:~$ ssh usuario@debianA ls -l /home #Listar de forma extendida o contido do directorio /home do host
debianA. O comando é executado en debianA accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é
amosada en kaliB
```

```
kali@kaliB:~$ ssh usuario@debianA du -hs /home #Amosar en formato humano canto ocupa o cartafol /home do
host debianA. O comando é executado en debianA accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa
saída é amosada en kaliB
```

```
kali@kaliB:~$ ssh usuario@debianA ls -ld /var/backups/ #Listar soamente os permisos do cartafol /var/backups
do host debianA, é dicir, listar os permisos do propio cartafol pero non os do seu contido. O comando é executado en debianA
accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é amosada en kaliB
```

```
kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) en kaliB a través dos permisos configurados co
comando sudo (/etc/sudoers, visudo)
```

```
root@kaliB:~# dpkg -l rsync ; [ $(echo $? ) -eq '1' ] && apt update && apt -y install rsync
#Verificar se o paquete rsync está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e
instálase.
```

```
root@kaliB:~# rsync -avz --progress --rsh="ssh" usuario@debianA:/home/ /var/backups #Copiar a
estructura arbórea do contido do cartafol /home de debianA dentro do cartafol /var/backups de kaliB, en modo arquivo (-a)
(Ver NOTAS), de forma detallada (-v), comprimida (-z) e amosando o proceso de copia (--progress). O comando é executado
en debianA accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é amosada en kaliB. Como é
a primeira que se empregan esas rutas (orixe e destino) faise unha copia completa da ruta orixe na ruta destino,
xa que na ruta destino non existe ningunha copia da ruta orixe.
```

```
root@kaliB:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/ do host kaliB
```

```
root@kaliB:~# du -hs /var/backups #Amosar en formato humano canto ocupa o cartafol /var/backups do host
kaliB
```

```
root@kaliB:~# rsync -avz --progress --rsh="ssh" usuario@debianA:/home/ /var/backups #Copiar a
estructura arbórea do contido do cartafol /home de debianA dentro do cartafol /var/backups de kaliB, en modo arquivo (-a)
(Ver NOTAS), de forma detallada (-v), comprimida (-z) e amosando o proceso de copia (--progress). O comando é executado
en debianA accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é amosada en kaliB. Como é
a segunda vez que se empregan esas rutas (orixe e destino) faise unha copia incremental da ruta orixe na ruta
destino, xa que na ruta destino existe unha copia completa da ruta orixe. Neste caso como non existen cambios
non se produce copia de arquivos/directorios.
```

```
root@kaliB:~# rsync -avz --progress --rsh="ssh" usuario@debianA:/home /var/backups #Copiar a
estructura arbórea /home de debianA dentro do cartafol /var/backups de kaliB, en modo arquivo (-a)(Ver NOTAS), de forma
detallada (-v), comprimida (-z) e amosando o proceso de copia (--progress). O comando é executado en debianA accedendo
co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é amosada en kaliB. Como é a primeira que se
emprega esa ruta orixe (/home != /home/) con esa ruta destino (/var/backups) faise unha copia completa da ruta
orixe na ruta destino, xa que na ruta destino non existe ningunha copia da ruta orixe.
```

**IMPORTANTE:** Notar a diferencia co comando anterior. Agora estamos a pór como *ruta\_orixe* o cartafol /home e non /home/. Así, como *ruta\_orixe*:

- /home → indica o cartafol /home e todo o seu contido
- /home/ → indica todo o contido do cartafol /home, pero non o propio cartafol /home

```
root@kaliB:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/ do host kaliB
```

```
root@kaliB:~# du -hs /var/backups #Amosar en formato humano canto ocupa o cartafol /var/backups do host
kaliB
```

```
root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de kali.
```

```
kali@kaliB:~$
```



## 14. Exemplo7. cron rsync - estrutura arbórea: copia local todos os días

Mediante unha tarefa programada imos crear unha copia de respaldo incremental cada día de todo o que colga do directorio /home. A copia farase no propio host.

usuario@debianA:~\$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@debianA:~# rm -rf /var/backups/home /var/backups/usuario #Eliminar os directorios /var/backups/home, /var/backups/usuario e todo o que colga (-r) destes sen pedir confirmación (-f).

root@debianA:~# echo '0 0 \* \* \* root rsync -avz --progress /home/ /var/backups' >> /etc/crontab  
#Copiar de forma incremental a estrutura arbórea do contido do cartafol /home dentro do cartafol /var/backups, en modo arquivo (-a)(Ver NOTAS), de forma detallada (-v), comprimida (-z) e amosando o proceso de copia (--progress). **A primeira que se execute esta tarefa programada con esas rutas (orixe e destino) farase unha copia completa da ruta orixe na ruta destino, xa que na ruta destino non existe ningunha copia da ruta orixe. A partir da segunda vez que se execute esta tarefa programada (con esas rutas orixe e destino) farase unha copia incremental da ruta orixe na ruta destino, xa que na ruta destino existe unha copia completa da ruta orixe.** Esta tarefa programada execútase como o usuario root tódolos días ás 00:00. **Equivale a @daily ou @midnight**

root@debianA:~# echo '@daily root rsync -avz --progress /home/ /var/backups' >> /etc/crontab #Equivale ao comando anterior.

root@debianA:~# echo '@midnight root rsync -avz --progress /home/ /var/backups' >> /etc/crontab #Equivale ao comando anterior.

root@debianA:~# grep crontab /var/log/syslog Buscar o patrón crontab dentro do ficheiro de rexistros /var/log/syslog. Podemos observar que existe 1 nova entrada, a referente á recarga do ficheiro /etc/crontab.

Jan 11 20:26:01 debianA cron[358]: (\*system\*) RELOAD (/etc/crontab)

**IMPORTANTE:** Co realizado en /etc/crontab o primeiro día farase unha copia de respaldo completa e a partir do segundo día(inclusive) farase unha copia de respaldo incremental, pero soamente poderíamos recuperar a copia de respaldo do último día xa que sempre se foi gardando a copia na mesma ruta. Pero que pasa se quixeramos recuperar de mais días atrás? Ou que pasa se a copia queda corrupta? Entón, estaría ben mellorar a xestión de copias de respaldo: creando máis tarefas programadas ou modificando esta tarefa programada para que non execute soamente o comando rsync, senón que execute un script o cal xa permite a xestión que necesitamos. Así, poderíamos gardar primeiro a copia completa e logo en distintas rutas as copias incrementais.

root@debianA:~# cat > /root/backupHOME-incremental.sh <<EOF #Comezo do ficheiro a crear /root/backupHOME-incremental.sh

#!/bin/bash

#VARIABLES

DATA=\$(date +%F)

#main()

rsync -avz --progress /home/ /var/backups/\\${DATA}\_backupHOME

EOF #Fin do ficheiro a crear /root/backupHOME-incremental.sh

root@debianA:~# echo '0 0 \* \* \* root /bin/bash /root/backupHOME-incremental.sh' >> /etc/crontab #Engadir 1 liña ao ficheiro /etc/crontab, a cal permite executar como tarefa programada o script /root/backupHOME-incremental.sh como o usuario root tódolos días ás 00:00. Cada día de semana o ficheiro posúe un nome distinto, así, por exemplo:

Día 1(luns 2021-01-11) → 2021-01-11\_backupHOME  
Día 2(martes 2021-01-12) → 2021-01-12\_backupHOME  
Día 3(mércores 2021-01-13) → 2021-01-13\_backupHOME  
Día 4(xoves 2021-01-14) → 2021-01-14\_backupHOME  
Día 5(venres 2021-01-15) → 2021-01-15\_backupHOME  
Día 6(sábado 2021-01-16) → 2021-01-16\_backupHOME  
Día 7(domingo 2021-01-17) → 2021-01-17\_backupHOME  
...

**Agora farase unha copia de respaldo completa o primeiro día(2021-01-11) que se execute o script, e copias de respaldo incrementais os seguintes días(>2021-01-11), de tal xeito que poderemos recuperar dende calquera día, tendo que restaurar primeiro a copia completa e logo todas as copias incrementais ata a que consideremos.**

root@debianA:~# grep crontab /var/log/syslog Buscar o patrón crontab dentro do ficheiro de rexistros /var/log/syslog. Podemos observar que existe 1 nova entrada, a referente á recarga do ficheiro /etc/crontab.

Jan 11 20:52:01 debianA cron[358]: (\*system\*) RELOAD (/etc/crontab)

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.  
usuario@debianA:~\$

## 15. Exemplo8. rsync - estrutura arbórea: recuperación dunha copia local

Imos recuperar todo o que colga do directorio /home mediante o volcado da copia de respaldo completa e as sucesivas copias de respaldo incremental realizada na tarefa programada do Exemplo7. A recuperación farase de forma local dende o propio host. Para isto imos provocar o desastre en /home, é dicir, imos eliminar /home en debianA para logo recuperalo mediante a restauración da copia de respaldo completa e as sucesivas copias de respaldo incremental.

### Prerrequisito: Realizar o Exemplo7.

usuario@debianA:~\$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@debianA:~# rm -rf /home #Eliminar o directorio /home e todo o que colga (-r) deste sen pedir confirmación (-f).

root@debianA:~# ls -l /home #Listar de forma extendida o contido do directorio /home. Neste caso obteremos un erro xa que acabamos de eliminar ese directorio.

root@debianA:~# du -hs /home #Amosar en formato humano canto ocupa o cartafol /home. Neste caso obteremos un erro xa que acabamos de eliminar ese directorio.

root@debianA:~# mkdir /home #Crear o directorio /home.

**IMPORTANTE:** É necesario crear o directorio /home posto que as copias de respaldo realizadas co script do Exemplo7 posúen o contido do directorio /home, pero non o propio directorio /home.

root@debianA:~# for i in \$(ls -d /var/backups/2021\*); do rsync -avz --progress \${i}/ /home/ ; done #Restaurar a copia de respaldo completa e as sucesivas copias incrementais do directorio /home

**Agora restauramos mediante a copia de respaldo completa do primeiro día(2021-01-11) e as copias de respaldo incrementais dos seguintes días(>2021-01-11). Estamos a considerar que esta última copia non está corrupta, senón deberíamos intentar o respaldo con copias de respaldo completa e incrementais de días anteriores.**

root@debianA:~# ls -l /home #Listar de forma extendida o contido do directorio /home.

root@debianA:~# du -hs /home #Amosar en formato humano canto ocupa o cartafol /home.

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.

usuario@debianA:~\$

## 16. Exemplo9. rdiff-backup - estrutura arbórea: copia local

Imos crear unha copia de respaldo de todo o que colga do directorio /home. A copia farase no propio host.

```
usuario@debianA:~$ ls -l /home #Listar de forma extendida o contido do directorio /home
```

```
usuario@debianA:~$ du -hs /home #Amosar en formato humano canto ocupa o cartafol /home
```

```
usuario@debianA:~$ ls -ld /var/backups/ #Listar soamente os permisos do cartafol /var/backups, é dicir, listar os permisos do propio cartafol pero non os do seu contido.
```

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# dpkg -l rdiff-backup ; [ $(echo $?)-eq '1' ] && apt update && apt -y install rdiff-backup #Verificar se o paquete rdiff-backup está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase.
```

```
root@debianA:~# rdiff-backup /home /var/backups/backup-home-rdiff #Realizar a copia do contido do directorio /home dentro do directorio /var/backups/backup-home-rdiff. Se o directorio de destino non existe créase. Como é a primeira que se empregan esas rutas (orixe e destino) faise unha copia completa da ruta orixe na ruta destino, xa que na ruta destino non existe ningunha copia da ruta orixe.
```

```
root@debianA:~# rdiff-backup /home/ /var/backups/backup-home-rdiff-2 #Equivale ao comando anterior empregando unha ruta_destino diferente. Se o directorio destino non existe créase. Como é a primeira que se empregan esa ruta destino faise unha copia completa da ruta orixe na ruta destino, xa que na ruta destino non existe ningunha copia da ruta orixe.
```

```
root@debianA:~# ls -l /var/backups/backup-home-rdiff #Listar de forma extendida o contido do directorio /var/backups/backup-home-rdiff
```

```
root@debianA:~# ls -l /var/backups/backup-home-rdiff-2 #Listar de forma extendida o contido do directorio /var/backups/backup-home-rdiff-2
```

**IMPORTANTE:** Notar que non existe diferenca na *ruta\_orixe* entre os 2 comandos anteriores, é dicir, para rdiff-backup as *rutas\_orixe* **/home** e **/home/** son equivalentes, polo cal obtense o mesmo contido nas *rutas\_destino* */var/backups/backup-home-rdiff* e */var/backups/backup-home-rdiff-2*.

```
root@debianA:~# du -hs /var/backups/backup-home-rdiff* #Amosar en formato humano canto ocupan o cartafol /var/backups/backup-home-rdiff e /var/backups/backup-home-rdiff-2
```

```
root@debianA:~# rdiff-backup -l /var/backups/backup-home-rdiff #Amosar listado con todas as copias xeradas en /var/backups/backup-home-rdiff.
```

```
Found 0 increments:
Current mirror: Mon Jan 11 23:33:14 2021
```

Neste caso como soamente fixemos unha copia de respaldo, a copia é completa(mirror) e non temos ningunha copia incremental a maiores.

```
root@debianA:~# cp -pv /etc/passwd /home/usuario/ #Copiar o ficheiro /etc/passwd en modo verbose (detallado) e preservando permisos e datas do ficheiro dentro do directorio /home/usuario/
```

```
root@debianA:~# chown usuario. /home/usuario/passwd #Cambiar usuario propietario usuario e grupo propietario usuario ao ficheiro /home/usuario/passwd
```

```
root@debianA:~# rdiff-backup /home /var/backups/backup-home-rdiff #Realizar de novo a copia do contido do directorio /home dentro do directorio /var/backups/backup-home-rdiff. Como xa non é a primeira que se empregan esas rutas (orixe e destino) e existen cambios en /home/usuario respecto á primeira copia pois faise unha copia diferencial respecto á completa realizada anteriormente.
```

```
root@debianA:~# rdiff-backup -l /var/backups/backup-home-rdiff #Amosar listado con todas as copias xeradas en /var/backups/backup-home-rdiff.
```

```
Found 1 increments:
  increments.2021-01-11T23:33:14+01:00.dir   Mon Jan 11 23:33:14 2021
Current mirror: Mon Jan 11 23:50:09 2021
```

Neste caso como existen cambios sobre a copia de respaldo realizada anteriormente teremos unha copia incremental sobre a anterior copia completa, obtendo unha copia de respaldo diferencial(mirror).

```
root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.
```

```
usuario@debianA:~$
```

## 17. Exemplo10. rdiff-backup - estrutura arbórea: copia remota

Imos crear unha copia de respaldo de todo o que colga do directorio /home. A copia farase remotamente de forma cifrada mediante conexión SSH.

```
kali@kaliB:~$ ssh usuario@debianA ls -l /home #Listar de forma extendida o contido do directorio /home do host
debianA. O comando é executado en debianA accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é
amosada en kaliB
```

```
kali@kaliB:~$ ssh usuario@debianA du -hs /home #Amosar en formato humano canto ocupa o cartafol /home do
host debianA. O comando é executado en debianA accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa
saída é amosada en kaliB
```

```
kali@kaliB:~$ ssh usuario@debianA ls -ld /var/backups/ #Listar soamente os permisos do cartafol /var/backups
do host debianA, é dicir, listar os permisos do propio cartafol pero non os do seu contido. O comando é executado en debianA
accedendo co usuario de nome usuario nunha conexión cifrada SSH, e a súa saída é amosada en kaliB
```

```
kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) en kaliB a través dos permisos configurados co
comando sudo (/etc/sudoers, visudo)
```

```
root@kaliB:~# dpkg -l rdiff-backup ; [ $(echo $?) -eq '1' ] && apt update && apt -y install rdiff-
backup #Verificar se o paquete rdiff-backup está instalado. Se non está instalado, actualízase a lista de paquetes dos
repositorios e instálase.
```

```
root@kaliB:~# rdiff-backup usuario@debianA::/home/ /var/backups/backup-remoto-home-rdiff
#Copiar a estrutura arbórea do contido do cartafol /home de debianA dentro do cartafol /var/backups/backup-remoto-home-
rdiff de kaliB. Se o directorio de destino non existe créase. O comando é executado en debianA accedendo co usuario de
nome usuario nunha conexión cifrada SSH, e a súa saída é amosada en kaliB. Como é a primeira que se empregan esas
rutas (orixe e destino) faise unha copia completa da ruta orixe na ruta destino, xa que na ruta destino non
existe ningunha copia da ruta orixe.
```

```
root@kaliB:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/ do host kaliB
```

```
root@kaliB:~# du -hs /var/backups #Amosar en formato humano canto ocupa o cartafol /var/backups do host
kaliB
```

```
root@kaliB:~# ls -l /var/backups/backup-remoto-home-rdiff #Listar de forma extendida o contido do
directorio /var/backups/backup-remoto-home-rdiff do host kaliB
```

```
root@kaliB:~# du -hs /var/backups/backup-remoto-home-rdiff #Amosar en formato humano canto ocupa
o cartafol /var/backups/backup-remoto-home-rdiff do host kaliB
```

```
root@kaliB:~# rdiff-backup usuario@debianA::/home /var/backups/backup-remoto-home-rdiff
#Copiar a estrutura arbórea do contido do cartafol /home de debianA dentro do cartafol /var/backups/backup-remoto-home-
rdiff de kaliB. Se o directorio de destino non existe créase. O comando é executado en debianA accedendo co usuario de
nome usuario nunha conexión cifrada SSH, e a súa saída é amosada en kaliB. Como non existe diferenca entre as rutas
orixe /home e /home/ é a segunda vez que se empregan esas rutas (orixe e destino), co cal faise unha copia
diferencial da ruta orixe na ruta destino, xa que na ruta destino existe unha copia completa da ruta orixe.
```

**IMPORTANTE:** Para rdiff-backup as *rutas\_orixe* **/home** e **/home/** son equivalentes.

```
root@kaliB:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/ do host kaliB
```

```
root@kaliB:~# du -hs /var/backups #Amosar en formato humano canto ocupa o cartafol /var/backups do host
kaliB
```

```
root@kaliB:~# ls -l /var/backups/backup-remoto-home-rdiff #Listar de forma extendida o contido do
directorio /var/backups/backup-remoto-home-rdiff do host kaliB
```

```
root@kaliB:~# du -hs /var/backups/backup-remoto-home-rdiff #Amosar en formato humano canto ocupa
o cartafol /var/backups/backup-remoto-home-rdiff do host kaliB
```

```
root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de kali.
```

```
kali@kaliB:~$
```

## 18. Exemplo11. cron rdiff-backup - estrutura arbórea: copia local todos os días

Mediante unha tarefa programada imos crear unha copia de respaldo diferencial cada día de todo o que colga do directorio /home. A copia farase no propio host.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# echo '0 0 * * * root rdiff-backup /home/ /var/backups/TP-backup-home-rdiff' >> /etc/crontab #Copiar de forma diferencial a estrutura arbórea do contido do cartafol /home dentro do cartafol /var/backups/TP-backup-home-rdiff. A primeira que se execute esta tarefa programada con esas rutas (orixe e destino) farase unha copia completa da ruta orixe na ruta destino, xa que na ruta destino non existe ningunha copia da ruta orixe. A partir da segunda vez que se execute esta tarefa programada (con esas rutas orixe e destino) farase unha copia diferencial da ruta orixe na ruta destino, xa que na ruta destino existe unha copia completa da ruta orixe. Esta tarefa programada execútase como o usuario root tódolos días ás 00:00. Equivale a @daily ou @midnight
```

```
root@debianA:~# echo '@daily root rdiff-backup /home/ /var/backups/TP-backup-home-rdiff' >> /etc/crontab #Equivale ao comando anterior.
```

```
root@debianA:~# echo '@midnight root rdiff-backup /home/ /var/backups/TP-backup-home-rdiff' >> /etc/crontab #Equivale ao comando anterior.
```

```
root@debianA:~# grep crontab /var/log/syslog Buscar o patrón crontab dentro do ficheiro de rexistros /var/log/syslog. Podemos observar que existe 1 nova entrada, a referente á recarga do ficheiro /etc/crontab.
```

```
Jan 11 23:56:01 debianA cron[358]: (*system*) RELOAD (/etc/crontab)
```

**Co realizado en /etc/crontab farase unha copia de respaldo completa o primeiro día(2021-01-11) que se execute o script, e copias de respaldo diferenciais os seguintes días(>2021-01-11), de tal xeito que poderemos recuperar dende calquera día, soamente indicando no comando rdiff-backup un parámetro coa data a ter en conta para a restauración. Por exemplo:**

```
root@debianA:~# rdiff-backup -r now /var/backups/TP-backup-home-rdiff /home/ #Permitir copiar a última copia de respaldo diferencial realizada, é dicir, recuperar a última copia mirror do cartafol /home realizada con rdiff-backup
```

```
root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.
```

```
usuario@debianA:~$
```



## 19. Exemplo12. rdiff-backup - estrutura arbórea: recuperación dunha copia local

Imos recuperar todo o que colga do directorio /home mediante o volcado da copia de respaldo completa e a última copia diferencial realizada na tarefa programada do Exemplo11. A recuperación farase de forma local dende o propio host.

### Prerrequisito: Realizar o Exemplo11.

usuario@debianA:~\$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@debianA:~# rm -rf /home #Eliminar o directorio /home e todo o que colga (-r) deste sen pedir confirmación (-f).

root@debianA:~# ls -l /home #Listar de forma extendida o contido do directorio /home. Neste caso obteremos un erro xa que acabamos de eliminar ese directorio.

root@debianA:~# du -hs /home #Amosar en formato humano canto ocupa o cartafol /home. Neste caso obteremos un erro xa que acabamos de eliminar ese directorio.

root@debianA:~# mkdir /home #Crear o directorio /home.

**IMPORTANTE:** É necesario crear o directorio /home posto que as copias de respaldo realizadas posúen o contido do directorio /home, pero non o propio directorio /home.

root@debianA:~# rdiff-backup -r now /var/backups/TP-backup-home-rdiff /home/ #Volcar a última copia de respaldo realizada, é dicir, recuperar a copia completa e a última diferencial de /home

root@debianA:~# rdiff-backup -r 2D /var/backups/TP-backup-home-rdiff /home/ #Volcar a copia de respaldo realizada fai 2días ou a máis próxima logo de 2 días, é dicir, recuperar a copia completa e a diferencial de /home de fai 2 días ou a máis próxima logo de 2 días.

root@debianA:~# ls -l /home #Listar de forma extendida o contido do directorio /home.

root@debianA:~# du -hs /home #Amosar en formato humano canto ocupa o cartafol /home.

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.

usuario@debianA:~\$

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**