

# Práctica BRS - Backup

## ESCENARIO

### Máquinas virtuales ou físicas:

RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado

Rede: 192.168.120.0/24

Rede Interna(enp0s3) e NAT(enp0s8)

SO: Debian amd64 xfce instalado

sda: SO instalado

BIOS: Permite arranque HD

User/Passwd: usuario/abc123.

User/Passwd: root/abc123.

### Máquina A:

IP/MS: 192.168.120.100/24

Servidor SSH

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

Servidor Amanda (amanda-server)

Cliente Amanda (amanda-client)

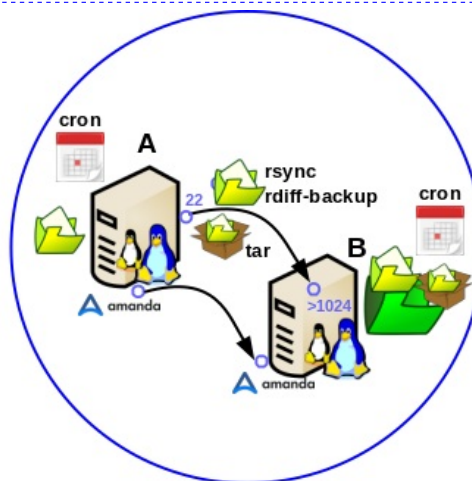
### Máquina B:

IP/MS: 192.168.120.101/24

Cliente SSH

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

Cliente Amanda (amanda-client)



**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 BRS - RAID por Software: mdadm](#)

- [Amanda](#) [Amanda - fpngenred.es](#)

- [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

- 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

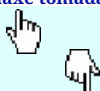
- cron: /etc/crontab (global) , /var/spool/cron/crontabs/username

**/etc/crontab (global)**

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



Imaxe tomada de <https://linuxconfig.org>



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

\$ man rdiff-backup

\$ rdiff-backup 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 list increments 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 restore 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 list increments

**doc-examples rdiff-backup**

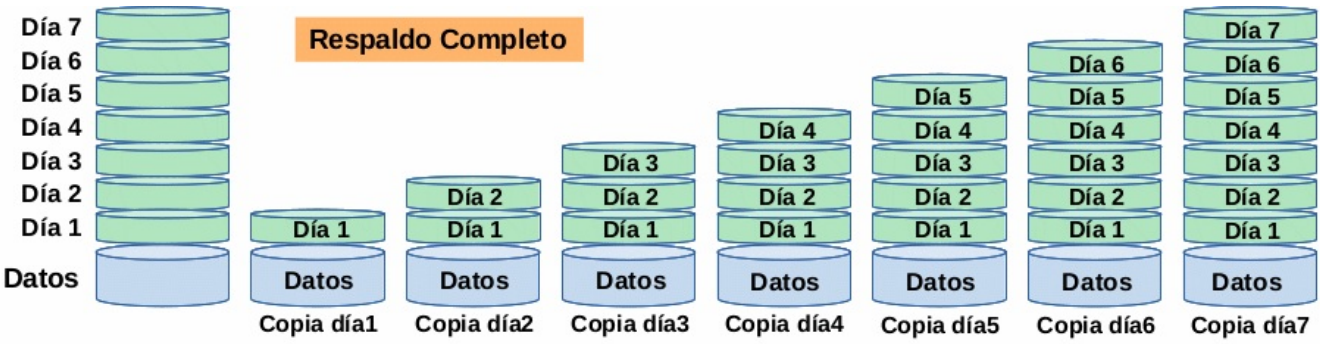
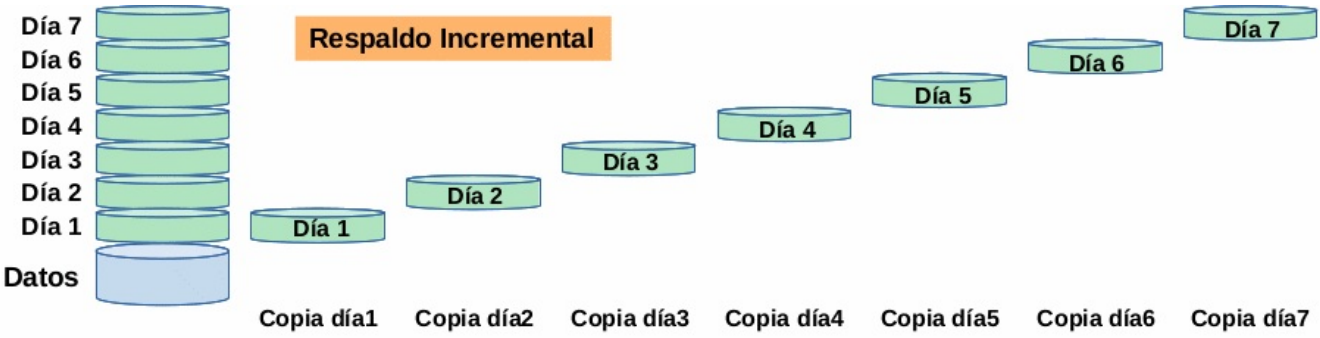
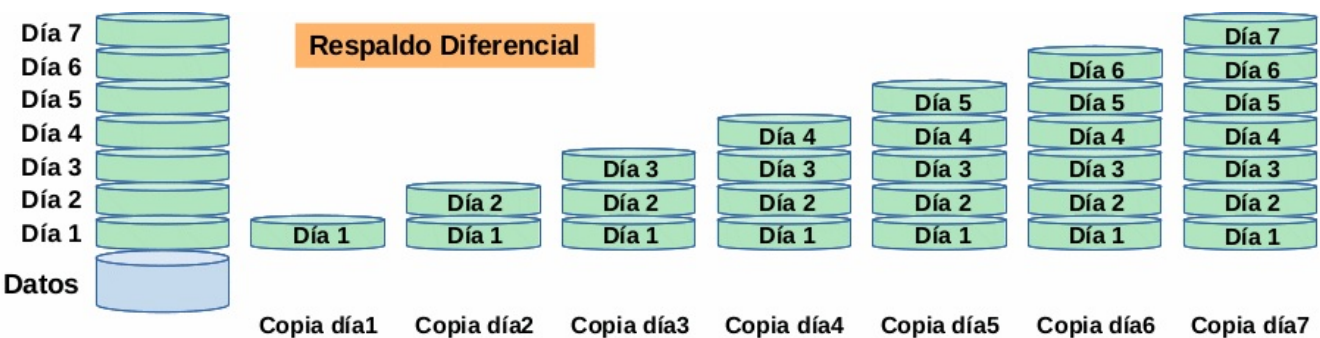


rdiff-backup-data

→ 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	 <p><b>Respaldo Completo</b></p> <p>Copiase todo, é dicir, cada día gárdase todo (existan ou non cambios)</p> <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	 <p><b>Respaldo Incremental</b></p> <p>Copiase soamente o que cambia dunha copia a outra anterior partindo inicialmente dunha copia completa, é dicir, non se garda a copia completa soamente as diferenzas existentes cada día.</p> <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	 <p><b>Respaldo Diferencial</b></p> <p>Gárdase todo o que cambia dende a copia completa, é dicir, cada día gárdanse todas as diferenzas incluíndo ás dos días anteriores.</p> <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>

### 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.



### Máquina virtual A: Debian amd64

#### 1. Cambiar hostname da máquina virtual A. Por debianA como hostname:

Na contorna gráfica abrir un terminal e executar:

usuario@debian:~\$ su - #Acceder á consola de root(administrador) a través do comando *su*, o cal solicita o contrasinal do usuario *root*.

Por como contrasinal **abc123**. (Olo que o contrasinal ten un caracter punto final).

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 su na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debian:~\$ exit #Pegar o terminal saíndo da consola local do usuario **usuario**.

#### 2. Configurar a rede. Na contorna gráfica abrir un terminal e executar:

usuario@debianA:~\$ su - #Acceder á consola de root(administrador) a través do comando *su*, o cal solicita o contrasinal do usuario

*root*. Por como contrasinal **abc123**. (Olo que o contrasinal ten un caracter punto final).

root@debianA:~# /etc/init.d/avahi-daemon stop || systemctl stop avahi-daemon #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:~# systemctl stop NetworkManager || pkill NetworkManager #Parar o demo NetworkManager(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 NetworkManager #Impide que o servizo NetworkManager 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

### 3. 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 de **root** na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~\$

### 4. Cambiar hostname da máquina virtual B. Por debianB como hostname:

Na contorna gráfica abrir un terminal e executar:

usuario@debian:~\$ su - #Acceder á consola de root(administrador) a través do comando su, o cal solicita o contrasinal do usuario root.

Por como contrasinal **abc123**. (Olo que o contrasinal ten un caracter punto final).

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

root@debian:~# echo 'kernel.hostname=debianB' >> /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 su na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debian:~\$ exit #Pegar o terminal saíndo da consola local do usuario **usuario**.

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

usuario@debianB:~\$ su - #Acceder á consola de root(administrador) a través do comando su, o cal solicita o contrasinal do usuario

root. Por como contrasinal **abc123**. (Olo que o contrasinal ten un caracter punto final).

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

root@debianB:~# systemctl stop NetworkManager || pkill NetworkManager #Parar o demo NetworkManager(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@debianB:~# systemctl disable NetworkManager #Impide que o servizo NetworkManager sexa iniciado no arranque xerando os links K\* nos runlevels (/etc/rcX.d)

root@debianB:~# ip addr add 192.168.120.101/24 dev enps03 #Configurar a tarxeta de rede interna enp0s3, coa IP: 192.168.120.101 e máscara de subrede: 255.255.255.0.

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

root@debianB:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local enp0s3

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

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

SSH

### 6. **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:

usuario@debianB:~\$ 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.

usuario@debianB:~\$ 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.

usuario@debianB:~\$ 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:~\$

## 7. 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:~$ su - #Acceder á consola de root(administrador) a través do comando su, o cal solicita o contrasinal do usuario root. Por como contrasinal abc123. (Ollo que o contrasinal ten un caracter punto final).
```

```
root@debianA:~# tar cvfj /var/backups/backupHOME.tar.bz2 /home #Crear un ficheiro comprimido mediante bzip2 do que foi empacutado mediante tar, o cal contén o contido do directorio /home eliminando as barras iniciais '/' dos nomes, é dicir, empaqueta home e non /home. Independentemente 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 su 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 usuario en debianB.
```

```
usuario@debianB:~$
```



## 8. 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.

```
usuario@debianB:~$ 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 debianB
```

```
usuario@debianB:~$ 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 debianB
```

```
usuario@debianB:~$ 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 debianB
```

```
usuario@debianB:~$ su - #Acceder á consola de root(administrador) a través do comando su, o cal solicita o contrasinal do usuario
root. Por como contrasinal abc123. (Ollo que o contrasinal ten un carácter punto final).
```

```
root@debianB:~# ssh usuario@debianA tar cvfj - /home > /var/backups/backupHOME.tar.bz2 #Crear un
ficheiro comprimido mediante bzip2 en debianB, 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 debianB. Independientemente do
número de veces que se execute este comando sempre farase unha copia de respaldo completa
```

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

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

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

```
root@debianB:~# exit #Saír da consola local su na que estabamos a traballar para voltar á consola local del usuario.
```

```
usuario@debianB:~$
```

## 9. 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:~\$ su - #Acceder á consola de root(administrador) a través do comando su, o cal solicita o contrasinal do usuario root. Por como contrasinal **abc123**. (Ollo que o contrasinal ten un caracter punto final).

```
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 sobreescíbese 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 su na que estabamos a traballar para voltar á consola local de **usuario**.

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

## 10. 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. Así, pechar sesión gráfica do usuario **usuario** e iniciar sesión gráfica co usuario **root**. Abrir unha consola e executar:

```
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
```

## 11. 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:~$ su - #Acceder á consola de root(administrador) a través do comando su, o cal solicita o contrasinal do usuario root. Por como contrasinal abc123. (Ollo que o contrasinal ten un caracter punto final).
```

```
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 su na que estabamos a traballar para voltar á consola local de usuario.
```

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

## 12. Exemplo6. rsync - estrutura arborea: 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.

usuario@debianB:~\$ 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 debianB

usuario@debianB:~\$ 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 debianB

usuario@debianB:~\$ 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 debianB

usuario@debianB:~\$ su - #Acceder á consola de root(administrador) a través do comando *su*, o cal solicita o contrasinal do usuario *root*. Por como contrasinal **abc123**. (Olo que o contrasinal ten un carácter punto final).

root@debianB:~# 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@debianB:~# rsync -avz --progress --rsh="ssh" usuario@debianA:/home/ /var/backups #Copiar a estrutura arborea do contido do cartafol /home de debianA dentro do cartafol /var/backups de debianB, 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 debianB. **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@debianB:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/ do host debianB

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

root@debianB:~# rsync -avz --progress --rsh="ssh" usuario@debianA:/home/ /var/backups #Copiar a estrutura arborea do contido do cartafol /home de debianA dentro do cartafol /var/backups de debianB, 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 debianB. **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@debianB:~# rsync -avz --progress --rsh="ssh" usuario@debianA:/home /var/backups #Copiar a estrutura arborea /home de debianA dentro do cartafol /var/backups de debianB, 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 debianB. **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 diferenza 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@debianB:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/ do host debianB

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

root@debianB:~# exit #Saír da consola local su na que estabamos a traballar para voltar á consola local del **usuario**.  
usuario@debianB:~\$



### 13. Exemplo7. cron rsync - estrutura arborea: 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:~\$ su - #Acceder á consola de root(administrador) a través do comando su, o cal solicita o contrasinal do usuario root. Por como contrasinal **abc123**. (Ollo que o contrasinal ten un caracter punto final).

```
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 arborea 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 executase 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 su na que estabamos a traballar para voltar á consola local de \*\*usuario\*\*.](#)

usuario@debianA:~\$

#### 14. 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. Así, pechar sesión gráfica do usuario **usuario** e iniciar sesión gráfica co usuario **root**. Abrir unha consola e executar:

##### Prerrequisito: Realizar o Exemplo7.

```
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.
```

## 15. 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:~$ su - #Acceder á consola de root(administrador) a través do comando su, o cal solicita o contrasinal do usuario root. Por como contrasinal abc123. (Olo que o contrasinal ten un caracter punto final).
```

```
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 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 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 diferenza 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 list increments /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 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 list increments /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 su na que estabamos a traballar para voltar á consola local de usuario.
```

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



## 16. Exemplo10. rdiff-backup - estrutura arborea: 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.

```
usuario@debianB:~$ 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 debianB
```

```
usuario@debianB:~$ 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 debianB
```

```
usuario@debianB:~$ 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 debianB
```

```
usuario@debianB:~$ su - #Acceder á consola de root(administrador) en debianB a través do comando su, o cal solicita o contrasinal
do usuario root. Por como contrasinal abc123. (Olo que o contrasinal ten un caracter punto final).
```

```
root@debianB:~# 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@debianB:~# rdiff-backup backup usuario@debianA::/home/ /var/backups/backup-remoto-home-rdiff
#Copiar a estrutura arborea do contido do cartafol /home de debianA dentro do cartafol /var/backups/backup-remoto-home-rdiff de
debianB. 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 debianB. 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@debianB:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/ do host debianB
```

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

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

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

```
root@debianB:~# rdiff-backup backup usuario@debianA::/home /var/backups/backup-remoto-home-rdiff
#Copiar a estrutura arborea do contido do cartafol /home de debianA dentro do cartafol /var/backups/backup-remoto-home-rdiff de
debianB. 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 debianB. Como non existe diferenca entre as rutas orixe /home e /home/ é a
segunda vez que se emmpregan 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@debianB:~# ls -l /var/backups/ #Listar de forma extendida o contido do directorio /var/backups/ do host debianB
```

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

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

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

```
root@debianB:~# exit #Saír da consola local su na que estabamos a traballar para voltar á consola local del usuario.
usuario@debianB:~$
```

## 17. Exemplo11. cron rdiff-backup - estrutura arborea: 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:~\$ su - #Acceder á consola de root(administrador) a través do comando su, o cal solicita o contrasinal do usuario root. Por como contrasinal **abc123**. (Olo que o contrasinal ten un caracter punto final).

root@debianA:~# echo '0 0 \* \* \* root rdiff-backup backup /home/ /var/backups/TP-backup-home-rdiff' >> /etc/crontab #Copiar de forma diferencial a estrutura arborea 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 backup /home/ /var/backups/TP-backup-home-rdiff' >> /etc/crontab
#Equivale ao comando anterior.
```

```
root@debianA:~# echo '@midnight root rdiff-backup 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 restore --at 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 su na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~\$

## 18. 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. Así, pechar sesión gráfica do usuario **usuario** e iniciar sesión gráfica co usuario **root**. Abrir unha consola e executar:

### Prerrequisito: Realizar o Exemplo11.

```
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 restore --at 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 restore --at 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.
```

# Apéndice

## A. Amanda vs Bacula



### A.1 Amanda

Amanda (Advanced Maryland Automatic Network Disk Archiver) é unha solución de backup centralizada que se enfoca principalmente na automatización das copias de seguridade, permitindo a realización de backups de sistemas Windows e Linux nunha única infraestrutura.

#### Facilidade de uso

Amanda é coñecida por ser relativamente sinxela de configurar e utilizar. Está pensada para entornos que non requiren unha xestión de backup extremadamente complexa, e o seu enfoque é a facilidade de uso con interfaces gráficas ou de liña de comandos.

#### Plataformas

Amanda funciona ben en entornos mixtos que inclúen tanto sistemas GNU/Linux como Windows, xa que proporciona clientes para as dúas plataformas.

#### Escalabilidade

Amanda é adecuada para pequenas e medianas redes, pero tamén pode escalar para redes máis grandes, se ben non está tan optimizada para xestionar miles de clientes.

#### Características principais

- Copias de seguridade incrementais, completas e diferenciais.
- Copias de seguridade baseadas en rede.
- Pode realizar backups en discos locais ou remotos.
- Administración centralizada dende un servidor de backups.
- Opcionalmente, pode utilizarse con formatos de respaldo como tar e dump.
- Compatibilidade con redes mixtas (GNU/Linux, Windows).

#### Usabilidade

Ideal para pequenas e medianas empresas ou entornos de escritorio. Requere un certo coñecemento técnico, pero a súa configuración non é tan complexa como Bacula.

### A.2 Bacula

Bacula é unha solución de copias de seguridade máis completa e flexible, destinada a satisfacer as necesidades de grandes infraestruturas empresariais. Permite realizar copias de seguridade, restauracións e verificacións de datos nunha rede de servidores.

#### Facilidade de uso

Bacula é moito máis potente e flexible que Amanda, pero tamén máis complexo de configurar e xestionar. Necesita unha comprensión máis profunda dos seus componentes (Director, Storage, Client) para unha configuración e operación adecuadas.

#### Plataformas

Bacula tamén é compatible con varias plataformas, incluíndo GNU/Linux, Windows, macOS e outros sistemas Unix.

#### Escalabilidade

Bacula está deseñado para redes de maior tamaño, sendo ideal para empresas que necesiten realizar copias de seguridade a gran escala ou en entornos complexos con múltiples servidores e clientes.

## Características principais

- Máis granularidade na configuración de backups (incrementais, diferenciais, completos).
- Permite realizar copias de seguridade en discos, cintas ou nubes.
- Xestión detallada de backup e restauración.
- Compatibilidade cos formatos de cinta (tape) e soporte para hardware de backup avanzado.
- Gran flexibilidade e soporte para distintos tipos de almacenamiento.
- Funcións avanzadas de recuperación e verificación de datos.
- Xestión detallada de políticas de backup e de programación.

## Usabilidade

Ideal para grandes empresas ou redes empresariais con múltiples servidores. Requiere un maior nivel de coñecemento técnico para configuralo e mantelo funcionando adecuadamente. Ofrece unha interface gráfica e unha interface de liña de comandos, pero a súa complexidade xeralmente fai que sexa mellor para usuarios máis experimentados.

## Comparativa

Características	Amanda	Bacula
Facilidade de uso	Relativamente sinxelo de configurar	Máis complexo, require maior coñecemento
Escalabilidade	Bo para pequenas e medianas redes	Excelente para redes grandes e complexas
Plataformas soportadas	Windows, GNU/Linux, Unix	Windows, GNU/Linux, macOS, Unix
Soporte de cintas	Limitado, mellor para discos ou almacéns en rede	Soporte avanzado para cintas e hardware especializado
Automatización	Alta, pero sen a mesma flexibilidade que Bacula	Alta, con políticas e programación detallada
Configuración	Configuración máis sinxela	Configuración detallada e complexa
Recuperación de datos	Boa, pero menos avanzada	Avanzada, con múltiples opcións de recuperación
Ideal para	Empresas pequenas ou medianas, usuarios case que novatos	Empresas grandes, administradores con experiencia

## Cando elixir Amanda ou Bacula?

**Elixe Amanda** se buscas unha solución sinxela, de fácil configuración e xestión para unha rede de tamaño pequeno a mediano, con sistemas operativos mixtos. Amanda tamén é mellor se o teu obxectivo principal é facer backups sen necesidade de funcións avanzadas ou hardware especializado (como cintas).

**Elixe Bacula** se precisas unha solución de backup máis potente e flexible para grandes redes, con requisitos complexos, incluíndo a necesidade de soportar cintas ou outras formas avanzadas de almacenamento. Bacula tamén é ideal se precisas funcións avanzadas de recuperación ou backup a gran escala.

## Conclusión

Ambas ferramentas son boas opcións para realizar copias de seguridade, pero a elección depende das túas necesidades específicas. Se precisas algo sinxelo e directo, **Amanda** pode ser mellor. Se necesitas unha solución de backup empresarial máis robusta e flexible, **Bacula** será a mellor opción.



## B. Amanda para facer copias de seguridade de sistemas Windows e GNU/Linux

Amanda (Advanced Maryland Automatic Network Disk Archiver) é unha ferramenta de código aberto deseñada para realizar copias de seguridade nunha rede. Utilízase para facer respaldos en sistemas tanto en entornos **Windows** como **GNU/Linux**.

### Como funciona Amanda?

Amanda funciona mediante un servidor de copias de seguridade centralizado, que xestiona o proceso de realizar copias de seguridade nos clientes da rede (tanto sistemas Windows como GNU/Linux). A estrutura de Amanda permite configurar un servidor de respaldo no que se xestionan os backups, e os clientes envían os datos a este servidor.

### Copias de seguridade en Windows con Amanda

1. **Cliente Amanda para Windows:** En sistemas Windows, Amanda utiliza o axente chamado **Windows client (amanda-client)**. Este cliente debe instalarse no sistema Windows que se vai a respaldar.
2. **Configuración do cliente:** Unha vez instalado, o cliente Windows configúraselle para permitir que Amanda faga copias de seguridade de directorios ou ficheiros específicos. O servidor de Amanda é o que orquestra as copias, mentres que o cliente de Windows envía os datos a través da rede.
3. **Tipos de respaldo:** En Windows, Amanda pode realizar copias completas ou incrementais dos ficheiros e directorios seleccionados, ao igual que en sistemas GNU/Linux.

### Copias de seguridade en GNU/Linux con Amanda

1. **Cliente de Linux:** En sistemas GNU/Linux, tamén debe instalarse o cliente de Amanda, e na maioría dos casos é o software estándar de **amanda-client**.
2. **Configuración:** En Linux, normalmente configúraselle o arquivo `/etc/amanda/<nome do plan de respaldo>/disklist`, onde se especifican os directorios e sistemas de ficheiros que se respaldarán.
3. **Copias de seguridade:** Amanda pode facer copias completas, incrementais e diferenciais, e normalmente usa ferramentas estándar como `tar` ou `dump` en Linux para realizar os respaldos.

### Vantaxes de usar Amanda

- **Multi-plataforma:** É unha solución eficiente para respaldar tanto sistemas Windows como GNU/Linux, o que a converte nunha opción ideal para redes mixtas.
- **Automatización:** Pódense automatizar as copias de seguridade segundo horarios e frecuencias específicas.
- **Escalabilidade:** Pódese usar para redes pequenas ou grandes con múltiples clientes.
- **Backup centralizado:** Amanda permite xestionar as copias de seguridade desde un servidor centralizado, o que facilita a administración.

### Resumo

Amanda pódese usar para facer copias de seguridade de sistemas tanto **Windows** como **GNU/Linux**. Para Windows, necesitas instalar un cliente de Amanda específico e configuralo para respaldar os datos necesarios. En GNU/Linux, Amanda utiliza ferramentas de respaldo estándar e ofrece unha gran flexibilidade para xestionar copias de seguridade de maneira centralizada e automatizada.

# Exemplos de configuración de backup con Amanda en Debian

A continuación, móstraranse exemplos de configuración de Amanda para realizar copias de seguridade completas, diferencias e incrementais en sistemas GNU/Linux Debian. Estes exemplos cubrirán a configuración básica de Amanda desde a instalación ata a creación de plans de backup para as diferentes copias de seguridade.

## Exemplo1: Backups en localhost con Amanda

### 1. Configuración do Servidor Amanda (Servidor de Backup)

Amanda require un servidor centralizado para xestionar os backups. Aquí estableceremos os arquivos de configuración no servidor de backups.

#### 1.1 Instalación de Amanda en Debian

Primeiro, debes instalar Amanda no sistema Debian. Para iso, podes executar os seguintes comandos:

```
# apt update
# apt -y install amanda-client amanda-server
```

#### 1.2 Crear o arquivo amanda.conf e a estrutura de directorios necesaria

O arquivo **amanda.conf** atópase en `/etc/amanda/<nome_do_plan>/amanda.conf`. O `nome_do_plan` pode ser calquera nome que queiras dar ao teu plan de backup, por exemplo, `daily`, `weekly`, etc.

A continuación, veremos como configurar o plan **daily**:

Os arquivos de configuración de Amanda para o plan `daily` atópanse en `/etc/amanda/daily`. Crea os directorios e establece os permisos necesarios onde se almacenarán información histórica das copias, os rexistros, índices e volumes virtuais de Amanda.

```
# mkdir -p /etc/amanda/daily/curinfo
# mkdir -p /etc/amanda/daily/log
# mkdir -p /etc/amanda/daily/index
# mkdir -p /var/backups/amanda/daily/vtapes

# chown -R backup:backup /etc/amanda/daily /var/backups/amanda/daily
```

O arquivo principal de configuración do plan será `/etc/amanda/daily/amanda.conf`. Crea este ficheiro según se indica en **Código1**

#### 1.3. Crear o ficheiro disklist e establece de novo os permisos correctos:

Agora, configura que directorios se deben incluír no backup e o tipo de copia para cada un (completo, diferencial ou incremental). Crea o ficheiro `disklist` en `/etc/amanda/daily/disklist` co seguinte contido mediante o comando:

```
# echo 'localhost /home completo
localhost /etc diferencial
localhost /var/log incremental' > /etc/amanda/daily/disklist
```

Establece os permisos correctos:

```
# chown -R backup:backup /etc/amanda
```

#### 1.4. Crear os volumes e etiquetar os slots

Agora necesitas crear e etiquetar os volumes virtuais nos slots. Estes pasos deben ser realizados polo usuario `backup`:

```
# for i in $(seq 1 14); do
    mkdir -p /var/backups/amanda/daily/vtapes/slot$i
done

# chown -R backup:backup /var/backups/amanda

# su backup -s /bin/bash -c "for i in \$(seq 1 14); do
    amlabel daily Etiqueta\${i} slot \${i}
done"
```

```

# echo 'org "daily" # Nome da configuración ou grupo de copias de seguridade.
mailto "usuario@dominio.local" # Correo electrónico para recibir informes de execución.
dumpuser "backup" # Usuario encargado de executar Amanda.
inparallel 1 # Número de procesos simultáneos permitidos para as copias.
netusage 100 # Ancho de banda máximo permitido (en KB/s).
dumpcycle 7 # Ciclo de copias en días. Aquí, 7 días.
runspcycle 7 # Número de execucións completas durante o ciclo (unha por día).
runtapes 1 # Non se usa, xa que as copias son exclusivamente a disco. (Mantida por compatibilidade).
tapecycle 14 # Número de volumes virtuais en rotación. Útil para control de espazo no disco.
infofile "/etc/amanda/daily/curinfo" # Ficheiro que almacena información histórica das copias.
logdir "/etc/amanda/daily/log" # Directorio onde se almacenan os rexistros.
indexdir "/etc/amanda/daily/index" # Directorio onde se gardan os índices das copias.
tapelist "/etc/amanda/daily/tapelist" # Lista de volumes dispoñibles.
tpchanger "chg-disk:/var/backups/amanda/daily/vtapes" # Cambiador de discos virtuais para as copias.
labelstr "^Etiqueta[0-9]+$" # Formato de etiquetas válidas (e.g., "Etiqueta1", "Etiqueta2").
tapetype DISCO # Tipo de volume configurado como disco duro.

define tapetype DISCO {
    comment "Copias de seguridade sobre disco duro" # Descrición do tipo de volume.
    length 8 gbytes # Tamaño máximo de cada volume virtual (8 GB).
}

define dumptype disco-global {
    comment "Definición de parámetros globais" # Configuración compartida por todos os tipos de copia.
    holdingdisk never # Non se usan discos temporais.
    index yes # Crear índices das copias realizadas.
    compress client fast # Compresión rápida feita no cliente.
}

define dumptype disco-tar {
    comment "Copia de seguridade en disco, usando tar" # Copia empregando o programa tar.
    disco-global # Herda as propiedades globais.
    program "GNUTAR" # Emprega GNUTAR como ferramenta de copia.
}

define dumptype completo {
    comment "Volcado total" # Copias completas de todos os datos.
    disco-tar # Herda as propiedades de disco-tar.
    dumpcycle 0 # Non ten ciclo; sempre realiza unha copia completa.
    compress none # Sen compresión.
    strategy noinc # Estratexia: Non fai copias incrementais.
}

define dumptype diferencial {
    comment "Volcado diferencial" # Só os datos cambiados desde a última copia completa.
    disco-tar # Herda as propiedades de disco-tar.
    compress client fast # Compresión rápida.
    dumpcycle 2 # Realiza copias diferenciais cada 2 días dende a última completa.
}

define dumptype incremental {
    comment "Volcado incremental" # Só os datos novos ou modificados desde a última copia.
    disco-tar # Herda as propiedades de disco-tar.
    compress client fast # Compresión rápida.
    dumpcycle 1 # Realiza copias incrementais dos últimos cambios diarios.
}' > /etc/amanda/daily/amanda.conf

```

#### Código1. Ficheiro amanda.conf

## 2. Comprobación e Execución do Backup

### 2.1. Verificación da Configuración con amcheck

Antes de realizar o backup, verifica que a configuración é correcta:

```
# su backup -s /bin/bash -c "/usr/sbin/amcheck daily"
```

A saída debería ser algo como:

```
Amanda Tape Server Host Check
-----
slot 1: volume 'Etiqueta1'
Will write to volume 'Etiqueta1' in slot 1.
NOTE: skipping tape-writable test
NOTE: host info dir '/etc/amanda/daily/curinfo/localhost' does not exist
      It will be created on the next run
NOTE: index dir '/etc/amanda/daily/index/localhost' does not exist
      it will be created on the next run
Server check took 0.115 seconds
Amanda Backup Client Hosts Check
-----
Client check: 1 host checked in 2.097 seconds. 0 problems found.
(brought to you by Amanda 3.5.1)
```

### 2.2. Executar o Backup con amdump

Poderás realizar un backup manual en calquera momento usando o comando amdump:

```
# su backup -s /bin/bash -c "/usr/sbin/amdump daily"
```

Este comando executará o backup para **/home** (completo), **/etc** (diferencial) e **/var/log** (incremental).

Imos facer cambios e facer de novo un backup manual:

```
# echo 'F facendo cambios en home' > /home/usuario/cambios-home-1.txt
# chown usuario:usuario /home/usuario/cambios-home-1.txt
# echo 'F facendo cambios en etc' > /etc/cambios-etc-1.txt
# echo 'F facendo cambios en var_log' > /var/log/cambios-var_log-1.txt
# su backup -s /bin/bash -c "/usr/sbin/amdump daily"
```

Imos facer cambios de novo e facer outro backup manual:

```
# echo 'F facendo de novo cambios en home' > /home/usuario/cambios-home-2.txt
# chown usuario:usuario /home/usuario/cambios-home-2.txt
# echo 'F facendo de novo cambios en etc' > /etc/cambios-etc-2.txt
# echo 'F facendo de novo cambios en var_log' > /var/log/cambios-var_log-2.txt
# su backup -s /bin/bash -c "/usr/sbin/amdump daily"
```

## 3. Comprobación Despois do Backup

### 3.1. Verificar os Rexistros de Execución

Revisa os rexistros de execución para asegurarte de que non houbo erros durante o backup:

```
# less /etc/amanda/daily/log/amdump.*
...
FAILED QUEUE: empty
DONE QUEUE:
  0: localhost /var/log
  1: localhost /etc
  2: localhost /home
...
```

### 3.2. Verificar o Estado dos Volumes e Slots

Comproba os volumes nos slots e se foron usados correctamente:

```
# su backup -s /bin/bash -c "amtape daily show"
scanning all 14 slots in changer:
slot  1: date 20241203121350 label Etiqueta1
slot  2: date 20241203121438 label Etiqueta2
slot  3: date 20241203121524 label Etiqueta3
slot  4: date X                label Etiqueta4
slot  5: date X                label Etiqueta5
slot  6: date X                label Etiqueta6
slot  7: date X                label Etiqueta7
slot  8: date X                label Etiqueta8
slot  9: date X                label Etiqueta9
slot 10: date X                label Etiqueta10
slot 11: date X                label Etiqueta11
slot 12: date X                label Etiqueta12
slot 13: date X                label Etiqueta13
slot 14: date X                label Etiqueta14
```

### 3.3. Comprobar a Creación dos Directorios

Comproba que os directorios e arquivos de backup se crearon correctamente:

```
# ls -R /etc/amanda/daily/curinfo
# ls -R /etc/amanda/daily/index
# ls -R /var/backups/amanda/daily/vtapes | less
```



## 4. Programación do Backup con Cron

### 4.1. Configuración de Cron para Realizar o Backup Diario

Para automatizar a execución do backup, engadiremos unha tarefa programada co cron que execute o backup diariamente ás 3:00h.

Para configurar cron, edita o crontab do usuario backup con:

```
# crontab -u backup -e
```

Engade a seguinte liña no ficheiro crontab para executar amdump ás 3:00h todos os días:

```
0 3 * * * /usr/sbin/amdump daily
```

Así, unha vez que gardes e peches o crontab, o backup de Amanda será executado automaticamente ás 3:00h cada día.

## 5. Restauración de Backups con Amanda

### 5.1. amrestore vs amrecover

#### Características de amrestore

- Ferramenta de baixo nivel que recupera datos directamente desde as etiquetas do backup.
- Non permite explorar ou seleccionar ficheiros específicos.
- Útil para situacións de emerxencia ou cando non tes os índices de Amanda.
- **Limitacións:** É funcional con cintas físicas ou virtuais, pero **cando os backups están almacenados en discos (usando chg-disk), amrestore non funciona directamente debido a que espera un dispositivo de cinta ou ficheiro de cinta. Isto significa que non pode procesar ficheiros como os de slot1, slot2, etc..** Para discos, a extracción manual co comando dd ou o uso de amrecover son as solucións máis adecuadas.

#### Características de amrecover

- Ferramenta interactiva de alto nivel que utiliza os índices xerados por Amanda.
- Permite explorar backups e seleccionar ficheiros ou directorios específicos.
- Requere acceso á configuración completa e aos índices de Amanda.

#### Comparación

Característica	amrestore	amrecover
Interfaz	Baseada en comandos directos	Interactiva e exploratoria
Selección parcial	Non, só restaura o tarball completo	Si, permite seleccionar ficheiros
Dependencias	Acceso á etiqueta do backup	Config. completa e índices de Amanda
Facilidade de uso	Complexo	Intuitivo
Soporte para cintas	Si	Si
Soporte para discos (chg-disk)	Non	Si

#### Exemplo de uso para backups en disco: Restauración con dd

Cando os backups están almacenados en discos e non se pode usar amrestore, dd é unha ferramenta alternativa que permite extraer os ficheiros TAR de backups gardados en ficheiros AMANDA.

Supoñamos que o ficheiro /var/backups/amanda/daily/vtapes/slot1/00003.localhost.\_home.0 contén o backup de /home:

```
# dd if=/var/backups/amanda/daily/vtapes/slot1/00003.localhost._home.0 bs=32k skip=1 of=/tmp/home.tar
# tar xvf /tmp/home.tar -C /tmp/restauracion
```

O comando dd elimina o encabezado AMANDA (mediante skip=1) para crear un ficheiro TAR que pode ser extraído con tar.

### 5.2. Comprobar a Configuración dos Volumes

Antes de restaurar, asegúrate de que os volumes virtuais (vtapes) están dispoñibles e correctamente configurados:

```
# su backup -s /bin/bash -c "amtape daily show"
```

Este comando amosará os slots dispoñibles e os volumes asignados. Verifica que o volume que precisas está presente.

### 5.3. Preparar o Ambiente para a Restauración

Crea un directorio temporal onde se restaurarán os ficheiros:

```
# mkdir -p /tmp/restauracion
# chown backup:backup /tmp/restauracion
```

Soamente farase a restauración no directorio orixe da copia cando se comprobe que restauración no directorio temporal foi correcta.

### 5.4. Restaurar o Backup Completo de /home

Para restaurar utiliza o comando amrecover.

#### 1. Inicia amrecover:

```
# amrecover daily
AMRECOVER Version 3.5.1. Contacting server on localhost ...
NAK: user root from localhost is not allowed to execute the service amindexd:
Please add the line "localhost root amindexd amidxtaped" to /var/backups/.amandahosts on the server
```

A execución anterior amosa un erro que debemos solucionar. Este erro indica que o usuario root no host localhost non está autorizado a executar o servizo amindexd (parte do software Amanda, usado para xestión de copias de seguridade) debido a unha configuración incorrecta no ficheiro .amandahosts do servidor. A mensaxe indica que debes engadir unha liña específica ao ficheiro /var/backups/.amandahosts no servidor para permitir que o usuario root acceda ao servizo.

```
# ls -l /var/backups/.amandahosts
lrwxrwxrwx 1 root root 16 nov 29 15:54 /var/backups/.amandahosts -> /etc/amandahosts
# ls -l /etc/amandahosts
-rw----- 1 backup backup 17 dic 3 2023 /etc/amandahosts
# cat /etc/amandahosts
localhost backup
# echo 'localhost root amindexd amidxtaped' >> /etc/amandahosts
# amrecover daily
AMRECOVER Version 3.5.1. Contacting server on localhost ...
220 computer AMANDA index server (3.5.1) ready.
Setting restore date to today (2024-12-03)
200 Working date set to 2024-12-03.
200 Config set to daily.
501 Host computer is not in your disklist.
Trying host computer ...
501 Host computer is not in your disklist.
Trying host computer ...
501 Host computer is not in your disklist.
Use the sethost command to choose a host to recover
amrecover>
```

#### 2. Prepara e configura o directorio de restauración:

```
amrecover> listhost
amrecover> sethost localhost
amrecover> listdisk
amrecover> setdisk /home
amrecover> lcd /tmp/restauracion
```

### 3. Selecciona e acepta a restauración:

Confirma cando se che solicite. Os ficheiros restauraranse no directorio especificado por lcd.

```
amrecover> add *
amrecover> list
TAPE daily:Etiqueta3:3 LEVEL 0 DATE 2024-12-03-12-15-24
/usuario
```

Como se pode observar por defecto estase a coller o último backup completo (LEVEL 0 DATE 2024-12-03-12-15-24). Se quixeramos escoller o backup completo a restaurar debemos quitar o agregado e indicar a data do backup a restaurar:

```
# su - backup -s /bin/bash -c "amtape daily show"
scanning all 14 slots in changer:
slot  1: date 20241203121350 label Etiqueta1
slot  2: date 20241203121438 label Etiqueta2
slot  3: date 20241203121524 label Etiqueta3
slot  4: date X                label Etiqueta4
slot  5: date X                label Etiqueta5
slot  6: date X                label Etiqueta6
slot  7: date X                label Etiqueta7
slot  8: date X                label Etiqueta8
slot  9: date X                label Etiqueta9
slot 10: date X                label Etiqueta10
slot 11: date X                label Etiqueta11
slot 12: date X                label Etiqueta12
slot 13: date X                label Etiqueta13
slot 14: date X                label Etiqueta14
amrecover> delete /home/*
amrevover> setdate 2024-12-03-12-13-50
200 Working date set to 2024-12-03-12-13-50.
amrecover> add *
Added dir /usuario/ at date 2024-12-03-12-13-50
amrecover> list
TAPE daily:Etiqueta1:3 LEVEL 0 DATE 2024-12-03-12-13-50
/usuario
amrecover> extract
amrecover> quit
# ls -l /tmp/restauracion/usuario/cambios-home*
ls: no se puede acceder a '/tmp/restauracion/usuario/cambios-home-*':
No existe el fichero o el directorio
```

```
amrecover> extract
amrecover> quit
# ls -l /tmp/restauracion/usuario/cambios-home-*
-rw-r--r-- 1 usuario usuario 24 dic  3 12:14 /tmp/restauracion/usuario/cambios-home-1.txt
-rw-r--r-- 1 usuario usuario 32 dic  3 12:14 /tmp/restauracion/usuario/cambios-home-2.txt
```

### 5.5. Restaurar o Backup Diferencial de /etc

Os backups diferenciais dependen da última copia completa realizada. Usa amrecover para xestionar a restauración.

#### 1. Inicia amrecover:

```
# amrecover daily
AMRECOVER Version 3.5.1. Contacting server on localhost ...
220 computer AMANDA index server (3.5.1) ready.
Setting restore date to today (2024-12-03)
200 Working date set to 2024-12-03.
200 Config set to daily.
501 Host computer is not in your disklist.
Trying host computer ...
501 Host computer is not in your disklist.
Trying host computer ...
501 Host computer is not in your disklist.
Use the sethost command to choose a host to recover
amrecover>
```

#### 2. Prepara e configura o directorio de restauración:

```
amrecover> listhost
amrecover> sethost localhost
amrecover> listdisk
amrecover> setdisk /etc
amrecover> lcd /tmp/restauracion
```

#### 3. Selecciona e acepta a restauración:

Confirma cando se che solicite. Os ficheiros restauraranse no directorio especificado por lcd.

```
amrecover> add *
amrecover> list
TAPE daily:Etiqueta1:1 LEVEL 0 DATE 2024-12-03-12-13-50
  /.pwd.lock
  /adduser.conf
  /adjtime
  ...
TAPE daily:Etiqueta3:2 LEVEL 1 DATE 2024-12-03-12-15-24
  /ImageMagick-6
  /ModemManager
  /NetworkManager
  ...
```

Como se pode observar para restaurar o backup diferencial estase a partir dun backup completo (LEVEL 0 DATE 2024-12-03-12-13-50) e logo os arquivos que cambiaron dende ese backup completo (LEVEL 1 DATE 2024-12-03-12-15-24):

```
# su - backup -s /bin/bash -c "amtape daily show"
scanning all 14 slots in changer:
slot   1: date 20241203121350 label Etiqueta1
slot   2: date 20241203121438 label Etiqueta2
slot   3: date 20241203121524 label Etiqueta3
slot   4: date X                label Etiqueta4
...
slot  14: date X                label Etiqueta14
```

```
amrecover> extract
amrecover> quit
# ls -l /tmp/restauracion/cambios-etc-*
-rw-r--r-- 1 root root 23 dic  3 12:14 /tmp/restauracion/cambios-etc-1.txt
-rw-r--r-- 1 root root 31 dic  3 12:15 /tmp/restauracion/cambios-etc-2.txt
```

## 5.6 Restaurar o Backup Incremental de /var/log

O proceso para os backups incrementais é similar aos diferenciais.

1. Inicia amrecover:

```
# amrecover daily
```

2. Configura o directorio de restauración:

```
amrecover> listhost
amrecover> sethost localhost
amrecover> listdisk
amrecover> setdisk /var/log
amrecover> lcd /tmp/restauracion
```

3. Lista e selecciona os backups incrementais máis recentes:

```
amrecover> add *
amrecover> list
TAPE daily:Etiqueta1:2 LEVEL 0 DATE 2024-12-03-12-13-50
  /README
  /Xorg.0.log
  /Xorg.0.log.old
  ...
TAPE daily:Etiqueta3:1 LEVEL 1 DATE 2024-12-03-12-15-24
  /amanda
  /apt
  /cambios-var_log-1.txt
  /cambios-var_log-2.txt
  ...
amrecover> extract
amrecover> quit
# ls -l /tmp/restauracion/cambios-etc-*
-rw-r--r-- 1 root root 23 dic  3 12:14 /tmp/restauracion/cambios-etc-1.txt
-rw-r--r-- 1 root root 31 dic  3 12:15 /tmp/restauracion/cambios-etc-2.txt
```

### Por que Amanda usa só 2 etiquetas en lugar de 3 para restaurar /var/log?

A razón de que amrecover só use **2 etiquetas** (tapes) en lugar de **3**, mesmo cando os backups de /var/log son incrementais, está relacionada co funcionamento de Amanda na xestión de copias incrementais.

#### Explicación técnica

1. **Backup Completo (LEVEL 0):**

- O backup inicial, localizado en Etiqueta1, é un backup completo que contén todos os arquivos de /var/log no momento da súa execución.

2. **Primeiro Backup Incremental (LEVEL 1):**

- O segundo backup, localizado en Etiqueta3, inclúe só os cambios realizados desde o último backup completo (Etiqueta1). Neste caso, inclúe cambios-var\_log-1.txt.

3. **Segundo Backup Incremental (LEVEL 1):**

- O terceiro backup tamén se basea no último backup completo (Etiqueta1), pero ao incorporar cambios máis recentes, o backup incremental de Etiqueta3 tamén inclúe cambios-var\_log-2.txt.

4. **Por que non inclúe Etiqueta2?**

- O slot Etiqueta2 non contén datos de /var/log. Está asociado a outro recurso ou directorio especificado na disklist (por exemplo, /etc).
- Amanda só solicita os volumes que conteñen datos relevantes para /var/log.

#### Confirmación a través da saída de amrecover

- **TAPE daily:Etiqueta1:2 LEVEL 0:**

Este é o backup completo inicial.

- **TAPE daily:Etiqueta3:1 LEVEL 1:**

Este é o backup incremental que inclúe todos os cambios desde o backup completo en Etiqueta1.

Amanda optimiza o proceso de restauración e non solicita backups incrementais adicionais se os cambios necesarios están cubertos por un backup máis recente (como ocorre con Etiqueta3).

### Solución e verificación

#### 1. Verificar os contidos de cada tape:

Usa o comando `amadmin daily find` para inspeccionar os backups realizados e confirmar que só Etiqueta1 e Etiqueta3 inclúen datos de `/var/log`.

#### 2. Verificar os arquivos restaurados:

Despois de realizar a extracción, confirma que os arquivos restaurados inclúen:

- Os arquivos orixinais de `/var/log` (desde o backup completo).
- Os cambios `cambios-var_log-1.txt` e `cambios-var_log-2.txt` (desde o backup incremental).

### Conclusión

Non é necesario que Amanda use os **3 backups** porque os datos requiridos para restaurar `/var/log` están completamente cubertos polos backups en Etiqueta1 (completo) e Etiqueta3 (incremental). O slot Etiqueta2 non é relevante para `/var/log`. Isto indica que o backup incremental está funcionando correctamente e Amanda está optimizando o proceso.

## 6. Notas Importantes

- **Integridade dos datos:** Os backups incrementais e diferenciais requiren a última copia completa para restaurarse correctamente.
- **Ampliar espazo:** Se o directorio de restauración ten pouco espazo, selecciona outro destino con máis capacidade.
- **Seguridade:** Mover os datos restaurados aos directorios orixinais só despois de verificar que son correctos.



## Exemplo2: Backups en Rede con Amanda

**Prerrequisito: Realizar previamente o Exemplo1: Backups en localhost con Amanda**

### 1. Configuración no Servidor (192.168.120.100)

#### 1.1. Configurar o ficheiro disklist

Edita o ficheiro /etc/amanda/daily/disklist para engadir o cliente remoto:

```
# Exemplo do ficheiro disklist
localhost /home completo
localhost /etc diferencial
localhost /var/log incremental
cliente1 /home completo
cliente1 /etc diferencial
cliente1 /var/log incremental
```

#### 1.2. Verificar a Conectividade entre o Servidor e o Cliente

Proba a conectividade entre o servidor e o cliente:

```
# Probar a conexión
$ ping -c2 192.168.120.101
```

#### 1.3. Configurar o DNS ou o Ficheiro Hosts

Se non tes un servidor DNS, engade a entrada no ficheiro /etc/hosts:

```
192.168.120.101 cliente1
```

#### 1.4. Configurar os Permisos para que o usuario root poida realizar a recuperación dos backups do cliente

Engade ao cliente o usuario root en /etc/amandahosts e establece os permisos correctos:

```
# echo '192.168.120.100 root amindexd amidxtaped' >> /etc/amandahosts
# chown backup:backup /etc/amandahosts
# chmod 600 /etc/amandahosts
```

### 2. Configuración no Cliente (192.168.120.101)

#### 2.1. Instalar Amanda no Cliente

Actualizar os paquetes e instalar Amanda:

```
# apt update
# apt -y install amanda-client
```

#### 2.2. Configurar o Ficheiro amanda-client.conf

Crea ou edita o ficheiro /etc/amanda/amanda-client.conf para que conteña a seguinte configuración:

```
# Exemplo de configuración do cliente
conf "daily"          # Nome da configuración no servidor
index_server "192.168.120.100" # IP do servidor de Amanda
tape_server "192.168.120.100" # IP do servidor de Amanda
auth "bsdtcp"         # Método de autenticación
```

```
# mkdir -p /etc/amanda
# echo 'conf "daily"          # Nome da configuración no servidor
index_server "192.168.120.100" # IP do servidor de Amanda
tape_server "192.168.120.100" # IP do servidor de Amanda
auth "bsdtcp"         # Método de autenticación' >> /etc/amanda/amanda.conf
# chown -R backup:backup /etc/amanda
```

## 2.3. Configurar os Permisos para que o usuario root poida realizar os backups do cliente mediante o comando amdump

Engade ao cliente o usuario root en /etc/amandahosts e establece os permisos correctos:

```
# echo '192.168.120.100 backup amdump' >> /etc/amandahosts
# chown backup:backup /etc/amandahosts
# chmod 600 /etc/amandahosts
```

## 3. Probar a Configuración de Amanda

### 3.1. Comprobar a Configuración do Cliente desde o Servidor

Executa o comando amcheck no servidor:

```
# su backup -s /bin/bash -c "/usr/sbin/amcheck daily"
Amanda Tape Server Host Check
-----
slot 4: volume 'Etiqueta4'
Will write to volume 'Etiqueta4' in slot 4.
NOTE: skipping tape-writable test
NOTE: host info dir '/etc/amanda/daily/curinfo/cliente1' does not exist
      It will be created on the next run
NOTE: index dir '/etc/amanda/daily/index/cliente1' does not exist
      it will be created on the next run
Server check took 0.111 seconds
Amanda Backup Client Hosts Check
-----
Client check: 2 hosts checked in 1.150 seconds.  0 problems found.
(brought to you by Amanda 3.5.1)
```

## 4. Executar os Backups

### 4.1. Backup Manual

Para realizar un backup manual:

```
# su backup -s /bin/bash -c "/usr/sbin/amdump daily"
```

Imos facer 2 veces máis o comando anterior para poder ter 3 copias de backup

```
# su backup -s /bin/bash -c "/usr/sbin/amdump daily"
# su backup -s /bin/bash -c "/usr/sbin/amdump daily"
```

### 4.2. Comprobación Despois do Backup

A partir de agora procede, de forma análoga, como se describe na sección de backup en localhosts a partir do apartado 3. Comprobación Despois do Backup.

### Resumo

- **Backup completo:** Fai unha copia de seguridade completa dos datos especificados.
- **Backup diferencial:** Fai unha copia de seguridade dos arquivos que cambiaron desde o último backup completo.
- **Backup incremental:** Realiza un backup dos arquivos que cambiaran desde o último backup completo ou incremental.

Amanda permite configurar os backups segundo a necesidade, realizando copias completas, diferenciais ou incrementais e xestionándoas desde un servidor centralizado para múltiples clientes en redes.