

Respaldos

El respaldo de los sistemas y de los datos que se incluyen en ellos es una de las responsabilidades del administrador del sistema. Los procedimientos que se tienen que implementar y la elección de las herramientas dependen de numerosos factores, que pueden ser a menudo complejos.

En el marco de la certificación LPIC2, vamos a estudiar diferentes comandos y herramientas de Linux que permiten efectuar respaldos y restauraciones, totales o parciales, y también vamos a evocar algunas aplicaciones especializadas en este ámbito.

1. ¿Qué hay que respaldar ?

Los respaldos deben incluir diferentes elementos: los datos de los usuarios, los datos de las aplicaciones (archivos y bases de datos), los archivos ejecutables y de configuración de las aplicaciones instaladas y los archivos ejecutables y de configuración del sistema operativo. Estos distintos elementos no necesitan las mismas estrategias de respaldo (frecuencia, tipo de respaldo y soportes que se deberán utilizar) ni las mismas herramientas de respaldo.

a. Sistemas de archivos

La mayoría de los sistemas Linux estructuran su arborescencia global de directorios y de archivos asociando diferentes sistemas de archivos. Una de las ventajas de esta organización es permitir respaldos autónomos de los diferentes sistemas de archivos. Podemos, prácticamente, desmontarlos todos y respaldarlos individualmente. Si se encuentran en volúmenes lógicos LVM o en sistemas de archivos que lo permiten, se puede usar la técnica de instantáneas (snapshots) para salvaguardar su estado en el momento del respaldo.

Sin embargo, algunos directorios necesarios para el arranque del sistema no pueden estar situados en los sistemas de archivos que no se encuentren en la raíz. Este es el caso particular del directorio `/etc`, que contiene archivos de configuraciones esenciales, y el directorio `/bin` que contiene ejecutables indispensables para inicializar el sistema (por ejemplo, el programa shell

`/bin/sh`). Por lo tanto, no podemos desmontar el sistema de archivos que los contiene para hacer un respaldo.

b. Directorios

Los diferentes directorios en la raíz del sistema tienen que ser o no respaldados, y los que lo necesiten pueden serlo con una frecuencia diferente. He aquí una lista de los principales directorios y sus características de respaldo típicas:

<code>/home</code>	Directorios de los usuarios, respaldo regular.
<code>/opt</code>	Directorio de las aplicaciones y de sus datos, respaldo regular.
<code>/var</code>	Archivos de registro, spools, archivos de cola, etc., respaldo regular (excepto <code>/var/tmp</code>).
<code>/etc</code>	Directorio de archivos de configuración, respaldo regular.
<code>/usr</code>	Directorio de ejecutables del sistema y de las aplicaciones, respaldo puntual.
<code>/bin</code>	Directorio de ejecutables y de las aplicaciones, respaldo puntual.
<code>/sbin</code>	Directorio de ejecutables, respaldo puntual.
<code>/lib</code>	Directorio de bibliotecas del sistema y de las aplicaciones, respaldo puntual.
<code>/lib64</code>	Directorio de bibliotecas del sistema y de las aplicaciones, respaldo puntual.
<code>/boot</code>	Directorio del núcleo, respaldo puntual.
<code>/dev</code>	Directorio de archivos especiales, no es necesario respaldarlo.

`/media` Directorio de montaje de periféricos, no es necesario respaldarlo.

`/mnt` Directorio de montaje temporal, no es necesario respaldarlo.

`/net` Directorio de montaje temporal, no es necesario respaldarlo.

`/srv` Directorio de montaje temporal, no es necesario respaldarlo.

`/proc` Directorio de montaje de sistema de archivos virtual, no es necesario respaldarlo.

`/run` Directorio de montaje del sistema de archivos virtual, no es necesario respaldarlo.

`/sys` Directorio de montaje del sistema de archivos virtual, no es necesario respaldarlo.

`/tmp` Directorio de archivos temporales, no es necesario respaldarlo.



Estos consejos tienen carácter indicativo, pero pueden ajustarse en función del tipo de uso del sistema (puesto de trabajo o servidor; servidor de aplicaciones, de redes, de bases de datos, de almacenamiento de datos de los usuarios, etc.).

2. ¿En qué soporte habría que hacer los respaldos?

Existen diferentes tipos de soportes de respaldo, cada uno presenta ventajas e

inconvenientes De nuevo, debe hacerse un estudio para determinar qué tipo de soportes habría que utilizar para qué tipo de respaldo.

a. Cinta magnética

La cinta magnética es un soporte barato que puede almacenar volúmenes importantes de datos. Su vida útil estimada es de tipo medio y demandan un riguroso mantenimiento.

Se trata de soportes lentos de lectura y escritura que necesitan un material específico relativamente caro.

Cada lector de cintas está asociado a dos archivos especiales: `/dev/stX` y `/dev/nstX`, donde `X` es un número de índice que empieza por 0. El primero es un archivo especial de acceso al lector en modo de rebobinado automático, mientras que el segundo no efectúa rebobinado al final de la escritura (`n` de *no rewind*), lo que permite almacenar distintos archivos comprimidos en la cinta.

Para facilitar el uso del lector de cintas, un archivo `/dev/tape` se asocia a menudo al lector por defecto del sistema `/dev/stX`.

El comando `mt` (*magnetic tape*) permite administrar lectores de cintas.

Sintaxis

```
mt [ -f ArchivoEspecial ] Cmd [ Núm ]
```

Parámetros principales

<code>-f Archivoespecial</code>	Archivo especial asociado al lector de cintas.
<code>Cmd</code>	Comando que se ejecutará.
<code>Núm</code>	Número usando como argumento del comando.

Descripción

El comando permite gestionar un lector de cintas. Sin la opción `-f`, usará el lector por defecto, `/dev/tape`.

Las opciones más habituales son:

`status`

Muestra la información sobre el estado del cartucho.

`fsf`

Avanza la posición actual con el `Núm` de archivos (1 por defecto).

`bsf`

Retrocede la posición actual con el `Núm` de archivos (1 por defecto).

`rewind`

Rebobina la cinta.

`erase`

Borra la cinta.

b. Disco duro

También se podría efectuar el respaldo en un archivo comprimido almacenado en un disco duro. Este disco puede ser extraíble o no.

Esta solución tiene muchas ventajas: velocidad de lectura y escritura, volumen importante

de datos y soporte asequible.

La gestión del almacenamiento puede ser compleja si hay que conservar durante mucho tiempo los archivos de respaldo.

c. Disco óptico

Se pueden usar CD/DVD grabables para efectuar respaldos. La velocidad del respaldo será media, así como la de restauración. La vida útil de los CD/DVD es buena y su almacenamiento fácil. Los inconvenientes son el volumen de almacenamiento limitado y no reutilizable.

d. Redes

El respaldo a través de la red puede efectuarse hacia sistemas remotos o redes de almacenamiento. Esta solución presenta muchas ventajas, la velocidad del respaldo y de restauración puede ser buena según el tipo de red (Ethernet banda ancha y fibra óptica), el espacio de almacenamiento puede situarse en un sitio geográficamente remoto y se pueden mutualizar equipos caros.

3. Estrategias de respaldo

Es necesario implementar una estrategia de respaldo para salvaguardar el sistema y los datos con regularidad. Esto se hará en función de los objetivos de seguridad y sostenibilidad de la información respaldada, del volumen de respaldos y de la velocidad de respaldo y de restauración.

Combinamos generalmente diferentes tipos de respaldos para implementar una estrategia periódica de respaldo.

- Respaldo completo: se respalda un conjunto de elementos (archivos, directorios, sistemas de archivos).
- Respaldo parcial (incremental o diferencial): se respaldan únicamente los elementos modificados desde el último respaldo parcial (modo incremental) o desde el último respaldo completo (modo diferencial).

Para restaurar todos los datos o una parte de ellos, utilizamos el último respaldo completo, opcionalmente seguido del último respaldo diferencial, o de respaldos incrementales sucesivos desde el último respaldo completo.

Para implementar estas estrategias, podemos definir distintos niveles de respaldo, generalmente un número de índice de 0 a n. Los comandos estándares de Linux, `tar` y `cpio`, saben gestionar este tipo de respaldo.

Ejemplos

Estrategia de respaldo incremental en una semana:

domingo	Respaldo completo, nivel 0.
lunes	Respaldo incremental, nivel 1.
martes	Respaldo incremental, nivel 2.
miércoles	Respaldo incremental, nivel 3.
jueves	Respaldo incremental, nivel 4.
viernes	Respaldo incremental, nivel 5.
sábado	Respaldo incremental, nivel 6.

Se guardará cada día lo que ha sido modificado desde el respaldo del día anterior. En el caso de que hubiera algún problema, habría que restaurar el respaldo del domingo anterior y, sucesivamente, cada respaldo diario hasta el último respaldo efectuado.

Estrategia de respaldo diferencial en una semana:

domingo	Respaldo completo, nivel 0.
lunes	Respaldo diferencial, nivel 1.
martes	Respaldo diferencial, nivel 1.
miércoles	Respaldo diferencial, nivel 1.
jueves	Respaldo diferencial, nivel 1.
viernes	Respaldo diferencial, nivel 1.
sábado	Respaldo diferencial, nivel 1.

Se guardará cada día lo que ha sido modificado desde el respaldo completo del domingo. En el caso de que haya algún problema, habría que restaurar el respaldo del domingo anterior, y después el último respaldo efectuado.

4. Los comandos de respaldo y de restauración

Linux dispone de distintos comandos estándares de respaldo y de restauración. Permiten efectuar respaldos completos o parciales y trabajan en el nivel de la arborescencia de directorios y archivos.

Estos comandos existen en todas las distribuciones, con el mismo formato de almacenaje. Es por esto por lo que podemos restaurar una distribución completa o una parte de ella desde un respaldo efectuado en otra distribución.



Estos comandos existen en todos los sistemas de tipo Unix y, en principio, son compatibles entre sistemas de tipos diferentes.

a. El comando tar

El comando `tar` es un estándar de Linux para el intercambio de archivos comprimidos entre sistemas. Permite efectuar el respaldo y la restauración, completa o parcial, de un conjunto de archivos y/o directorios.

Sintaxis

```
tar [ Opciones ] [-f ArchivoComprimido [ Dir Archivos ]
```

Parámetros principales

Opciones	Opciones (con o sin guion).
<code>[-f ArchivoComprimido]</code>	Archivo comprimido.
<code>Dir Archivos</code>	Directorio(s) y/o archivo(s) para comprimir o restaurar.

Descripción

El comando permite gestionar el respaldo o la restauración de directorios (con toda la arborescencia incluida) y de archivos. Por defecto, el comando almacena en un archivo el camino de acceso a los objetos sin el carácter / inicial. La restauración, por defecto, se hace en el directorio actual.

El nombre del archivo comprimido es libre, pero se aconseja ponerle una extensión para poder reconocer su origen (`.tar`) y el modo de compresión si fuera necesario (`.gz`).

.bz2, etc.).

Las opciones pueden estar especificadas con o sin guion, excepto las opciones de tipo GNU que tienen que llevar delante un doble guion.

Hay muchas opciones distintas, he aquí las más usadas:

t|xc

Tipo de operación que se efectuará: **t** (*table of contents*) para listar el contenido del archivo comprimido, **x** (*eXtract*) para restaurar todo el archivo una parte de él, **c** (*create*) para crear el archivo comprimido.

z|Z|j|J|

Tipo de compresión: **z** para **gzip**, **Z** para **compress**, **j** para **bzip2**, **J** para **xz**.

v

Visualización detallada.

k

Restauración sin machacar los archivos existentes.

--keep-newer-files

Restauración sin machacar los archivos existentes más recientes que los del archivo comprimido.

-g ArchivoInfo

Modo incremental. El archivo `ArchivoInfo` contiene la información sobre el último respaldo (efectuado con la opción `-g ArchivoInfo`).

Ejemplos

Respaldo y restauración de un directorio que contiene los archivos de código fuente de la versión 1.7.10 de `nginx`, en el directorio actual del usuario:

ls -l nginx-1.17.10

```
total 836
drwxr-xr-x 6 pba pba 4096 abril 15 16:58 auto
-rw-r--r-- 1 pba pba 302754 abril 14 16:19 CHANGES
-rw-r--r-- 1 pba pba 462076 abril 14 16:19 CHANGES.ru
drwxr-xr-x 2 pba pba 4096 abril 15 14:38 conf
-rwxr-xr-x 1 pba pba 2502 abril 14 16:19 configure
drwxr-xr-x 4 pba pba 4096 abril 15 14:38 contrib
drwxr-xr-x 2 pba pba 4096 abril 15 14:38 html
-rw-r--r-- 1 pba pba 1397 abril 14 16:19 LICENSE
-rw-r--r-- 1 root root 376 abril 15 17:02 Makefile
drwxr-xr-x 2 pba pba 4096 abril 15 14:38 man
-rw-r--r-- 1 pba pba 45043 abril 15 16:52 nginx.patch
drwxr-xr-x 3 root root 4096 abril 15 17:03 objs
-rw-r--r-- 1 pba pba 49 abril 14 16:19 README
drwxr-xr-x 9 pba pba 4096 abril 15 14:38 src
```

Se crea un archivo comprimido con formato `xz`, en el directorio actual:

tar cvJf nginx-1.17.10.tar.xz nginx-1.17.10

```
nginx-1.17.10/
nginx-1.17.10/CHANGES.ru
nginx-1.17.10/contrib/
nginx-1.17.10/contrib/README
nginx-1.17.10/contrib/vim/
nginx-1.17.10/contrib/vim/syntax/
[...]
nginx-1.17.10/objs/src/os/unix/nginx_setaffinity.o
nginx-1.17.10/objs/src/misc/
nginx-1.17.10/objs/src/mail/
```

```

nginx-1.17.10/objs/nginx.8
nginx-1.17.10/objs/nginx_auto_config.h
nginx-1.17.10/objs/nginx
nginx-1.17.10/objs/Makefile
nginx-1.17.10/Makefile

```

Se hace un listado del contenido del archivo:

```

tar tvJf nginx-1.17.10.tar.xz nginx-1.17.10
drwxr-xr-x pba/pba      0 2020-04-15 18:55 nginx-1.17.10/
-rw-r--r-- pba/pba    462076 2020-04-14 16:19 nginx-1.17.10/CHANGES.ru
drwxr-xr-x pba/pba      0 2020-04-15 14:38 nginx-1.17.10/contrib/
-rw-r--r-- pba/pba     543 2020-04-14 16:19 nginx-1.17.10/contrib/README
drwxr-xr-x pba/pba      0 2020-04-14 16:19 nginx-1.17.10/contrib/vim/
drwxr-xr-x pba/pba      0 2020-04-14 16:19 nginx-1.17.10/contrib/vim/syntax/
-rw-r--r-- pba/pba   130074 2020-04-14 16:19 nginx-1.17.10/contrib/vim/syntax/
nginx.vim
[...]
nginx-1.17.10/objs/src/os/unix/nginx_setaffinity.o
nginx-1.17.10/objs/src/misc/
nginx-1.17.10/objs/src/mail/
nginx-1.17.10/objs/nginx.8
nginx-1.17.10/objs/nginx_auto_config.h
nginx-1.17.10/objs/nginx
nginx-1.17.10/objs/Makefile
nginx-1.17.10/Makefile

```

Se puede observar que el archivo ha guardado los caminos de acceso en modo relativo, sin la / inicial. Es por esto por lo que la restauración tendrá lugar, por defecto, en el directorio actual.

Con la cuenta de usuario `root` y en el directorio de conexión de este usuario, restauramos el archivo:

```

tar xJvf ~pba/nginx-1.17.10.tar.xz
nginx-1.17.10/

```

```

nginx-1.17.10/CHANGES.ru
nginx-1.17.10/contrib/
nginx-1.17.10/contrib/README
nginx-1.17.10/contrib/vim/
nginx-1.17.10/contrib/vim/syntax/
nginx-1.17.10/contrib/vim/syntax/nginx.vim
nginx-1.17.10/contrib/vim/ftdetect/
nginx-1.17.10/contrib/vim/ftdetect/nginx.vim
nginx-1.17.10/contrib/vim/ftplugin/
nginx-1.17.10/contrib/vim/ftplugin/nginx.vim
nginx-1.17.10/contrib/vim/indent/
nginx-1.17.10/contrib/vim/indent/nginx.vim
[...]
nginx-1.17.10/objs/nginx.8
nginx-1.17.10/objs/nginx_auto_config.h
nginx-1.17.10/objs/nginx
nginx-1.17.10/objs/Makefile
nginx-1.17.10/Makefile

```

Se hace un listado del nuevo directorio:

```

ls -l nginx-1.17.10
total 836
drwxr-xr-x 6 pba pba 4096 abril 15 16:58 auto
-rw-r--r-- 1 pba pba 302754 abril 14 16:19 CHANGES
-rw-r--r-- 1 pba pba 462076 abril 14 16:19 CHANGES.ru
drwxr-xr-x 2 pba pba 4096 abril 15 14:38 conf
-rwxr-xr-x 1 pba pba 2502 abril 14 16:19 configure
drwxr-xr-x 4 pba pba 4096 abril 15 14:38 contrib
drwxr-xr-x 2 pba pba 4096 abril 15 14:38 html
-rw-r--r-- 1 pba pba 1397 abril 14 16:19 LICENSE
-rw-r--r-- 1 root root 376 abril 15 17:02 Makefile
drwxr-xr-x 2 pba pba 4096 abril 15 14:38 man
-rw-r--r-- 1 pba pba 45043 abril 15 16:52 nginx.patch
drwxr-xr-x 3 root root 4096 abril 15 17:03 objs
-rw-r--r-- 1 pba pba 49 abril 14 16:19 README
drwxr-xr-x 9 pba pba 4096 abril 15 14:38 src

```

Observamos que los objetos restaurados han conservado sus atributos (propietario, grupo y

fechas).

Respaldo y restauración diferencial:

Efectuamos un respaldo completo del directorio de la aplicación `nginx`, guardándolo como si fuera de nivel 0 en un archivo de información:

```
tar cvJf nginx-0.tar.xz -g ./nginx-lasttar /usr/local/nginx/
```

```
tar: /usr/local/nginx: El directorio es nuevo
tar: /usr/local/nginx/conf: El directorio es nuevo
tar: /usr/local/nginx/html: El directorio es nuevo
tar: /usr/local/nginx/logs: El directorio es nuevo
tar: /usr/local/nginx/sbin: El directorio es nuevo
tar: Eliminando la «/» inicial de los nombres
/usr/local/nginx/
/usr/local/nginx/conf/
/usr/local/nginx/html/
/usr/local/nginx/logs/
/usr/local/nginx/sbin/
/usr/local/nginx/conf/fastcgi.conf
/usr/local/nginx/conf/fastcgi.conf.default
/usr/local/nginx/conf/fastcgi_params
/usr/local/nginx/conf/fastcgi_params.default
/usr/local/nginx/conf/koi-utf
/usr/local/nginx/conf/koi-win
/usr/local/nginx/conf/mime.types
/usr/local/nginx/conf/mime.types.default
/usr/local/nginx/conf/nginx.conf
/usr/local/nginx/conf/nginx.conf.default
/usr/local/nginx/conf/scgi_params
/usr/local/nginx/conf/scgi_params.default
/usr/local/nginx/conf/uwsgi_params
/usr/local/nginx/conf/uwsgi_params.default
/usr/local/nginx/conf/win-utf
/usr/local/nginx/html/50x.html
/usr/local/nginx/html/index.html
/usr/local/nginx/sbin/nginx
/usr/local/nginx/sbin/nginx-nopatch
/usr/local/nginx/sbin/nginx.old
```

Se hace un listado de los dos archivos que resultan del comando:

```
ls -l ./nginx-lasttar
-rw-r--r-- 1 root root 647 abril 16 14:56 ./nginx-lasttar
ls -l nginx-0.tar.xz
-rw-r--r-- 1 root root 1093980 abril 16 14:56 nginx-0.tar.xz
```

Modificamos el contenido del archivo de configuración de `nginx`:

```
vi /usr/local/nginx/conf/nginx.conf
#####
##### MODIFICACIÓN EFECTUADA DESPUÉS DEL RESPALDO DE NIVEL 0 #####
#####
#user nobody;
worker_processes 1;

#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;
[...]
```

Efectuamos un respaldo diferencial:

```
tar cvJf nginx-1.tar.xz -g ./nginx-lasttar /usr/local/nginx/
tar: Eliminando la «/» inicial de los nombres
/usr/local/nginx/
/usr/local/nginx/conf/
/usr/local/nginx/html/
/usr/local/nginx/logs/
/usr/local/nginx/sbin/
/usr/local/nginx/conf/nginx.conf
```

Se hace un listado de los dos archivos que resultan del comando:

```
ls -l nginx-lasttar nginx-1.tar.xz
-rw-r--r-- 1 root root 1416 abril 16 15:01 nginx-1.tar.xz
```

```
-rw-r--r-- 1 root root 645 abril 16 15:01 nginx-lasttar
```

El archivo comprimido es pequeñito, el archivo de información se ha actualizado.

Se hace un listado del contenido del archivo comprimido:

```
tar tvJf nginx-1.tar.xz
drwxr-xr-x root/root    25 2020-04-15 16:06 usr/local/nginx/
drwxr-xr-x root/root   238 2020-04-16 15:01 usr/local/nginx/conf/
drwxr-xr-x root/root    23 2020-04-15 16:06 usr/local/nginx/html/
drwxr-xr-x root/root     1 2020-04-15 16:06 usr/local/nginx/logs/
drwxr-xr-x root/root    34 2020-04-15 17:03 usr/local/nginx/sbin/
-rw-r--r-- root/root  2868 2020-04-16 15:01 usr/local/nginx/conf/nginx.conf
```

Solamente contiene el archivo modificado `usr/local/nginx/conf/nginx.conf`.

Queremos efectuar una restauración completa del directorio guardado en otro directorio. Para ello, primero habrá que restaurar el nivel `0`:

```
mkdir /opt/nginxbis
cd /opt/nginxbis
tar xvJf ~root/nginx-0.tar.xz
usr/local/nginx/
usr/local/nginx/conf/
usr/local/nginx/html/
usr/local/nginx/logs/
usr/local/nginx/sbin/
[...]
usr/local/nginx/conf/fastcgi.conf
usr/local/nginx/conf/fastcgi.conf.default
usr/local/nginx/conf/fastcgi_params
usr/local/nginx/conf/fastcgi_params.default
usr/local/nginx/conf/koi-utf
usr/local/nginx/conf/koi-win
usr/local/nginx/conf/mime.types
usr/local/nginx/conf/mime.types.default
```



```
usr/local/nginx/conf/nginx.conf
usr/local/nginx/conf/nginx.conf.default
usr/local/nginx/conf/scgi_params
usr/local/nginx/conf/scgi_params.default
usr/local/nginx/conf/uwsgi_params
usr/local/nginx/conf/uwsgi_params.default
usr/local/nginx/conf/win-utf
usr/local/nginx/html/50x.html
usr/local/nginx/html/index.html
usr/local/nginx/sbin/nginx
usr/local/nginx/sbin/nginx-nopatch
usr/local/nginx/sbin/nginx.old
```

Aplicamos después la restauración del archivo comprimido diferencial:

```
tar xvJf ~root/nginx-1.tar.xz
usr/local/nginx/
usr/local/nginx/conf/
usr/local/nginx/html/
usr/local/nginx/logs/
usr/local/nginx/sbin/
usr/local/nginx/conf/nginx.conf
```

Mostramos el contenido del archivo nginx.conf restaurado:

```
vi usr/local/nginx/conf/nginx.conf
#####
##### MODIFICACIÓN EFECTUADA DESPUÉS DEL RESPALDO DE NIVEL 0 #####
#####
#user nobody;
worker_processes 1;
[...]
```

La restauración ha funcionado correctamente.

b. El comando cpio

`cpio` es otro comando, también de origen Unix, que permite hacer respaldos y restauraciones. Se usa menos frecuentemente que `tar`, porque sus diferentes formatos de archivo comprimido no siempre son compatibles entre diferentes implementaciones.

Sintaxis

```
cpio [-Opciones] [ListaArchivos] [<ListaArchivos|ArchivoComprimido]
[>ArchivoComprimido]
```

Parámetros principales

Opciones	Opciones.
ListaArchivos	Lista de los directorios(s) y/o archivo(s) que se van a restaurar.
> ArchivoComprimido	Archivo comprimido que se va a crear.
< ListaArchivos ArchivoComprimido	Lista de los directorio(s) y/o archivo(s) que se van a comprimir, o archivo comprimido que se va a restaurar.

Descripción

El comando permite realizar el respaldo o la restauración de directorios (incluyendo toda la arborescencia) y de archivos.

El nombre del archivo comprimido es libre, pero para facilitar su uso, se acostumbra a ponerle una extensión para precisar su origen (`.cpio`). El comando no soporta los formatos de compresión estándares.

Por defecto, el comando utiliza la salida estándar para generar el archivo comprimido, y su entrada estándar para leer la lista de los archivos y/o directorios que se tendrán que

respaldar. Se usa generalmente el comando `find` para generar la lista de los objetos que se tienen que respaldar, lista que será enviada al comando `cpio` usando un pipe, por ejemplo:

```
find . | cpio -ov > archivo.cpio
```

Las opciones son muy numerosas, he aquí las más usadas:

`t`ljo

Tipo de operación que se efectuará: `t` (*table of contents*) para listar el contenido del archivo comprimido, `i` (*input*) para restaurar el archivo comprimido completo o una parte de él, `o` (*output*) para crear un archivo comprimido.

`v`

Visualización detallada.

Ejemplo

Respaldo y restauración de un directorio que contiene los archivos de código fuente de la versión 1.7.10 de `nginx`, en el directorio actual del usuario:

```
ls -l nginx-1.17.10
total 836
drwxr-xr-x 6 pba pba 4096 abril 15 16:58 auto
-rw-r--r-- 1 pba pba 302754 abril 14 16:19 CHANGES
-rw-r--r-- 1 pba pba 462076 abril 14 16:19 CHANGES.ru
drwxr-xr-x 2 pba pba 4096 abril 15 14:38 conf
-rwxr-xr-x 1 pba pba 2502 abril 14 16:19 configure
drwxr-xr-x 4 pba pba 4096 abril 15 14:38 contrib
drwxr-xr-x 2 pba pba 4096 abril 15 14:38 html
-rw-r--r-- 1 pba pba 1397 abril 14 16:19 LICENSE
-rw-r--r-- 1 root root 376 abril 15 17:02 Makefile
drwxr-xr-x 2 pba pba 4096 abril 15 14:38 man
```

```
-rw-r--r-- 1 pba pba 45043 abril 15 16:52 nginx.patch
drwxr-xr-x 3 root root 4096 abril 15 17:03 objs
-rw-r--r-- 1 pba pba 49 abril 14 16:19 README
drwxr-xr-x 9 pba pba 4096 abril 15 14:38 src
```

Se crea un archivo comprimido en el directorio actual:

```
find nginx-1.17.10 | cpio -ov > nginx-1.17.10.cpio
nginx-1.17.10
nginx-1.17.10/CHANGES.ru
nginx-1.17.10/contrib
nginx-1.17.10/contrib/README
nginx-1.17.10/contrib/vim
nginx-1.17.10/contrib/vim/syntax
nginx-1.17.10/contrib/vim/syntax/nginx.vim
[...]
nginx-1.17.10/objs/nginx.8
nginx-1.17.10/objs/nginx_auto_config.h
nginx-1.17.10/objs/nginx
nginx-1.17.10/objs/Makefile
nginx-1.17.10/Makefile
43507 blocs
```

Comparamos el tamaño del archivo `cpio` con el tamaño de formato `tar.xz` del ejemplo anterior:

```
ls -l nginx-1.17.10.cpio nginx-1.17.10.tar.xz
-rw-r--r-- 1 pba pba 22275584 abril 16 15:49 nginx-1.17.10.cpio
-rw-r--r-- 1 pba pba 2681396 abril 16 14:42 nginx-1.17.10.tar.xz
```

El archivo comprimido `tar.xz` es mucho menos voluminoso.

Listamos el contenido del archivo comprimido:

```
cpio -tv < nginx-1.17.10.cpio
```

```

drwxr-xr-x  9 pba  pba      0 Apr 15 18:55 nginx-1.17.10
-rw-r--r--  1 pba  pba    462076 Apr 14 16:19 nginx-1.17.10/CHANGES.ru
drwxr-xr-x  4 pba  pba      0 Apr 15 14:38 nginx-1.17.10/contrib
-rw-r--r--  1 pba  pba    543 Apr 14 16:19 nginx-1.17.10/contrib/README
drwxr-xr-x  6 pba  pba      0 Apr 14 16:19 nginx-1.17.10/contrib/vim
drwxr-xr-x  2 pba  pba      0 Apr 14 16:19 nginx-1.17.10/contrib/vim/syntax
-rw-r--r--  1 pba  pba   130074 Apr 14 16:19 nginx-1.17.10/contrib/vim/syntax/
nginx.vim
[...]
-rw-r--r--  1 root  root    6841 Apr 15 17:02 nginx-1.17.10/objs/nginx_auto_config.h
-rwxr-xr-x  1 root  root  4835696 Apr 15 17:03 nginx-1.17.10/objs/nginx
-rw-r--r--  1 root  root   39124 Apr 15 17:02 nginx-1.17.10/objs/Makefile
-rw-r--r--  1 pba  pba     376 Apr 15 17:02 nginx-1.17.10/Makefile

```

Podemos observar que el archivo comprimido ha guardado los caminos de acceso en modo relativo, sin la / inicial. La restauración tendrá lugar, por defecto, en el directorio actual.

Usando la cuenta de usuario `root` y en su directorio de conexión, restauramos el archivo comprimido:

```

cpio -iv < ~pba/nginx-1.17.10.cpio
[...]
nginx-1.17.10/objs/src/mail
nginx-1.17.10/objs/nginx.8
nginx-1.17.10/objs/nginx_auto_config.h
nginx-1.17.10/objs/nginx
nginx-1.17.10/objs/Makefile
nginx-1.17.10/Makefile
43507 bloques

```

Hacemos un nuevo listado del directorio:

```

ls -l nginx-1.17.10
total 836
drwxr-xr-x 6 pba pba  4096 abril 16 15:52 auto
-rw-r--r-- 1 pba pba 302754 abril 16 15:52 CHANGES

```

```

-rw-r--r-- 1 pba pba 462076 abril 16 15:52 CHANGES.ru
drwxr-xr-x 2 pba pba 4096 abril 16 15:52 conf
-rwxr-xr-x 1 pba pba 2502 abril 16 15:52 configure
drwxr-xr-x 4 pba pba 4096 abril 16 15:52 contrib
drwxr-xr-x 2 pba pba 4096 abril 16 15:52 html
-rw-r--r-- 1 pba pba 1397 abril 16 15:52 LICENSE
-rw-r--r-- 1 pba pba 376 abril 16 15:52 Makefile
drwxr-xr-x 2 pba pba 4096 abril 16 15:52 man
-rw-r--r-- 1 pba pba 45043 abril 16 15:52 nginx.patch
drwxr-xr-x 3 root root 4096 abril 16 15:52 objs
-rw-r--r-- 1 pba pba 49 abril 16 15:52 README
drwxr-xr-x 9 pba pba 4096 abril 16 15:52 src

```

Se puede observar que los objetos restaurados han conservado sus atributos (propietario, grupo y fechas).

5. Los softwares de respaldo

Para aplicar una estrategia periódica de respaldo de uno o varios sistemas, la mayoría de las veces, se implementará una aplicación dedicada para ello. En el entorno Linux, existen diferentes soluciones de tipo open source.

Para la certificación LPIC-2 tendrá que conocer los softwares más utilizados.

a. Bacula y Bareos

Bacula es una solución completa de respaldo, bajo licencia open source, que permite respaldar localmente o en red, en discos o en cintas, los datos de los sistemas Linux, Unix y Windows. Es el más utilizado de los softwares libres de gestión de respaldos. Ha sido desarrollado principalmente por Kern Sibbald.

Bacula usa su propio formato de compresión.

Bareos es el resultado de un fork del software Bacula, en 2010. Existe una versión libre y otra comercial.

b. AMANDA

AMANDA (*Advanced Maryland Automatic Network Disk Archiver*) es una aplicación potente de respaldo, creada inicialmente por la Universidad de Maryland bajo licencia BSD. Está disponible en licencia abierta (gratis) o comercial. AMANDA realiza respaldos locales o en red, en discos o en cintas, de los sistemas Linux, Unix y Windows.

AMANDA usa el formato de compresión estándar (TAR).

c. BackupPC

BackupPC es una solución de respaldo en red, bajo licencia GPL, que permite respaldar en discos o en cintas los datos de los sistemas Linux, Unix o Windows. No precisa de un software de tipo cliente, el servidor proporciona una interfaz web.

d. Softwares comerciales

La mayoría de los softwares comerciales de respaldo en red (ArcServe, Backup Exec, EMC NetWorker, IBM Tivoli Storage Manager y NetBackup) funcionan con Linux y permiten integrar los respaldos Linux con los de otros tipos de sistemas.

6. Duplicación de disco: el comando dd

El comando `dd` (*disk duplication*) permite realizar copias de bajo nivel, completas o parciales, de un espacio de almacenamiento, disco, cinta, partición o volumen lógico.

Este comando multiuso puede ser empleado para respaldar un espacio de almacenamiento duplicándolo.

Sintaxis

```
dd if=ArchivoOrig of=ArchivoDest [bs=TamañoBloque] [count=NúmBloques]
```

Parámetros principales

<code>if=ArchivoOrig</code>	Archivo especial del espacio de almacenamiento que se va a duplicar.
<code>of=ArchivoDest</code>	Archivo destino de la duplicación.
<code>[bs=TamañoBloque]</code>	Tamaño de los bloques.
<code>[count=NºmBloques]</code>	Número de bloques que se copiarán.
<code>[skip=NºmBloques]</code>	Número de bloques que se saltarán al principio del archivo de entrada.
<code>[seek=NºmBloques]</code>	Número de bloques que se saltarán al principio del archivo de salida.

Descripción

El comando copia el archivo de entrada `ArchivoOrig` en el archivo de salida `ArchivoDest`. El parámetro `bs=` permite indicar el tamaño de bloques del archivo de entrada, el parámetro `count=` el número de bloques que se copiarán, el parámetro `skip=` el número de bloques del archivo de entrada que se saltarán antes de empezar la copia, `seek=` el número de bloques del archivo de salida que se saltarán antes de empezar la copia.

Los archivos de entrada y de salida pueden ser archivos especiales asociados a periféricos o archivos normales del sistema de archivos global. También se puede copiar el sistema de archivos de un disco, de una partición o de un volumen lógico, en un archivo almacenado en un disco, este último puede ser montado en la arborescencia como sistema de archivos de origen.

Ejemplos

Copia del primer sector de un disco duro (MBR) hacia un archivo.


```
dd if=/dev/sda of=mbr.sv bs=512 count=1
1+0 registros leídos
1+0 registros escritos
512 bytes copied, 0,00225934 s, 227 kB/s
ls -l mbr.sv
-rw-r--r-- 1 root root 512 abril 16 16:51 mbr.sv
```

Creación de un archivo vacío (lleno por bytes a 0) de 100 MB:

```
dd if=/dev/zero of=archivo10M bs=1024 count=10000
10000+0 registros leídos
10000+0 registros escritos
10240000 bytes (10 MB, 9,8 MiB) copied, 0,0330657 s, 310 MB/s
ls -l archivo10M
-rw-r--r-- 1 root root 10240000 abril 16 16:54 archivo10M
```

Copia de un disco duro `/dev/sdb` en un archivo:

```
dd if=/dev/sdb of=sdb.sv
30322688+0 registros leídos
30322688+0 registros escritos
15525216256 bytes (16 GB, 14 GiB) copied, 1072,59 s, 14,5 MB/s
ls -l sdb.sv
-rw-r--r-- 1 root root 15525216256 16 abril 15:25 sdb.sv
```

Montamos el archivo para comprobar su contenido:

```
mount sdb.sv /mnt
ls -l /mnt
total 24
drwxr-xr-x. 2 root root 4096 16 abril 15:06 bin
drwxr-xr-x. 2 root root 4096 16 abril 15:06 datas
drwx----- 2 root root 16384 16 abril 14:58 lost+found
```

El sistema de archivos se ha montado correctamente y es accesible en lectura y escritura:

```
echo "Archivo en un archivo" > /mnt/archivo
```

```
ls -l /mnt
```

```
total 28
```

```
drwxr-xr-x. 2 root root 4096 16 abril 15:06 bin
```

```
drwxr-xr-x. 2 root root 4096 16 abril 15:06 datas
```

```
-rw-r--r--. 1 root root 24 16 abril 16:06 archivo
```

```
drwx-----. 2 root root 16384 16 abril 14:58 lost+found
```

7. Sincronización de datos: el comando rsync

El comando `rsync` permite sincronizar arborescencias de directorios y archivos entre ellas. Se utiliza a menudo como herramienta de respaldo de datos a través de una red, porque solo se transfieren las diferencias entre la arborescencia de origen y la de destino, lo que limita el tráfico de red y optimiza el rendimiento del respaldo.

Este comando puede funcionar en dos modos distintos:

- Modo servidor `rsync`: el servidor `rsync` debe correr y estar en escucha en un puerto TCP. En el cliente, el comando `rsync` se conecta al servidor especificado gracias al número de puerto usado por el servicio y ejecuta la sincronización de datos solicitada.
- Modo `ssh`: el comando `rsync` se conecta al servidor `ssh` especificado y ejecuta la sincronización de datos solicitada. Es el modo más seguro y por lo tanto el más usado.

a. Sintaxis

El comando permite sincronizar el sistema remoto hacia el sistema local o del sistema local hacia el remoto.

```
rsync [Opciones] -e ssh [[Usuario@]Host:]Origen [[Usuario@]Host:]Destino
```

Parámetros principales

Opciones	Opciones.
-e ssh	Modo ssh.
[Usuario@]Host	Cuenta de usuario y servidor remoto.
Origen	Arborescencia de referencia.
Destino	Arborescencia que será sincronizada.

Descripción

El comando sincroniza los elementos entre `Destino` y `Origen`. Por defecto, se usa la cuenta de usuario local, excepto si se especifica otra cuenta con el argumento `Usuario@`. En modo `ssh`, el comando `rsync` debe existir en el sistema remoto.

Las opciones más usadas son las siguientes:

-v

Visualización detallada.

-a

Modo archivo, los derechos y las fechas se preservarán.

-z

Los datos serán comprimidos para el envío.

b. Ejemplo

Respaldo usando `rsync` de un directorio incluido en el directorio de conexión del usuario `pba`, de un sistema Debian 10 hacia un sistema CentOS 8.

En el sistema Debian 10:

```
ls -l dir_actual
total 8
drwxr-xr-x 2 pba pba 4096 abril 16 18:32 network
drwxr-xr-x 9 pba pba 4096 abril 16 18:32 nginx-1.17.10
rsync -avze ssh ./dir_actual centos8:
pba@centos8's password:XXXX
sending incremental file list
./
.bash_history
.bashrc
.profile
.cache/
.config/
.config/procps/
.gnupg/
.gnupg/private-keys-v1.d/
.ssh/
.ssh/known_hosts
datas/
[...]
dir_actual/nginx-1.17.10/src/stream/nginx_stream_variables.h
dir_actual/nginx-1.17.10/src/stream/nginx_stream_write_filter_module.c

sent 6,445,025 bytes received 10,694 bytes 993,187.54 bytes/sec
total size is 22,297,716 speedup is 3.45
```

En el sistema de destino, el directorio `dir_actual` ha sido creado dentro del directorio de conexión del usuario `pba`:

```
ls -ld /home/pba/dir_actual
drwxr-xr-x. 4 pba pba 42 16 abril 2020 /home/pba/dir_actual
```

```
ls -l /home/pba/dir_actual
total 0
drwxr-xr-x. 2 pba pba 141 16 abril 2020 network
drwxr-xr-x. 9 pba pba 205 16 abril 2020 nginx-1.17.10
```

La copia ha sido efectuada.

En el sistema Debian 10, añadimos un archivo en el directorio `dir_actual`:

```
echo "Nuevo" > dir_actual/archivo
```

Sincronizamos la copia del servidor CentOS 8:

```
rsync -ae ssh ./dir_actual centos8:
pba@centos8's password:XXXX
```

Mostramos el contenido del directorio sincronizado en el servidor CentOS 8:

```
ls -l /home/pba/dir_actual
total 4
-rw-r--r--. 1 pba pba 8 16 abril 2020 archivo
drwxr-xr-x. 2 pba pba 141 16 abril 2020 network
drwxr-xr-x. 9 pba pba 205 16 abril 2020 nginx-1.17.10
cat /home/pba/dir_actual/archivo
Nuevo
```

La sincronización se ha efectuado correctamente.