1 Copias de seguridad

La realización de copias de seguridad es uno de los aspectos fundamentales de la administración de sistemas. Siempre existe la posibilidad de que ocurra un desastre o una pérdida accidental de información (como por ejemplo un comando rm mal escrito), por tanto, independientemente de los niveles de seguridad hardware (como RAID) deben establecerse políticas de copia de seguridad periódica de los datos. Existen múltiples herramientas de gestión de copias de seguridad sobre Linux, algunas privativas y otras libres. A continuación veremos algunas herramientas estándar, disponibles en cualquier Linux a través de la consola de terminal que nos permitirán realizar esta tarea.

1.1 Comando tar

Este comando es una utilidad que permite empaquetar o archivar ficheros en un único archivo con extensión .tar. Es por tanto una herramienta que nos permite manejar varios archivos como una única unidad, muy útil por ejemplo para mover archivos por la red o para hacer copias de seguridad del contenido de directorios. Su sintaxis es:

tar [opciones] archivos

Opciones principales:

- -f: Especifica el fichero que se va a utilizar como archivador
- -c: Indica que se creará el fichero archivador tar
- -C: Especifica directorio de destino en el que se extraerán los contenidos del tar
- -v: Verboso, muestra información por pantalla de la ejecución del comando
- -x: Se utiliza para extraer los ficheros del archivador
- -t: Se utiliza para listar el contenido de un archivador tar
- -p: Conserva los permisos de los ficheros
- -z: Soporte para compresión mediante gzip
- -j: Soporte para compresión mediante bz2

Es importante a la hora de archivar los ficheros determinar si se usarán rutas absolutas o relativos. Veámoslo con un ejemplo:

```
tar -cvf archivo.tar .
```

Archiva todos los archivos del directorio actual en el fichero archivo.tar, almacenando directamente en él los archivos sin añadir rutas absolutas

```
tar -cvf archivo.tar /home/usuario
```

Archiva todos los archivos del directorio /home/usuario en archivo.tar, manteniendo la información de ruta absoluta.

La diferencia entre mantener rutas absolutas o no es que a la hora de extraer se crearán los directorios correspondientes en el sistema de archivos destino

Tras crear un archivador tar suele ser buena idea listar sus contenidos para comprobar que se ha realizado correctamente el comando:

```
tar -tvf archivo.tar
```

Mostraría el listado de las rutas de los archivos en el archivador .tar

Ejemplo de uso de compresión:

```
tar -cvzf archivo.tar.gz /home/usuario
```

Notad que si usamos la opción -f para especificar el fichero archivador, éste debe de ser la última opción pues requiere a continuación el parámetro que indica el nombre del archivo. Si queremos utilizar el algoritmo de compresión **bz2 en lugar de zip, usaríamos la opción -j en lugar de -z**:

```
tar -cvjf archivo.tar.bz2 /home/usuario
```

Ejemplo de extracción:

```
tar -xvf archivo.tar
```

Extrae el contenido del archivador archivo.tar en el directorio actual, en el que previamente debemos habernos posicionado con cd.

Para extraer el contenido de un tar en un directorio especificado utilizamos la opción **-C** (el directorio debe existir). Ejemplo:

```
tar -xvf archivo.tar -C /home/usuario/salidatar
```

Ejemplo de extracción y descompresión:

```
tar -xzvf archivo.tar.gz
```

Descomprime y extrae el contenido del archivador archivo.tar.gz en el directorio /home/usuario. Sino se especifica el destino lo extraerá en el directorio actual. Si el archivo estuviese comprimido en .bz2 sustituiríamos la opción -z por -j

```
tar -xjvf archivo.tar.bz2
```

1.2 Comando rsync

rsync es una utilidad muy flexible y potente para realizar transferencias y sincronización de archivos en local y remoto. Tiene la propiedad de que, una vez que se ha realizado una transferencia con rsync, la próxima vez que hagamos otra transferencia relacionada con archivos ya transferidos sólo se sincronizarán los cambios, en caso de que existan respecto a la primera versión transferida. La sintaxis básica es la siguiente:

rsync [opciones] fuente [destino]

dónde el destino puede ser local o un sistema remoto. Para acceso remoto se puede utilizar SSH o bien el propio daemon de rsync. Sino se especifica el destino entonces se listan los archivos en lugar de copiarlos. Como siempre, para conocer exactamente las opciones y la sintaxis podéis utilizar el comando man para acceder a las páginas del manual correspondientes al comando:

man rsync

Veamos algunos ejemplos de uso:

```
rsync -a /home/usuario/dir1/ /home/usuario/dir2
```

La opción -a indica que los archivos son transferidos en "archive mode" lo cual indica que los links y permisos se conservan recursivamente en el destino. Por tanto -a será una opción que usaremos siempre. Es importante que se ponga la última "/" en el directorio origen. Sino se pone se creará un directorio dir1 dentro del destino: /home/usuario/dir2/dir1

Otras opciones:

- -v: modo "verbose", muestra información sobre lo que hace el comando por pantalla
- -z: comprime los datos durante la transferencia de modo transparente
- --delete: Esta opción conserva los borrados. Por defecto si después de realizar una primera transferencia rsync se elimina un archivo del origen, éste sigue existiendo en el destino. Con esta opción forzamos a que los borrados también se sincronizen

Listado de opciones más comunes:

- -a: Archiva. Es una buena configuración. Es sinonimo que especificar ?-DGgloprt?
- -b: Hace una copia de los ficheros, aún estando en el destino. Normalmente no es útil utilizar esta opción, a no ser que se quiera tener copia de cada una de las versiones del fichero.
- -e comando_shell: Usa la shell para crear un la conexión con el sistema remoto. Útil para especificar conexiones ssh, con -e ssh.
- -g: Preserva el grupo de los ficheros que van a ser replicados. Importante para los backups.
- -H: Preserva los ?hard-links?. Esta opción relentiza el copiado, pero es muy recomendable.
- -l: Copia los ?symlinks? como ?symlinks?. Sin esta opción marcada, un link simbólico sería respaldado como un fichero.
- -n: Dry run (Ejecución Seca): Lista los ficheros que van a ser transferidos, pero no hace el respaldo.
- -o: Preserva el usuario posesor de los ficheros que van a ser replicados. Importante.
- -p: Preserva los permisos de los ficheros. Importante.
- -r: Activa la recursividad. Para incluir todos los subdirectorios de una carpeta.
- --rsh='ssh': Utiliza SSH par la transmisión de datos. Es recomendable ya que sinó usa el protocolo inseguro rsh. Se usa cuando el servidor remoto tiene instalado SSH, evidentemente.
- -t: Preserva las fechas de modificación de cada fichero.
- -v: Hace una de los ficheros que son transferidos.
- -vv: Igual que -v, pero además muestra los ficheros que no van a ser copiados.
- -vvv: Igual que -vv, pero también imprime información de debug de rsync

- -z: Activa la compresión. Importante si hacemos respaldos a través de Internet.

Copiado remoto con ssh:

```
rsync -avz -e ssh /home/usuario/ usuario@servidor.com:/home/usuario
```

Copia archivos locales de /home/usuario a una ubicación remota utilizando ssh en el servidor remoto "servidor.com" con el usuario "usuario". Será necesario que esté instalado un servidor ssh en el servidor remoto, como openssh.

```
rsync -avz -e ssh usuario@servidor.com:/home/usuario /home/usuario
```

Es la inversa de la anterior, se toma como fuente el sistema remoto y se copia en el sistema local

También se puede hacer copias remotas utilizando el propio daemon rsync, sin necesidad de utilizar ssh.

1.3 Referencias

Rsync y SSH

http://troy.jdmz.net/rsync/index.html

Volver

JavierFP 16:31 08 ene 2019 (CET)