
Ejercicios Paso a Paso: LPIC-2 Objetivo 206.2 - Realización de Copias de Seguridad

Escenario: Tienes un servidor `ubuntu` (Ubuntu Server) con el usuario `curso` con capacidad `sudo`. Necesitarás acceso a internet para descargar el código fuente si lo requieres, aunque para estos ejercicios se trabajará con archivos locales. Si no tienes una unidad de cinta real, los ejercicios con `mt` serán conceptuales.

Ejercicio 1: Creación y Verificación de Copias de Seguridad con `tar`

Objetivo: Practicar la creación de copias de seguridad de directorios, incluyendo la exclusión de archivos y la verificación básica del contenido del archivo de backup.

Parte A: Preparación del Entorno

1. Conéctate a tu servidor `ubuntu`:

```
ssh curso@ubuntu
```

2. Crea una estructura de directorios y archivos de prueba:

```
mkdir -p ~/mis_datos/documentos ~/mis_datos/configuracion
echo "Mi informe secreto." > ~/mis_datos/documentos/informe.txt
echo "Contraseña123" > ~/mis_datos/configuracion/credenciales.conf
echo "Log de hoy." > ~/mis_datos/log.txt
mkdir -p ~/backups_locales
```

3. Crea un archivo y un directorio para excluirlos del backup:

```
echo "Archivo temporal, no respaldar." > ~/mis_datos/temp.tmp
mkdir ~/mis_datos/cache
echo "Archivos de cache inútiles." > ~/mis_datos/cache/junk.log
```

Parte B: Creación del Backup Completo

1. Realiza una copia de seguridad comprimida del directorio `~/mis_datos`:

```
tar -czvf ~/backups_locales/mis_datos_backup_$(date +%Y%m%d).tar.gz ~/mis_datos
```

- `tar -c`: Crea el archivo.
- `-z`: Comprime con `gzip`.
- `-v`: Muestra una lista de los archivos mientras se añaden (verbose).

- `-f`: Especifica el nombre del archivo de salida.
- `$(date +%Y%m%d)`: Inserta la fecha actual en el nombre del archivo (ej., `mis_datos_backup_20250628.tar.gz`).

2. Verifica el tamaño del archivo de backup creado:

```
ls -lh ~/backups_locales/
```

3. Lista el contenido del archivo de backup para verificar qué se incluyó:

```
tar -tf ~/backups_locales/mis_datos_backup_$(date +%Y%m%d).tar.gz
```

- Salida Esperada: Deberías ver todos los archivos y directorios de `~/mis_datos`, incluyendo `temp.tmp` y `cache/`.

Parte C: Creación del Backup con Exclusiones

1. Realiza una nueva copia de seguridad, excluyendo los archivos `.tmp` y el directorio `cache`:

```
tar -czvf ~/backups_locales/mis_datos_excl_backup_$(date +%Y%m%d).tar.gz \
    --exclude='*.tmp' \
    --exclude='mis_datos/cache' \
    ~/mis_datos
```

- `--exclude`: Utilizado para especificar patrones o rutas a excluir. La ruta para el directorio `cache` (`mis_datos/cache`) es relativa al punto de partida del backup (`~/mis_datos`).

2. Lista el contenido del nuevo archivo de backup para confirmar las exclusiones:

```
tar -tf ~/backups_locales/mis_datos_excl_backup_$(date +%Y%m%d).tar.gz
```

- Salida Esperada: Ni `mis_datos/temp.tmp` ni `mis_datos/cache/junk.log` deberían aparecer en la lista.

Ejercicio 2: Restauración de Datos con `tar` y Verificación de Integridad

Objetivo: Practicar la restauración total y parcial de datos, y verificar la integridad de un archivo de backup usando `sha256sum`.

Parte A: Preparación para la Restauración

1. Crea un directorio para restaurar los datos (no uses el original para evitar sobrescribir):

```
mkdir -p ~/restauracion_prueba
```

2. Genera una suma de verificación para tu backup inicial (el que no tiene exclusiones):

```
cd ~/backups_locales/
sha256sum mis_datos_backup_$(date +%Y%m%d).tar.gz > mis_datos_backup_$(date +%Y
%m%d).tar.gz.sha256
```

- Esto creará un pequeño archivo de texto con el hash SHA256 y el nombre del archivo de backup.

3. Visualiza la suma de verificación generada:

```
cat mis_datos_backup_$(date +%Y%m%d).tar.gz.sha256
```

Parte B: Simulación de Corrupción y Verificación

1. Simula la corrupción del archivo de backup (¡hazlo solo con copias de backup, NUNCA con archivos originales!):

```
echo "Datos corruptos añadidos" >> mis_datos_backup_$(date +%Y%m%d).tar.gz
```

2. Verifica la integridad del archivo de backup nuevamente:

```
sha256sum -c mis_datos_backup_$(date +%Y%m%d).tar.gz.sha256
```

- Salida Esperada: Deberías ver un mensaje de `FAILED` junto al nombre del archivo, indicando que el archivo ha sido modificado y su integridad comprometida.

Parte C: Restauración Parcial

1. Elimina el backup corrupto y su suma de verificación. Luego, restaura el backup original (sin la corrupción simulada) para poder trabajar con un archivo íntegro:

```
rm mis_datos_backup_$(date +%Y%m%d).tar.gz mis_datos_backup_$(date +%Y%m
%d).tar.gz.sha256
# Vuelve a generar el backup si eliminaste el original, o usa uno limpio si ya
tenías uno.
# Por simplicidad, volvamos al directorio mis_datos y creamos uno nuevo limpio
cd ~
rm -rf ~/mis_datos_temp_corrupt
mv ~/mis_datos ~/mis_datos_temp_corrupt # Mueve el original temporalmente
mkdir -p ~/mis_datos/documentos ~/mis_datos/configuracion
echo "Mi informe secreto." > ~/mis_datos/documentos/informe.txt
echo "Contraseña123" > ~/mis_datos/configuracion/credenciales.conf
echo "Log de hoy." > ~/mis_datos/log.txt
cd ~/backups_locales/
tar -czvf mis_datos_backup_$(date +%Y%m%d).tar.gz ~/mis_datos # Crea un backup
nuevo y limpio
```

2. Restaura solo el archivo `informe.txt` del backup original al directorio `~/restauracion_prueba`:

```
tar -xzvf mis_datos_backup_$(date +%Y%m%d).tar.gz \
    -C ~/restauracion_prueba \
    mis_datos/documentos/informe.txt
```

- `-C`: Cambia al directorio especificado antes de extraer los archivos.
- `mis_datos/documentos/informe.txt`: Esta es la ruta del archivo dentro del tarball.

3. Verifica que solo ese archivo se ha restaurado en la ruta completa:

```
ls -R ~/restauracion_prueba/
cat ~/restauracion_prueba/mis_datos/documentos/informe.txt
```

Parte D: Restauración Total

1. Limpia el directorio de restauración de prueba:

```
rm -rf ~/restauracion_prueba/*
```

2. Restaura todo el contenido del backup al directorio `~/restauracion_prueba`:

```
tar -xzvf mis_datos_backup_$(date +%Y%m%d).tar.gz \
    -C ~/restauracion_prueba
```

- Al no especificar ningún archivo o directorio después del nombre del tarball, `tar` extrae todo su contenido.

3. Verifica que toda la estructura de directorios y archivos se ha restaurado:

```
ls -R ~/restauracion_prueba/mis_datos/
```

- Salida Esperada: Deberías ver toda la estructura original de `~/mis_datos`.

Ejercicio 3: Sincronización y Backup con `rsync`

Objetivo: Utilizar `rsync` para realizar copias de seguridad incrementales y eficientes, ideal para sincronizar directorios.

Parte A: Backup Inicial con `rsync`

1. Asegúrate de que tu directorio `~/mis_datos` esté en su estado original y crea un destino para `rsync`:

```
cd ~
rm -rf ~/mis_datos # Limpia si existe, para empezar con una copia limpia
```

```
mkdir -p ~/mis_datos/documentos ~/mis_datos/configuracion
echo "Mi informe secreto." > ~/mis_datos/documentos/informe.txt
echo "Contraseña123" > ~/mis_datos/configuracion/credenciales.conf
echo "Log de hoy." > ~/mis_datos/log.txt

mkdir -p ~/rsync_backups/mis_datos_sincronizados
```

2. Realiza la primera copia de seguridad (sincronización) con `rsync`:

```
rsync -avz --delete --stats ~/mis_datos/
~/rsync_backups/mis_datos_sincronizados/
```

- `-a`: Modo archivo (archiving), preserva permisos, propietario, grupo, tiempos de modificación, enlaces simbólicos, etc.
- `-v`: Muestra el progreso detallado (verbose).
- `-z`: Comprime los datos durante la transferencia (útil para red).
- `--delete`: Elimina archivos en el destino que ya no existen en el origen.
- `--stats`: Muestra un resumen de las transferencias.
- ¡Importante! La barra final `/` en `~/mis_datos/` significa "el contenido de este directorio". Si la omites, `rsync` copiará el propio directorio `mis_datos` dentro del destino, resultando en `~/rsync_backups/mis_datos_sincronizados/mis_datos/`.

3. Verifica el contenido del destino de `rsync`:

```
ls -R ~/rsync_backups/mis_datos_sincronizados/
```

Parte B: Backup Incremental con `rsync`

1. Modifica un archivo existente y crea uno nuevo en el origen:

```
echo "Información adicional del informe." >> ~/mis_datos/documentos/informe.txt
echo "Este es un nuevo archivo de datos." > ~/mis_datos/nuevo_dato.txt
```

2. Elimina un archivo del origen para probar la opción `--delete`:

```
rm ~/mis_datos/log.txt
```

3. Ejecuta `rsync` de nuevo con los mismos parámetros:

```
rsync -avz --delete --stats ~/mis_datos/
~/rsync_backups/mis_datos_sincronizados/
```

- Salida Esperada: Observa la salida. Deberías ver que `rsync` solo transfiere `informe.txt` (porque fue modificado) y `nuevo_dato.txt` (porque es nuevo). También debería indicar que `log.txt` fue eliminado del destino debido a `--delete`.

4. Verifica el contenido actualizado en el destino:

```
cat ~/rsync_backups/mis_datos_sincronizados/documentos/informe.txt
ls ~/rsync_backups/mis_datos_sincronizados/
```

- Salida Esperada: `informe.txt` debería tener el contenido actualizado, `nuevo_dato.txt` debería estar presente, y `log.txt` debería haber desaparecido.

Ejercicio 4: Copia a Nivel de Bloque con `dd` (Precaución)

Objetivo: Comprender el uso de `dd` para crear imágenes de disco/partición.

¡Advertencia! `dd` es una herramienta muy poderosa y peligrosa. Un error en la especificación de `if` (input file) u `of` (output file) puede resultar en la pérdida total de datos de tu sistema. Para este ejercicio, copiaremos una partición pequeña y segura o un archivo de prueba. No uses `dd` en tus particiones de sistema activas sin entenderlo completamente.

Parte A: Crear una Imagen de un Archivo

1. Crea un archivo de prueba para usar como "partición":

```
cd ~
dd if=/dev/zero of=test_partition.img bs=1M count=10
ls -lh test_partition.img
```

- Esto crea un archivo de 10 MB lleno de ceros.

2. Copia este "archivo de partición" a otro archivo usando `dd`:

```
dd if=test_partition.img of=test_partition_backup.img bs=1M status=progress
```

- `status=progress`: Muestra el progreso de la copia.

3. Verifica que la copia se realizó y tiene el mismo tamaño:

```
ls -lh test_partition.img test_partition_backup.img
```

Parte B: Concepto de Backup de Partición (No Ejecutar en Particiones Críticas)

1. Identifica una partición no crítica en tu sistema (ej., una partición de datos si tienes una):

```
lsblk
```

- ¡Importante! NUNCA uses `dd` directamente en `/dev/sda` o `/dev/vda` sin una comprensión completa, o si no estás en un entorno de máquina virtual desechable. Si no tienes una partición separada, no ejecutes el siguiente paso y solo entiéndelo conceptualmente.

2. Comando conceptual para hacer backup de una partición (NO EJECUTAR si no estás seguro):

```
# sudo dd if=/dev/sda1 of=/ruta/a/backup/sda1_image_$(date +%Y%m%d).img bs=4M status=progress
```

- Explicación: Esto copiaría la partición `/dev/sda1` a un archivo de imagen. Es útil para copiar particiones completas, incluyendo el espacio no utilizado, y es esencial para la recuperación ante desastres a nivel de bloque.

Ejercicio 5: Gestión de Cintas con `mt` (Conceptuales)

Objetivo: Comprender el propósito y uso básico de la utilidad `mt` para controlar unidades de cinta.

Nota: Este ejercicio es completamente conceptual a menos que dispongas de una unidad de cinta física conectada a tu máquina virtual o servidor.

1. Listar los dispositivos de cinta (si existen en tu sistema):

```
ls /dev/st* /dev/nst*
```

- `/dev/st0`: Primera unidad de cinta, rebobina automáticamente después de cada operación.
- `/dev/nst0`: Primera unidad de cinta, NO rebobina automáticamente (útil para escribir múltiples archivos en secuencia).

2. Comando conceptual para verificar el estado de una unidad de cinta:

```
# sudo mt -f /dev/nst0 status
```

- Salida esperada (ejemplo): Información sobre el tipo de cinta, estado (online, offline), si hay datos, etc.

3. Comando conceptual para rebobinar la cinta:

```
# sudo mt -f /dev/nst0 rewind
```

- Explicación: Mueve la cinta al principio del rollo.

4. Comando conceptual para avanzar la cinta un "archivo" (backup lógico):

```
# sudo mt -f /dev/nst0 fsf 1
```

- Explicación: Si has escrito múltiples backups en una sola cinta (usando `tar` con `/dev/nst0`), este comando te permite saltar al inicio del siguiente backup.

5. Comando conceptual para descargar/expulsar la cinta:

```
# sudo mt -f /dev/nst0 offline
```

- Explicación: Prepara la cinta para ser retirada físicamente de la unidad.
