



ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA
Universidad de Córdoba



Copias de seguridad y restauración

Tarea Administración Tema 8

INGENIERÍA INFORMÁTICA

Programación y Administración de Sistemas

Realizado por: **Jesús Fernández López**

- Crea un fichero **crontab** (e incluyelo en **/etc/cron.d**) que realice las siguientes copias de seguridad de **/dev/sda1**, regularmente, sobre el fichero **/var/copiaNivelX**, usando **dump**:

- Una copia de nivel 0 una vez al mes.
- Una copia de nivel 2 una vez a la semana.
- Una copia de nivel 5 a diario.

En el peor escenario, ¿cuántos comandos de restauración tendrías que aplicar para recuperar un fichero del que se realizó copia con esta configuración?

OBJETIVO: Copias de seguridad automáticas

(de 3 niveles distintos)

Para ello modificaremos un archivo **crontab**, y lo añadiremos al directorio **/etc/cron.d**, desde este ejecutaremos un script encargado de realizar las copias de seguridad con el comando **dump** visto en clase.

Un fichero **crontab** es un archivo utilizado en sistemas Unix y Linux para programar tareas automatizadas. Estas tareas se ejecutan en momentos específicos, definidos por el usuario. Crontab es la abreviatura de "cron table".

El **directorio /etc/cron.d** en sistemas Unix y Linux tiene una función especial dentro del sistema de programación de tareas automáticas con cron. En él, se pueden encontrar archivos adicionales que contienen entradas de crontab. Aquí hay algunos puntos clave sobre **/etc/cron.d**

1. Creamos un script que nos permita de forma sencilla generar copias de seguridad de distintos niveles, lo creamos desde el directorio raíz con el comando **sudo nano /var/backup.sh**, importante conocer la dirección del .sh

```
1.  #!/bin/bash
2.  # Verificar que el número de argumentos sea 1
3.  if [ $# -ne 1 ]; then
4.      echo "Uso: $0 <Nivel>"
5.      exit 1
6.  fi
7.
8.  # Verificar que el argumento es un número
9.  if ! [[ "$1" =~ ^[0-9]+$ ]]; then
10.     echo "Error: El nivel debe ser un número"
11.     exit 1
12.  fi
13.
14.  # Obtener la fecha actual
15.  echo -e "\n[$fecha_actual]"
16.  fecha_actual=$(date +%Y-%m-%d)
17.
18.  # Definir el nombre del archivo de respaldo
19.  nombre_respaldo="/var/copiaNivel$1-$fecha_actual"
20.
21.  # Crear el respaldo usando dump
22.  dump $1 -u -f "$nombre_respaldo" /dev/sda1
23.
24.  # Verificar si dump fue exitoso
25.  if [ $? -eq 0 ]; then
26.      echo "Respaldo creado exitosamente: $nombre_respaldo"
27.  else
28.      echo "Error al crear el respaldo"
29.      exit 1
30.  fi
```

Probando su funcionamiento podemos ver que necesita ser ejecutado con sudo, pero funciona correctamente creando las copias de seguridad del nivel que se le pase como argumento dentro del directorio /var, como se puede apreciar en la última línea del comando dump tras ejecutarlo:

Respaldo creado exitosamente: /var/copiaNivel0-2024-05-15

2. Abrimos el fichero `/etc/crontab`, donde tendremos una plantilla explicativa, con el comando: **sudo nano /etc/crontab**, y añadimos las siguientes líneas:

```
0 0 1 * * root /var/backup.sh 0 >>
/var/log/copias_seguridad/c0 2>&1
```

```
0 0 * * 1 root /var/backup.sh 2 >>
/var/log/copias_seguridad/c2 2>&1
```

```
0 0 * * * root /var/backup.sh 5 >>
/var/log/copias_seguridad/c5 2>&1
```

Como se ha explicado antes, el script necesitaba ejecutarse con **sudo**, justo por este motivo lo ejecutamos desde el usuario `root`, además vamos a redireccionar tanto la salida de error (respuesta del cron) como la salida estándar (se añade la fecha) a sus respectivos logs, uno por cada nivel.

3. Guardamos y copiamos el archivo al directorio `/etc/cron.d/`

```
sudo cp /etc/crontab /etc/cron.d/copias_seguridad
```

4. Cambiamos los permisos para asegurarnos de que pueda ser ejecutado y modificado por `root` pero otros usuarios solo puedan leerlo, esto se hace con el comando:

```
sudo chmod 744 /etc/cron.d/copias_seguridad
```

Además, ejecutamos:

```
sudo chown root:root /etc/cron.d/copias_seguridad
```

5. Comprobamos los cambios añadiendo una línea adicional y programando una próxima ejecución

```
[minuto próximo] * * * * root /var/backup.sh 3 >>
/var/log/copias_seguridad/pruebac3 2>&1
```