



RFID TRAFFIC ANALYST

**– SCRIPTS –
Copias de seguridad**

**Xavi Conde
Gerard Soteras**



Índice

Pasos Previos.....	2
Script de Backup MySQL.....	2
Proceso del script.....	3
Servidor de Recepción.....	3
Script de BackUp Full/Incremental.....	4
Proceso del script.....	7
Servidor de Recepción.....	8
Script de Restore.....	9
Proceso del script.....	10



Pasos Previos

Recordamos que para que los scripts se realicen correctamente, vamos a usar los siguientes comandos, para compartir las claves **RSA** y que no pida autenticación cada vez que realice una conexión.

```
ssh-keygen -t rsa
ssh-copy-id usuari@ip
```

Script de Backup MySQL

Para el primer script, hemos adaptado el código para nuestro proyecto de final de curso. El funcionamiento de este script consiste en hacer solo copias de seguridad de la base de datos **MySQL** que le especifiquemos.

Este es el script que hemos programado:

```
#!/bin/bash
# Script de backup de MySQL con compresión ZIP y transferencia remota

# Configuración de variables
dest_dir="backup" # Directorio de destino en la máquina remota
dest_server="xconde@100.77.20.40" # Servidor de respaldo
backup_date=$(date +%Y-%m-%d_%H%M) # Fecha del backup
log_file="$HOME/bkplog/bkp.log" # Archivo log

# Credenciales de MySQL (el usuario y contraseña están en ~/.my.cnf)
database_name="testdb"

# Comprobaciones
if [[ -z "$dest_dir" || -z "$dest_server" ]]; then
    echo "[$(date)] ERROR: Faltan datos en la configuración." | tee -a "$log_file"
    exit 1
fi

mkdir -p "$(dirname "$log_file")" # Crear carpeta de logs si no existe
mkdir -p "/tmp/backup_temp"

# Backup de la base de datos
mysql_backup="/tmp/backup_mysql_${backup_date}.sql"
echo "[$(date)] Realizando backup de la base de datos MySQL..." | tee -a "$log_file"
mysqldump --defaults-extra-file=~/.my.cnf --databases "$database_name" >
"$mysql_backup"

if [ $? -eq 0 ]; then
    echo "[$(date)] Backup de MySQL completado." | tee -a "$log_file"
else
    echo "[$(date)] ERROR al hacer backup de MySQL." | tee -a "$log_file"
    exit 1
fi

# Comprimir en ZIP
```



```
zip_file="/tmp/backup_${backup_date}.zip"
echo "[$(date)] Creando archivo ZIP: $zip_file" | tee -a "$log_file"
zip -r "$zip_file" "$mysql_backup" >> "$log_file" 2>&1

# Transferir al servidor
ssh "$dest_server" "mkdir -p $dest_dir"
scp "$zip_file" "$dest_server:$dest_dir/backup_${backup_date}.zip" >> "$log_file" 2>&1
ssh "$dest_server" "ls -t $dest_dir/backup_*.zip | tail -n +6 | xargs rm -f"

# Generar hash SHA-256
echo "[$(date)] Generando hash SHA-256 del backup" | tee -a "$log_file"
sha256sum "$zip_file" | tee -a "$log_file"

# Limpiar archivos temporales
rm -f "$zip_file" "$mysql_backup"

echo "[$(date)] Backup finalizado correctamente." | tee -a "$log_file"
```

Proceso del script

Al ejecutar el script, este nos hará un pequeño resumen del proceso que se ha seguido durante la ejecución de este script.

```
~/Escritorio
./dbbackup.sh
[mar 25 feb 2025 16:55:38 CET] Realizando backup de la base de datos MySQL...
[mar 25 feb 2025 16:55:38 CET] Backup de MySQL completado.
[mar 25 feb 2025 16:55:38 CET] Creando archivo ZIP: /tmp/backup_2025-02-25_1655.zip
[mar 25 feb 2025 16:55:39 CET] Generando hash SHA-256 del backup
95a60e53a50e31ae412d4f04431fe2cad6a41ec7d0e764503cfb0eabfc6e4c8 /tmp/backup_2025-02-25_1655.zip
[mar 25 feb 2025 16:55:39 CET] Backup finalizado correctamente.
```

Vemos que dentro del path **/home/xconde/bkpllog**, se encuentra un archivo llamado **bkp.log**, este se ha generado durante la ejecución del script.

Este nos muestra un resumen de la ejecución y nos registra los errores.

```
~/Escritorio
cat ~/bkpllog/bkp.log
[mar 25 feb 2025 16:55:38 CET] Realizando backup de la base de datos MySQL...
[mar 25 feb 2025 16:55:38 CET] Backup de MySQL completado.
[mar 25 feb 2025 16:55:38 CET] Creando archivo ZIP: /tmp/backup_2025-02-25_1655.zip
adding: tmp/backup_mysql_2025-02-25_1655.sql (deflated 62%)
[mar 25 feb 2025 16:55:39 CET] Generando hash SHA-256 del backup
95a60e53a50e31ae412d4f04431fe2cad6a41ec7d0e764503cfb0eabfc6e4c8 /tmp/backup_2025-02-25_1655.zip
[mar 25 feb 2025 16:55:39 CET] Backup finalizado correctamente.
```

Servidor de Recepción

Si entramos dentro del servidor que recibe el backup, nos encontraremos en el path **/home/xconde/backup**, un archivo **.zip** que si lo descomprimos encontraremos la base de datos que hemos especificado en la copia de seguridad.

```
(xconde@xconde) - [~]
$ ls ~/backup
backup_2025-02-25_1655.zip
```



Script de BackUp Full/Incremental

Este es el script que vamos a utilizar:

```
#!/bin/bash

# Configuración de variables
DIRECTORIOS="$HOME/Escritorio" # Directorio a respaldar
CARPETA_TEMPORAL="/tmp/backup_temp" # Carpeta temporal para backup
LOG_FILE="$HOME/bkplog/backup.log" # Archivo log
ULTIMO_BACKUP="/tmp/ultimo_backup"
ARCHIVO_LISTA_BACKUP="$ULTIMO_BACKUP/lista_backup.txt" # Lista de archivos respaldados

# Datos del servidor de respaldo
SERVIDOR_RESPALDO="xconde@100.77.20.40" # Servidor de respaldo
RUTA_RESPALDO="backup" # Ruta en el servidor de respaldo

# Fecha del backup
FECHA_BACKUP=$(date +%Y-%m-%d_%H%M)

# Validar argumento de entrada
TIPO_BACKUP=$1 # El tipo de backup ahora se pasa como argumento

if [[ "$TIPO_BACKUP" != "full" && "$TIPO_BACKUP" != "incremental" ]]; then
    echo "Uso: ./backup.sh [full o incremental]"
    echo "Ejemplo: ./backup.sh full -> Realiza un backup completo"
    echo "          ./backup.sh incremental -> Realiza un backup incremental"
    exit 1
fi

# Crear directorios necesarios
mkdir -p "$(dirname "$LOG_FILE")"
mkdir -p "$CARPETA_TEMPORAL"
mkdir -p "$ULTIMO_BACKUP"

# Función para registrar mensajes en el log
log_message() {
    echo "[$(date)] $1" | tee -a "$LOG_FILE"
}

# Asegurar que no queden residuos de backups anteriores
rm -rf "$CARPETA_TEMPORAL"
mkdir -p "$CARPETA_TEMPORAL"

# Asignar identificador de tipo de backup
if [[ "$TIPO_BACKUP" == "full" ]]; then
    ID_BACKUP="F"
else
    ID_BACKUP="I"
fi

if [[ "$TIPO_BACKUP" == "full" ]]; then
```



```
log_message "Iniciando backup FULL."

# Copiar los archivos del directorio a la carpeta temporal
rsync -av --progress "$DIRECTORIOS/" "$CARPETA_TEMPORAL/" >> "$LOG_FILE"
2>&1

# Crear el archivo ZIP
ZIP_NAME="backup_${ID_BACKUP}_${FECHA_BACKUP}.zip"
log_message "Creando archivo ZIP: $ZIP_NAME"
cd "$CARPETA_TEMPORAL" || exit
zip -r "../$ZIP_NAME" .* >> "$LOG_FILE" 2>&1
cd - > /dev/null

# Verificar que el ZIP se creó correctamente
if [[ ! -f "/tmp/$ZIP_NAME" ]]; then
    log_message "ERROR: No se pudo crear el archivo ZIP."
    exit 1
fi

# Transferir el backup al servidor
ssh "$SERVIDOR_RESPALDO" "mkdir -p $RUTA_RESPALDO"
scp "/tmp/$ZIP_NAME" "$SERVIDOR_RESPALDO:$RUTA_RESPALDO/$ZIP_NAME"
>> "$LOG_FILE" 2>&1

# Guardar la lista de archivos respaldados con su fecha de modificación
find "$DIRECTORIOS" -type f -exec stat --format="%n %Y" {} \; >
"$ARCHIVO_LISTA_BACKUP"

log_message "Backup FULL completado y transferido."

elif [[ "$TIPO_BACKUP" == "incremental" ]]; then
    log_message "Iniciando backup INCREMENTAL."

    if [[ ! -f "$ARCHIVO_LISTA_BACKUP" ]]; then
        log_message "No hay backup previo. Se recomienda hacer un backup FULL
primero."
        exit 1
    fi

    # Crear lista de archivos modificados
    archivos_modificados=()

    while IFS= read -r line; do
        archivo=$(echo "$line" | cut -d ' ' -f 1)
        fecha_backup=$(echo "$line" | cut -d ' ' -f 2)

        if [[ -f "$archivo" ]]; then
            fecha_actual=$(stat --format="%Y" "$archivo")

            if [[ "$fecha_actual" -gt "$fecha_backup" ]]; then
                log_message "Archivo modificado: $archivo"
                archivos_modificados+=("$archivo")
            fi
        fi
    done
```



```
fi
done < "$ARCHIVO_LISTA_BACKUP"

if [[ ${#archivos_modificados[@]} -eq 0 ]]; then
    log_message "No se encontraron archivos modificados. No se realizará backup
incremental."
    exit 0
fi

# Copiar los archivos modificados a la carpeta temporal
for archivo in "${archivos_modificados[@]}"; do
    destino="$CARPETA_TEMPORAL/$archivo"
    mkdir -p "$(dirname "$destino")"
    cp "$archivo" "$destino"
done

# Crear el archivo ZIP
ZIP_NAME="backup_${ID_BACKUP}_${FECHA_BACKUP}.zip"
log_message "Creando archivo ZIP: $ZIP_NAME"
cd "$CARPETA_TEMPORAL" || exit
zip -r "$ZIP_NAME" ./* >> "$LOG_FILE" 2>&1
cd - > /dev/null

if [[ ! -f "/tmp/$ZIP_NAME" ]]; then
    log_message "ERROR: No se pudo crear el archivo ZIP."
    exit 1
fi

# Transferir el backup incremental al servidor
ssh "$SERVIDOR_RESPALDO" "mkdir -p $RUTA_RESPALDO"
scp "/tmp/$ZIP_NAME" "$SERVIDOR_RESPALDO:$RUTA_RESPALDO/$ZIP_NAME"
>> "$LOG_FILE" 2>&1

# Actualizar la lista de archivos respaldados con su fecha de modificación
find "$DIRECTORIOS" -type f -exec stat --format="%n %Y" {} \; >
"$ARCHIVO_LISTA_BACKUP"

log_message "Backup INCREMENTAL completado y transferido."

else
    log_message "ERROR: Tipo de backup no válido. Debes elegir 'full' o 'incremental'."
    exit 1
fi

# Limpiar archivos temporales
rm -rf "$CARPETA_TEMPORAL"

log_message "Backup finalizado correctamente."
```



Proceso del script

Vamos a ejecutar el script.

Vemos que cuando lo ejecutamos nos muestra un pequeño manual de uso, donde vemos que podemos elegir entre **full** o **incremental**.

```
~/Escritorio
./backup.sh
Uso: ./backup.sh [full o incremental]
Ejemplo: ./backup.sh full -> Realiza un backup completo
          ./backup.sh incremental -> Realiza un backup incremental
```

Si indicamos que queremos hacer una copia **full**, cojera todos los archivos y directorios dentro del directorio que hemos especificado en el script y los copiará enteros.

```
~/Escritorio
./backup.sh full
[mar 25 feb 2025 17:09:47 CET] Iniciando backup FULL.
[mar 25 feb 2025 17:09:47 CET] Creando archivo ZIP: backup_F_2025-02-25_1709.zip
[mar 25 feb 2025 17:09:47 CET] Backup FULL completado y transferido.
[mar 25 feb 2025 17:09:47 CET] Backup finalizado correctamente.
```

Y si elegimos **incremental**, vemos que busca archivos modificados para únicamente, hacer el backup de las modificaciones.

En este caso, vemos que no ha encontrado ningún archivo modificado y no ha realizado la copia incremental.

```
~/Escritorio
./backup.sh incremental
[mar 25 feb 2025 17:10:07 CET] Iniciando backup INCREMENTAL.
[mar 25 feb 2025 17:10:07 CET] No se encontraron archivos modificados. No se realizará backup incremental.
```

Pero si modificamos un archivo en este caso **/home/xconde/Escritorio/test**, vemos que no reconoce los archivos modificados y hace la copia de seguridad de estos archivos.

```
~/Escritorio
./backup.sh incremental
[mar 25 feb 2025 17:10:22 CET] Iniciando backup INCREMENTAL.
[mar 25 feb 2025 17:10:22 CET] Archivo modificado: /home/xconde/Escritorio/test
[mar 25 feb 2025 17:10:23 CET] Creando archivo ZIP: backup_I_2025-02-25_1710.zip
[mar 25 feb 2025 17:10:23 CET] Backup INCREMENTAL completado y transferido.
[mar 25 feb 2025 17:10:23 CET] Backup finalizado correctamente.
```




En esta captura, podemos ver el resultado del archivo de logs **backup.log**, que nos muestra todo el procedimiento, por si necesitamos identificar patrones o incidencias.

```
adding: M11/A1/origen/ (stored 0%)
adding: M11/A1/origen/que (stored 0%)
adding: M11/A1/origen/tal (stored 0%)
adding: M11/A1/origen/Hola (stored 0%)
adding: test (stored 0%)
[mar 25 feb 2025 17:09:47 CET] Backup FULL completado y transferido.
[mar 25 feb 2025 17:09:47 CET] Backup finalizado correctamente.
[mar 25 feb 2025 17:10:07 CET] Iniciando backup INCREMENTAL.
[mar 25 feb 2025 17:10:07 CET] No se encontraron archivos modificados. No se realizará backup incremental.
[mar 25 feb 2025 17:10:22 CET] Iniciando backup INCREMENTAL.
[mar 25 feb 2025 17:10:22 CET] Archivo modificado: /home/xconde/Escritorio/test
[mar 25 feb 2025 17:10:23 CET] Creando archivo ZIP: backup_I_2025-02-25_1710.zip
adding: home/ (stored 0%)
adding: home/xconde/ (stored 0%)
adding: home/xconde/Escritorio/ (stored 0%)
adding: home/xconde/Escritorio/test (stored 0%)
[mar 25 feb 2025 17:10:23 CET] Backup INCREMENTAL completado y transferido.
[mar 25 feb 2025 17:10:23 CET] Backup finalizado correctamente.
```

~/Escritorio
cat ~/bkplog/backup.log

Servidor de Recepción

Vemos que si buscamos en el servidor donde se alojan las copias de seguridad y en su directorio.

Además vemos que identifica los backups de la siguiente manera:

Full → **_F_**

Incremental → **_I_**

```
ssh xconde@100.77.20.40 "ls backup"
backup_F_2025-02-25_1709.zip
backup_I_2025-02-25_1710.zip
```



Script de Restore

Este es el script que hemos usado:

```
#!/bin/bash
# Script de recuperación de backups - Creado por Xavi y Gerard

# Asignamos variables
dest_dir="backup" # Directorio donde se encuentran los bkp
local_restore_dir="$HOME/Escritorio/restore" # Directorio donde restaurar los bkp
dest_server="xconde@100.77.20.40" # Dirección del servidor que contiene los bkp
log_file="$HOME/Escritorio/bkplog/restore.log" # Archivo log

# Si no existen, creamos los directorios
mkdir -p "$(dirname "$log_file")"
mkdir -p "$local_restore_dir"

# Listamos los ultimos 5 bkp's disponibles en el servidor
echo "[$(date)] Obteniendo lista de backups disponibles..." | tee -a "$log_file"
backups=$(ssh "$dest_server" "ls -t $dest_dir/backup_*.zip 2>/dev/null")

#Output de errores
if [ ${#backups[@]} -eq 0 ]; then
    echo "[$(date)] ERROR: No se encontraron backups disponibles en el servidor." | tee -a "$log_file"
    exit 1
fi

# Mostrar los últimos 5 backups disponibles
echo "Los últimos backups disponibles son:" | tee -a "$log_file"
for i in ${!backups[@]}; do
    echo "$((i + 1)). ${backups[$i]}" | tee -a "$log_file"
    if [ $i -eq 4 ]; then
        break
    fi
done

# Solicitar al usuario que elija un backup
echo -n "Introduce el número del backup que deseas restaurar (1-5): "
read backup_choice

#Output de errores
if ! [[ "$backup_choice" =~ ^[1-5]$ ]] || [ "$backup_choice" -gt ${#backups[@]} ]; then
    echo "[$(date)] ERROR: Elección no válida." | tee -a "$log_file"
    exit 1
fi

# Comprobar el bkp seleccionado
selected_backup=${backups[$((backup_choice - 1))]}
echo "[$(date)] Has seleccionado restaurar: $selected_backup" | tee -a "$log_file"

# Descargar el bkp seleccionado
local_zip="$local_restore_dir/${basename "$selected_backup"}"
```



```
echo "[$(date)] Descargando el backup desde el servidor: $selected_backup" | tee -a "$log_file"
scp "$dest_server:$selected_backup" "$local_zip" >> "$log_file" 2>&1

#Output de errores
if [ ! -f "$local_zip" ]; then
    echo "[$(date)] ERROR: El archivo no se descargó correctamente." | tee -a "$log_file"
    exit 1
fi

# Imprimimos el resultado
echo "[$(date)] El backup $local_zip, ha sido restaurado exitosamente." | tee -a "$log_file"
echo "Podras encontrar el backup comprimido en: $local_restore_dir." | tee -a "$log_file"
```

Proceso del script

Procederemos a ejecutar el script.

Nos pedirá que elijamos la copia de seguridad para restaurarla, en este caso, como las copias de seguridad almacenan como máximo 5 copias, nos pide que elijamos del 1-5.

```
soteras@soteras:~/Escritorio/andres$ ./recover.sh
[mar 25 feb 2025 15:56:08 CET] Obteniendo lista de backups disponibles...
Los últimos backups disponibles son:
1. backup/backup_2025-02-25_1556.zip
2. backup/backup_2025-02-25_1555.zip
3. backup/backup_2025-02-25_1554.zip
4. backup/backup_2025-02-25_1548.zip
5. backup/backup_2025-02-25_1547.zip
Introduce el número del backup que deseas restaurar (1-5):
```

En este caso para la prueba la haremos recuperando el backup con el número 1 asignado. Y veremos que en pantalla nos dice que todos los procesos se han realizado correctamente. Y en este caso, la ruta donde hemos recuperado los archivos es

/home/soteras/Escritorio/restore

```
Introduce el número del backup que deseas restaurar (1-5): 1
[mar 25 feb 2025 15:59:21 CET] Has seleccionado restaurar: backup/backup_2025-02-25_1556.zip
[mar 25 feb 2025 15:59:21 CET] Descargando el backup desde el servidor: backup/backup_2025-02-25_1556.zip
[mar 25 feb 2025 15:59:21 CET] El backup /home/soteras/Escritorio/restore/backup_2025-02-25_1556.zip, ha sido restaurado exitosamente.
Podras encontrar el backup comprimido en: /home/soteras/Escritorio/restore.
```



Si accedemos al path `/home/soteras/Escritorio`, veremos el directorio `restore` y encontraremos dos directorios `/restore` y `/bkplog`.

```
soteras@soteras:~/Escritorio$ ls -la
total 20
drwxr-xr-x  5 soteras soteras 4096 feb 25 15:42 .
drwxr-xr-x 15 soteras soteras 4096 feb 25 15:34 ..
drwxrwxr-x  2 soteras soteras 4096 feb 25 15:30 andres
drwxrwxr-x  2 soteras soteras 4096 feb 25 15:42 bkplog
drwxrwxr-x  2 soteras soteras 4096 feb 25 15:59 restore
```

En el directorio `/bkplog`, encontraremos el archivo `restore.log`.

```
soteras@soteras:~/Escritorio/bkplog$ ls -la
total 12
drwxrwxr-x 2 soteras soteras 4096 feb 25 15:42 .
drwxr-xr-x 5 soteras soteras 4096 feb 25 15:42 ..
-rw-rw-r-- 1 soteras soteras 1039 feb 25 15:59 restore.log
```

Si vemos el contenido, encontraremos que guarda los pasos que se han hecho durante todo el proceso de restauración.

```
soteras@soteras:~/Escritorio/bkplog$ cat restore.log
[mar 25 feb 2025 15:42:55 CET] Obteniendo lista de backups disponibles...
[mar 25 feb 2025 15:42:55 CET] ERROR: No se encontraron backups disponibles en el servidor.
[mar 25 feb 2025 15:46:37 CET] Obteniendo lista de backups disponibles...
[mar 25 feb 2025 15:46:37 CET] ERROR: No se encontraron backups disponibles en el servidor.
[mar 25 feb 2025 15:56:08 CET] Obteniendo lista de backups disponibles...
Los últimos backups disponibles son:
1. backup/backup_2025-02-25_1556.zip
2. backup/backup_2025-02-25_1555.zip
3. backup/backup_2025-02-25_1554.zip
4. backup/backup_2025-02-25_1548.zip
5. backup/backup_2025-02-25_1547.zip
[mar 25 feb 2025 15:59:21 CET] Has seleccionado restaurar: backup/backup_2025-02-25_1556.zip
[mar 25 feb 2025 15:59:21 CET] Descargando el backup desde el servidor: backup/backup_2025-02-25_1556.zip
[mar 25 feb 2025 15:59:21 CET] El backup /home/soteras/Escritorio/restore/backup_2025-02-25_1556.zip, ha sido restaurado exitosamente.
Podras encontrar el backup comprimido en: /home/soteras/Escritorio/restore.
```

Si entramos en el directorio `/restore`, encontraremos un archivo `.zip` con el backup que hemos seleccionado para restaurar.

```
soteras@soteras:~/Escritorio/restore$ ls -la
total 12
drwxrwxr-x 2 soteras soteras 4096 feb 25 15:59 .
drwxr-xr-x 5 soteras soteras 4096 feb 25 15:42 ..
-rw-r--r-- 1 soteras soteras 1053 feb 25 15:59 backup_2025-02-25_1556.zip
```