

# PRÁCTICA 7: AUTOMATIZACIÓN DE BACKUPS CON S3 Y CRONTAB

## Objetivos

Al finalizar esta práctica, serás capaz de:

- Conectar una VM Ubuntu con un bucket S3 existente en AWS Academy
- Crear scripts de backup automatizados para sincronizar archivos locales con S3
- Configurar tareas programadas con crontab para automatizar backups
- Gestionar logs de operaciones automáticas
- Restaurar archivos desde S3 a la máquina local
- Monitorizar ejecuciones automáticas en tiempo real

## Requisitos previos

- Cuenta activa en AWS Academy Learner Lab
- Bucket S3
- Máquina virtual Ubuntu 22.04 o superior
- Conexión a Internet activa en la MV
- AWS CLI ya instalado y configurado

## PARTE 0: Preparación del entorno

### Verificar AWS CLI

`aws --version`

```
root@santiago-VirtualBox:/home/santiago# aws --version
aws-cli/2.32.15 Python/3.13.11 Linux/6.14.0-37-generic exe/x86_64.ubuntu.24
```

`aws configure list`

```
root@santiago-VirtualBox:/home/santiago# aws configure list
NAME                : VALUE                                : TYPE                : LOCATION
profile             : <not set>                               : None                : None
access_key          : *****B40B                               : shared-credentials-file :
secret_key          : *****4q/8                               : shared-credentials-file :
region              : us-east-1                                : config-file         : ~/.aws/config
root@santiago-VirtualBox:/home/santiago#
```

## Crear estructura de directorios

```
mkdir -p ~/proyectos/backup-s3
mkdir -p ~/proyectos/backup-s3/datos
mkdir -p ~/proyectos/backup-s3/logs
mkdir -p ~/proyectos/backup-s3/scripts
cd ~/proyectos/backup-s3
```

```
root@santiago-VirtualBox:/home/santiago# mkdir -p ~/proyectos/backup-s3/datos
root@santiago-VirtualBox:/home/santiago# mkdir -p ~/proyectos/backup-s3/logs
root@santiago-VirtualBox:/home/santiago# mkdir -p ~/proyectos/backup-s3/scripts
root@santiago-VirtualBox:/home/santiago# cd ~/proyectos/backup-s3
root@santiago-VirtualBox:~/proyectos/backup-s3#
```

## Crear archivos de prueba

```
echo "Documento importante v1" > ~/proyectos/backup-s3/datos/documento1.txt
echo "Configuración de aplicación" > ~/proyectos/backup-s3/datos/config.json
echo "Datos de usuario" > ~/proyectos/backup-s3/datos/usuarios.csv date >
~/proyectos/backup-s3/datos/timestamp.txt
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# echo "Documento importante v1" > ~/proy
ectos/backup-s3/datos/documento1.txt
root@santiago-VirtualBox:~/proyectos/backup-s3# echo "Configuración de aplicación" > ~/
proyectos/backup-s3/datos/config.json
root@santiago-VirtualBox:~/proyectos/backup-s3# echo "Datos de usuario" > ~/proyectos/b
ackup-s3/datos/usuarios.csv date >
bash: error sintáctico cerca del elemento inesperado `newline'
root@santiago-VirtualBox:~/proyectos/backup-s3# echo "Datos de usuario" > ~/proyectos/b
ackup-s3/datos/usuarios.csv date >
bash: error sintáctico cerca del elemento inesperado `newline'
root@santiago-VirtualBox:~/proyectos/backup-s3# echo "Datos de usuario" > ~/proyectos/b
ackup-s3/datos/usuarios.csv date > ~/proyectos/backup-s3/datos/timestamp.txt
root@santiago-VirtualBox:~/proyectos/backup-s3#
```

## Verificar estructura creada

```
ls -la ~/proyectos/backup-s3/
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# ls -la ~/proyectos/backup-s3/
total 20
drwxr-xr-x 5 root root 4096 ene 30 09:38 .
drwxr-xr-x 3 root root 4096 ene 30 09:38 ..
drwxr-xr-x 2 root root 4096 ene 30 09:43 datos
drwxr-xr-x 2 root root 4096 ene 30 09:38 logs
drwxr-xr-x 2 root root 4096 ene 30 09:38 scripts
root@santiago-VirtualBox:~/proyectos/backup-s3#
```

```
tree ~/proyectos/backup-s3/
```

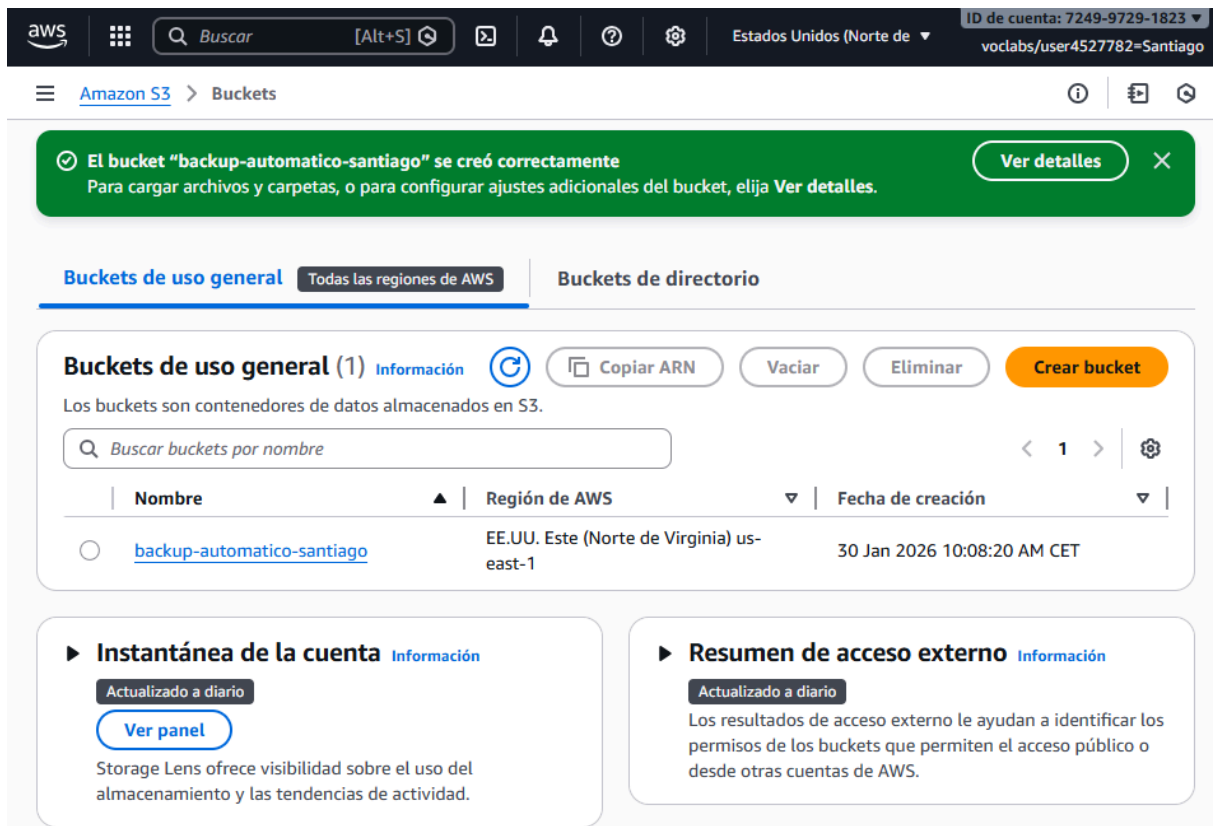
```
root@santiago-VirtualBox:~/proyectos/backup-s3# tree ~/proyectos/backup-s3/
/root/proyectos/backup-s3/
├── datos
│   ├── config.json
│   ├── documento1.txt
│   ├── timestamp.txt
│   └── usuarios.csv
├── logs
└── scripts

4 directories, 4 files
```

## PARTE 1: Crear el bucket S3 desde AWS

**IMPORTANTE:** Realiza estos pasos desde AWS Academy:

1. Accede a AWS
2. Accede al servicio **S3**
3. Haz clic en **"Create bucket"**
4. Configura el bucket:
  - **Bucket name:** backup-automatico-[tu-nombre]
  - **AWS Region:** eu-west-
  - **Block Public Access:** Desactiva "Block all public access"
  - Marca la casilla de confirmación
  - Deja el resto de opciones por defecto
5. Haz clic en **"Create bucket"**



## Verificar acceso desde la VM

aws s3 ls

```
root@santiago-VirtualBox:~/proyectos/backup-s3# aws s3 ls
2026-01-30 10:08:20 backup-automático-santiago
```

aws s3 ls s3://backup-automático-santiago

**NOTA:** Reemplaza [tu-nombre] por tu nombre real en todos los comandos siguientes.

## PARTE 2: Script de backup manual

Crea el script backup-manual.sh:

```
cat > ~/proyectos/backup-s3/scripts/backup-manual.sh <<'EOF'
```

```
#!/bin/bash
```

```
# Configuración
```

```
BUCKET_NAME="backup-automático-santiago"
```

```
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"
```

```
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
```

```
LOG_FILE="$HOME/proyectos/backup-s3/logs/backup_$(date +%Y%m%d_%H%M%S).log"
```

```
" echo "=== Inicio del backup: $(date) ===" | tee -a $LOG_FILE
```

```
# Sincronizar archivos con S3
```

```
aws s3 sync $LOCAL_DIR s3://$BUCKET_NAME/backups/  
 \ --delete \  
 2>&1 | tee -a $LOG_FILE
```

```
if [ $? -eq 0 ]; then
```

```
    echo "✓ Backup completado exitosamente" | tee -a $LOG_FILE
```

```
else
```

```
    echo "✗ Error durante el backup" | tee -a $LOG_FILE
```

```
    exit 1
```

```
fi
```

```
# Listar archivos en S3
```

```
echo -e "\n--- Archivos en S3 ---" | tee -a $LOG_FILE
```

```
aws s3 ls s3://$BUCKET_NAME/backups/ --recursive | tee -a $LOG_FILE
```

```
echo "=== Fin del backup: $(date) ===" | tee -a $LOG_FILE
```

```
EOF
```

**IMPORTANTE:** Edita el script y reemplaza [tu-nombre] con tu nombre

real: `nano ~/proyectos/backup-s3/scripts/backup-manual.sh`

Cambia la línea:

```
BUCKET_NAME="backup-automatico-santiago"
```

```
GNU nano 7.2 /root/proyectos/backup-s3/scripts/backup-manual.sh *  
#!/bin/bash  
# Configuración  
BUCKET_NAME="backup-automatico-santiago"  
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"  
TIMESTAMP=$(date +%Y%m%d_%H%M%S)  
LOG_FILE="$HOME/proyectos/backup-s3/logs/backup_$(date +%Y%m%d_%H%M%S).log" echo "=== Inicio del b  
# Sincronizar archivos con S3  
aws s3 sync $LOCAL_DIR s3://$BUCKET_NAME/backups/ \ --delete \  
2>&1 | tee -a $LOG_FILE  
if [ $? -eq 0 ]; then  
    echo "✓ Backup completado exitosamente" | tee -a $LOG_FILE else  
    echo "✗ Error durante el backup" | tee -a $LOG_FILE  
    exit 1  
fi  
# Listar archivos en S3  
echo -e "\n--- Archivos en S3 ---" | tee -a $LOG_FILE  
aws s3 ls s3://$BUCKET_NAME/backups/ --recursive | tee -a $LOG_FILE  
echo "=== Fin del backup: $(date) ===" | tee -a $LOG_FILE
```

## Dar permisos de ejecución

```
chmod +x ~/proyectos/backup-s3/scripts/backup-manual.sh
```

## Ejecutar el backup manual

```
~/proyectos/backup-s3/scripts/backup-manual.sh
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/backup-manual.sh
=== Inicio del backup: vie 30 ene 2026 10:27:11 CET ===

Unknown options:  --delete,
✓ Backup completado exitosamente
✗ Error durante el backup
```

## Verificar el resultado

```
cat ~/proyectos/backup-s3/logs/backup_*.log
```

```
aws s3 ls s3://backup-automatico-santiago/backups/ --recursive
```

## PARTE 3: Script de backup con versionado

Crea backup-versionado.sh:

```
cat > ~/proyectos/backup-s3/scripts/backup-versionado.sh
```

```
<<'EOF' Y dentro:
```

```
#!/bin/bash
```

```
BUCKET_NAME="backup-automatico-santiago"
```

```
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"
```

```
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
```

```
BACKUP_DIR="backups/version_${TIMESTAMP}"
```

```
LOG_FILE="$HOME/proyectos/backup-s3/logs/backup_versionado_${TIMESTAMP}.log"
```

```
echo "=== Backup versionado: $(date) ===" | tee -a $LOG_FILE
```

```
echo "Carpeta destino: $BACKUP_DIR" | tee -a $LOG_FILE
```

```
# Copiar archivos con timestamp
```

```
aws s3 cp $LOCAL_DIR s3://$BUCKET_NAME/$BACKUP_DIR/
```

```
\ --recursive \
```

```
2>&1 | tee -a $LOG_FILE
```

```
# Crear marcador de última versión
```

```
echo $TIMESTAMP > /tmp/last_backup.txt
```

```
aws s3 cp /tmp/last_backup.txt s3://$BUCKET_NAME/backups/LATEST.txt
```

```
echo "=== Backup completado ===" | tee -a $LOG_FILE
```

EOF

```
root@santiago-VirtualBox:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/backup-versionado.sh <<'EOF'
> #!/bin/bash
BUCKET_NAME="backup-automatiko-santiago"
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
BACKUP_DIR="backups/version_$(TIMESTAMP)"
LOG_FILE="$HOME/proyectos/backup-s3/logs/backup_versionado_$(TIMESTAMP).log"
echo "=== Backup versionado: $(date) ===" | tee -a $LOG_FILE
echo "Carpeta destino: $BACKUP_DIR" | tee -a $LOG_FILE
# Copiar archivos con timestamp
aws s3 cp $LOCAL_DIR s3://$BUCKET_NAME/$BACKUP_DIR/ \ --recursive \
2>&1 | tee -a $LOG_FILE
# Crear marcador de última versión
echo $TIMESTAMP > /tmp/last_backup.txt
aws s3 cp /tmp/last_backup.txt s3://$BUCKET_NAME/backups/LATEST.txt
echo "=== Backup completado ===" | tee -a $LOG_FILE
> EOF
```

Edita el script para reemplazar [tu-nombre]:

nano ~/proyectos/backup-s3/scripts/backup-versionado.sh

```
GNU nano 7.2 /root/proyectos/backup-s3/scripts/backup-versionado.sh
#!/bin/bash
BUCKET_NAME="backup-automatiko-santiago"
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
BACKUP_DIR="backups/version_$(TIMESTAMP)"
LOG_FILE="$HOME/proyectos/backup-s3/logs/backup_versionado_$(TIMESTAMP).log"
echo "=== Backup versionado: $(date) ===" | tee -a $LOG_FILE
echo "Carpeta destino: $BACKUP_DIR" | tee -a $LOG_FILE
# Copiar archivos con timestamp
aws s3 cp $LOCAL_DIR s3://$BUCKET_NAME/$BACKUP_DIR/ \ --recursive \
2>&1 | tee -a $LOG_FILE
# Crear marcador de última versión
echo $TIMESTAMP > /tmp/last_backup.txt
aws s3 cp /tmp/last_backup.txt s3://$BUCKET_NAME/backups/LATEST.txt
echo "=== Backup completado ===" | tee -a $LOG_FILE
```

Dar permisos y ejecutar:

chmod +x ~/proyectos/backup-s3/scripts/backup-versionado.sh

~/proyectos/backup-s3/scripts/[backup-versionado.sh](#)

## Verificar versiones creadas

aws s3 ls s3://backup-automatiko-[tu-nombre]/backups/

```
root@santiago-VirtualBox:~/proyectos/backup-s3# aws s3 ls s3://backup-automatiko-santiago/backups/
```

## PARTE 4: Script de restauración

Crea restaurar.sh:

```
cat > ~/proyectos/backup-s3/scripts/restaurar.sh <<'EOF'
```

Y dentro:

```
#!/bin/bash
```

```
BUCKET_NAME="backup-automatico-santiago"
```

```
RESTORE_DIR="$HOME/proyectos/backup-s3/restaurados"
```

```
LOG_FILE="$HOME/proyectos/backup-s3/logs/restore_$(date
```

```
 +%Y%m%d_%H%M%S).log" mkdir -p $RESTORE_DIR
```

```
echo "=== Iniciando restauración: $(date) ===" | tee -a $LOG_FILE
```

```
# Listar versiones disponibles
```

```
echo "Versiones disponibles:" | tee -a $LOG_FILE
```

```
aws s3 ls s3://$BUCKET_NAME/backups/ | tee -a $LOG_FILE
```

```
# Restaurar última versión
```

```
aws s3 sync s3://$BUCKET_NAME/backups/ $RESTORE_DIR \
```

```
2>&1 | tee -a $LOG_FILE
```

```
echo "Archivos restaurados en: $RESTORE_DIR" | tee -a $LOG_FILE
```

```
ls -lh $RESTORE_DIR | tee -a $LOG_FILE
```

```
echo "=== Restauración completada ===" | tee -a $LOG_FILE
```

```
EOF
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/res
taurar.sh <<'EOF'
> #!/bin/bash
BUCKET_NAME="backup-automatico-santiago"
RESTORE_DIR="$HOME/proyectos/backup-s3/restaurados"
LOG_FILE="$HOME/proyectos/backup-s3/logs/restore_$(date +%Y%m%d_%H%M%S).log" mkdir -p $
RESTORE_DIR
echo "=== Iniciando restauración: $(date) ===" | tee -a $LOG_FILE
# Listar versiones disponibles
echo "Versiones disponibles:" | tee -a $LOG_FILE
aws s3 ls s3://$BUCKET_NAME/backups/ | tee -a $LOG_FILE
# Restaurar última versión
aws s3 sync s3://$BUCKET_NAME/backups/ $RESTORE_DIR \
2>&1 | tee -a $LOG_FILE
echo "Archivos restaurados en: $RESTORE_DIR" | tee -a $LOG_FILE ls -lh $RESTORE_DIR | t
ee -a $LOG_FILE
echo "=== Restauración completada ===" | tee -a $LOG_FILE
> EOF
```

Edita el script para reemplazar [tu-nombre]:

`nano ~/proyectos/backup-s3/scripts/restaurar.sh`

```
GNU nano 7.2 /root/proyectos/backup-s3/scripts/restaurar.sh
#!/bin/bash
BUCKET_NAME="backup-automatico-santiago"
RESTORE_DIR="$HOME/proyectos/backup-s3/restaurados"
LOG_FILE="$HOME/proyectos/backup-s3/logs/restore_$(date +%Y%m%d_%H%M%S).log" mkdir -p >
echo "=== Iniciando restauración: $(date) ===" | tee -a $LOG_FILE
# Listar versiones disponibles
echo "Versiones disponibles:" | tee -a $LOG_FILE
aws s3 ls s3://$BUCKET_NAME/backups/ | tee -a $LOG_FILE
# Restaurar última versión
aws s3 sync s3://$BUCKET_NAME/backups/ $RESTORE_DIR \
2>&1 | tee -a $LOG_FILE
echo "Archivos restaurados en: $RESTORE_DIR" | tee -a $LOG_FILE ls -lh $RESTORE_DIR | >
echo "=== Restauración completada ===" | tee -a $LOG_FILE
```

Dar permisos:

`chmod +x ~/proyectos/backup-s3/scripts/restaurar.sh`

```
root@santiago-VirtualBox:~/proyectos/backup-s3# chmod +x ~/proyectos/backup-s3/scripts/
restaurar.sh
```

## Probar restauración

`~/proyectos/backup-s3/scripts/restaurar.sh`

```
root@santiago-VirtualBox:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/restaurar
.sh
=== Iniciando restauración: vie 30 ene 2026 10:35:05 CET ===
Versiones disponibles:

Unknown options:
tee: opción incorrecta -- «l»
Pruebe 'tee --help' para más información.
=== Restauración completada ===
```

`ls -la ~/proyectos/backup-s3/restaurados/`

```
root@santiago-VirtualBox:~/proyectos/backup-s3# ls -la ~/proyectos/backup-s3/restaurado
s/
total 8
drwxr-xr-x 2 root root 4096 ene 30 10:35 .
drwxr-xr-x 6 root root 4096 ene 30 10:35 ..
```

`cat ~/proyectos/backup-s3/restaurados/documento1.txt`

```
root@santiago-VirtualBox:~/proyectos/backup-s3# cat ~/proyectos/backup-s3/restaurados/d
ocumento1.txt
cat: /root/proyectos/backup-s3/restaurados/documento1.txt: No existe el archivo o el di
rectorio
```

## PARTE 5: Script optimizado para cron Crea backup-cron.sh:

```
cat > ~/proyectos/backup-s3/scripts/backup-cron.sh
```

```
<<'EOF' Y dentro:
```

```
#!/bin/bash
```

```
export PATH=/usr/local/bin:/usr/bin:/bin
```

```
export AWS_CONFIG_FILE="$HOME/.aws/config"
```

```
export AWS_SHARED_CREDENTIALS_FILE="$HOME/.aws/credentials"
```

```
BUCKET_NAME="backup-automatico-santiago"
```

```
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"
```

```
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
```

```
LOG_DIR="$HOME/proyectos/backup-s3/logs"
```

```
LOG_FILE="$LOG_DIR/cron_backup_${TIMESTAMP}.log"
```

```
" mkdir -p $LOG_DIR
```

```
{
```

```
echo "====="
```

```
echo "Backup automático iniciado"
```

```
echo "Fecha: $(date)"
```

```
echo "===== " aws s3
```

```
sync $LOCAL_DIR s3://$BUCKET_NAME/backups/ --delete
```

```
if [ $? -eq 0 ]; then
```

```
    echo "✓ Backup exitoso"
```

```
    echo "Archivos sincronizados:"
```

```
    aws s3 ls s3://$BUCKET_NAME/backups/ --recursive
```

```
else
```

```
    echo "✗ Error en el backup"
```

```
fi
```

```
echo "====="
```

```
echo "Backup finalizado: $(date)"
```

```
echo "====="
```

```
} >> $LOG_FILE 2>&1
```

```
# Limpiar logs antiguos (mantener últimos 7 días)
```

```
find $LOG_DIR -name "cron_backup_*.log" -mtime +7 -delete
```

```
EOF
```

```

root@santiago-VirtualBox:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/backup-cron.sh <<'EOF'
> #!/bin/bash
export PATH=/usr/local/bin:/usr/bin:/bin
export AWS_CONFIG_FILE="$HOME/.aws/config"
export AWS_SHARED_CREDENTIALS_FILE="$HOME/.aws/credentials"
BUCKET_NAME="backup-automático-santiago"
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
LOG_DIR="$HOME/proyectos/backup-s3/logs"
LOG_FILE="$LOG_DIR/cron_backup_$TIMESTAMP.log" mkdir -p $LOG_DIR
{
echo "=====" echo "Backup automático iniciado"
echo "Fecha: $(date)"
echo "=====" aws s3 sync $LOCAL_DIR s3://$BUCKET_NAME/ --delete
if [ $? -eq 0 ]; then
echo "✓ Backup exitoso"
echo "Archivos sincronizados:"
aws s3 ls s3://$BUCKET_NAME/backups/ --recursive
else
echo "X Error en el backup"
fi
echo "=====" echo "Backup finalizado: $(date)"
echo "=====" } >> $LOG_FILE 2>&1
# Limpiar logs antiguos (mantener últimos 7 días)
find $LOG_DIR -name "cron_backup_*.log" -mtime +7 -delete
> EOF

```

Edita el script para reemplazar [tu-nombre]:

**nano ~/proyectos/backup-s3/scripts/backup-cron.sh**

```

GNU nano 7.2 /root/proyectos/backup-s3/scripts/backup-cron.sh
#!/bin/bash
export PATH=/usr/local/bin:/usr/bin:/bin
export AWS_CONFIG_FILE="$HOME/.aws/config"
export AWS_SHARED_CREDENTIALS_FILE="$HOME/.aws/credentials"
BUCKET_NAME="backup-automático-santiago"
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
LOG_DIR="$HOME/proyectos/backup-s3/logs"
LOG_FILE="$LOG_DIR/cron_backup_$TIMESTAMP.log" mkdir -p $LOG_DIR
{
echo "=====" echo "Backup automático iniciado"
echo "Fecha: $(date)"
echo "=====" aws s3 sync $LOCAL_DIR s3://$BUCKET_NAME/ --delete
if [ $? -eq 0 ]; then
echo "✓ Backup exitoso"
echo "Archivos sincronizados:"
aws s3 ls s3://$BUCKET_NAME/backups/ --recursive
else
echo "X Error en el backup"
fi
echo "=====" echo "Backup finalizado: $(date)"
echo "=====" } >> $LOG_FILE 2>&1
# Limpiar logs antiguos (mantener últimos 7 días)
find $LOG_DIR -name "cron_backup_*.log" -mtime +7 -delete

```

Dar permisos de ejecución:

```
chmod +x ~/proyectos/backup-s3/scripts/backup-cron.sh
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# chmod +x ~/proyectos/backup-s3/scripts/backup-cron.sh
```

## Probar el script manualmente

```
~/proyectos/backup-s3/scripts/backup-cron.sh
```

```
cat ~/proyectos/backup-s3/logs/cron_backup_*.log
```

## PARTE 6: Configurar crontab para ejecución automática

### Editar el crontab del usuario

```
crontab -e
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# cat ~/proyectos/backup-s3/logs/cron_backup_*.log
cat: '/root/proyectos/backup-s3/logs/cron_backup_*.log': No existe el archivo o el directorio
root@santiago-VirtualBox:~/proyectos/backup-s3# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]:
```

Si es la primera vez, te preguntará qué editor usar. Selecciona nano (opción

### 1). Añadir la tarea programada

Añade la siguiente línea al final del archivo:

```
*/3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
```

**Explicación:**

- \*/3 = cada 3 minutos
- \* \* \* \* = cualquier hora, día, mes y día de la semana
- La ruta debe usar \$HOME para que funcione correctamente

Guardar y salir

```
GNU nano 7.2 /tmp/crontab.iz0i0s/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
```

## Verificar que cron está configurado

`crontab -l`

Deberías ver:

```
*/3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
```

## PARTE 7: Monitorización en tiempo real

### Script de monitorización

Crea monitor.sh:

```
cat > ~/proyectos/backup-s3/scripts/monitor.sh
```

<<'EOF' Y dentro:

```
#!/bin/bash
```

```
BUCKET_NAME="backup-automatico-santiago"
```

```
LOG_DIR="$HOME/proyectos/backup-s3/logs"
```

```
echo "=====
```

```
echo "MONITOR DE BACKUPS AUTOMÁTICOS"
```

```
echo "=====
```

```
echo ""
```

```
echo " Últimos 5 logs generados:"
```

```
ls -lht $LOG_DIR/cron_backup_*.log 2>/dev/null | head -5
```

```
echo ""
```

```
echo " Contenido del último backup:"
```

```
if ls $LOG_DIR/cron_backup_*.log 1> /dev/null 2>&1; then
```

```
tail -20 $(ls -t $LOG_DIR/cron_backup_*.log | head -1)
```

```
else
```

```
echo "Aún no hay logs de cron. Espera 3 minutos."
```

```
fi
```

```
echo ""
```

```
echo "Archivos actuales en S3:"
```

```
aws s3 ls s3://$BUCKET_NAME/backups/ --recursive --human-readable
```

```
echo ""
```

```
echo "Tamaño total del bucket:"
```

```
aws s3 ls s3://$BUCKET_NAME --recursive --summarize | grep "Total Size"
```

```
echo ""
```

```
echo "=====
```

```
EOF
```

```

root@santiago-VirtualBox:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/monitor.sh <<'EOF'
> #!/bin/bash
BUCKET_NAME="backup-automático-santiago" LOG_DIR="$HOME/proyectos/backup-s3/logs"
echo "===== " echo "MONITOR DE BACKUPS AUTOMÁTICOS"
echo "===== " echo ""
echo "Últimos 5 logs generados:"
ls -lht $LOG_DIR/cron_backup_*.log 2>/dev/null | head -5
echo ""
echo "Contenido del último backup:"
if ls $LOG_DIR/cron_backup_*.log 1> /dev/null 2>&1; then tail -20 $(ls -t $LOG_DIR/cron_backup_*.log | head -1) else
echo "Aún no hay logs de cron. Espera 3 minutos."
fi
echo ""
echo "Archivos actuales en S3:"
aws s3 ls s3://$BUCKET_NAME/backups/ --recursive --human-readable
echo ""
echo "Tamaño total del bucket:"
aws s3 ls s3://$BUCKET_NAME --recursive --summarize | grep "Total Size"
echo ""
echo "===== "
> EOF

```

Edita el script para reemplazar [tu-nombre]:

`nano ~/proyectos/backup-s3/scripts/monitor.sh`

```

GNU nano 7.2 /root/proyectos/backup-s3/scripts/monitor.sh
#!/bin/bash
BUCKET_NAME="backup-automático-santiago" LOG_DIR="$HOME/proyectos/backup-s3/logs"
echo "===== " echo "MONITOR DE BACKUPS AUTOMÁTICOS"
echo "===== " echo ""
echo "Últimos 5 logs generados:"
ls -lht $LOG_DIR/cron_backup_*.log 2>/dev/null | head -5
echo ""
echo "Contenido del último backup:"
if ls $LOG_DIR/cron_backup_*.log 1> /dev/null 2>&1; then tail -20 $(ls -t $LOG_DIR/cro
echo "Aún no hay logs de cron. Espera 3 minutos."
fi
echo ""
echo "Archivos actuales en S3:"
aws s3 ls s3://$BUCKET_NAME/backups/ --recursive --human-readable
echo ""
echo "Tamaño total del bucket:"
aws s3 ls s3://$BUCKET_NAME --recursive --summarize | grep "Total Size"
echo ""
echo "===== "

```

Dar permisos:

`chmod +x ~/proyectos/backup-s3/scripts/monitor.sh`

```

root@santiago-VirtualBox:~/proyectos/backup-s3# chmod +x ~/proyectos/backup-s3/scripts/monitor.sh

```

## Ejecutar el monitor

Espera 3-6 minutos después de configurar cron y

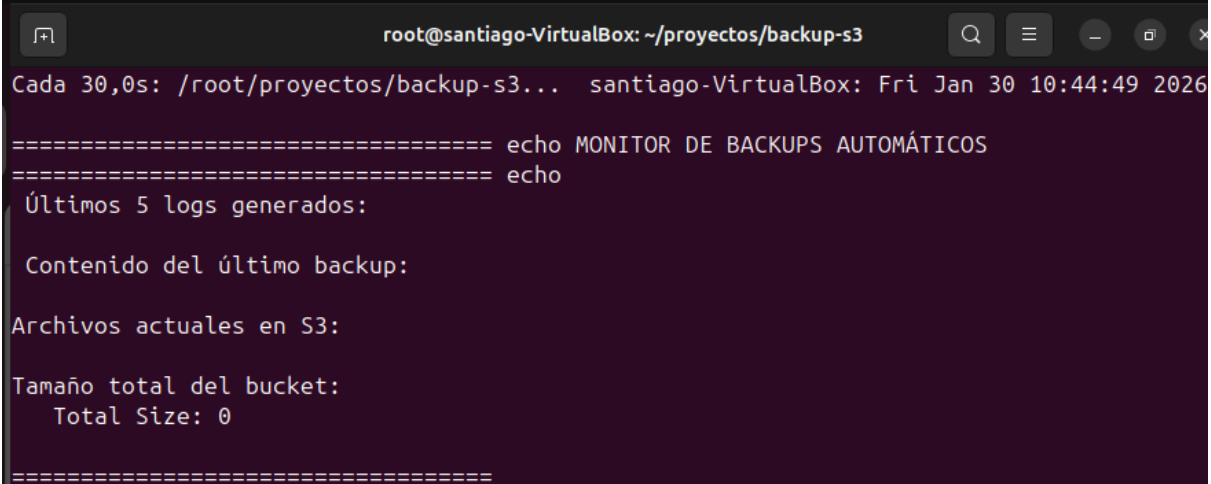
ejecuta: `~/proyectos/backup-s3/scripts/monitor.sh`

fzd

## Monitorización continua

Para ver actualizaciones cada 30 segundos:

`watch -n 30 ~/proyectos/backup-s3/scripts/monitor.sh`



```
root@santiago-VirtualBox: ~/proyectos/backup-s3
Cada 30,0s: /root/proyectos/backup-s3... santiago-VirtualBox: Fri Jan 30 10:44:49 2026
===== echo MONITOR DE BACKUPS AUTOMÁTICOS
===== echo
Últimos 5 logs generados:

Contenido del último backup:

Archivos actuales en S3:

Tamaño total del bucket:
  Total Size: 0
=====
```

Presiona Ctrl+C para detener.

## PARTE 8: Verificar ejecuciones de cron

### Ver los logs del sistema cron

`grep CRON /var/log/syslog | tail -20`

```
root@santiago-VirtualBox:~/proyectos/backup-s3# grep CRON /var/log/syslog | tail -20
2026-01-30T09:45:01.193433+01:00 santiago-VirtualBox CRON[3544]: (root) CMD (command -v
debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T09:55:01.459788+01:00 santiago-VirtualBox CRON[4809]: (root) CMD (command -v
debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T10:05:01.531592+01:00 santiago-VirtualBox CRON[4821]: (root) CMD (command -v
debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T10:15:01.679202+01:00 santiago-VirtualBox CRON[5037]: (root) CMD (command -v
debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T10:17:01.708299+01:00 santiago-VirtualBox CRON[5048]: (root) CMD (cd / && ru
n-parts --report /etc/cron.hourly)
2026-01-30T10:25:01.935877+01:00 santiago-VirtualBox CRON[5156]: (root) CMD (command -v
debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T10:30:01.969326+01:00 santiago-VirtualBox CRON[5252]: (root) CMD ([ -x /etc/
init.d/anacron ] && if [ ! -d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron
start >/dev/null; fi)
2026-01-30T10:35:01.999882+01:00 santiago-VirtualBox CRON[5341]: (root) CMD (command -v
debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T10:42:01.018724+01:00 santiago-VirtualBox CRON[5473]: (root) CMD ($HOME/proy
ectos/backup-s3/scripts/backup-cron.sh)
2026-01-30T10:42:01.022451+01:00 santiago-VirtualBox CRON[5472]: (CRON) info (No MTA in
stalled, discarding output)
2026-01-30T10:45:01.057326+01:00 santiago-VirtualBox CRON[5564]: (root) CMD ($HOME/proy
ectos/backup-s3/scripts/backup-cron.sh)
2026-01-30T10:45:01.057691+01:00 santiago-VirtualBox CRON[5565]: (root) CMD (command -v
debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T10:45:01.060204+01:00 santiago-VirtualBox CRON[5563]: (CRON) info (No MTA in
stalled, discarding output)
2026-01-30T10:48:01.083433+01:00 santiago-VirtualBox CRON[5786]: (root) CMD ($HOME/proy
ectos/backup-s3/scripts/backup-cron.sh)
2026-01-30T10:48:01.083520+01:00 santiago-VirtualBox CRON[5785]: (CRON) info (No MTA in
stalled, discarding output)
```

## Listar todos los logs generados

```
ls -lht ~/proyectos/backup-s3/logs/
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# ls -lht ~/proyectos/backup-s3/logs/
total 0
```

## Ver contenido de un log específico

```
cat ~/proyectos/backup-s3/logs/cron_backup_*.log
```

## Contar cuántos backups se han ejecutado

```
ls ~/proyectos/backup-s3/logs/cron_backup_*.log | wc -l
```

## PARTE 9: Simulación de cambios y verificación

### Modificar archivos locales

```
echo "Documento actualizado - $(date)" >>
~/proyectos/backup-s3/datos/documento1.txt echo "Nuevo archivo creado
automáticamente" >
~/proyectos/backup-s3/datos/nuevo_$(date +%H%M%S).txt
echo "Datos adicionales" > ~/proyectos/backup-s3/datos/datos_extra.txt
root@santiago-VirtualBox:~/proyectos/backup-s3# echo "Documento actualizado - $(date)"
>> ~/proyectos/backup-s3/datos/documento1.txt
root@santiago-VirtualBox:~/proyectos/backup-s3# echo "Nuevo archivo creado automáticam
ente" >
~/proyectos/backup-s3/datos/nuevo_$(date +%H%M%S).txt
bash: error sintáctico cerca del elemento inesperado `newline'
bash: /root/proyectos/backup-s3/datos/nuevo_105103.txt: No existe el archivo o el direc
torio
root@santiago-VirtualBox:~/proyectos/backup-s3# echo "Datos adicionales" > ~/proyectos/
backup-s3/datos/datos_extra.txt
```

### Verificar los cambios localmente

```
ls -lh ~/proyectos/backup-s3/datos/
root@santiago-VirtualBox:~/proyectos/backup-s3# ls -lh ~/proyectos/backup-s3/datos/
total 16K
-rw-r--r-- 1 root root 30 ene 30 09:40 config.json
-rw-r--r-- 1 root root 18 ene 30 10:51 datos_extra.txt
-rw-r--r-- 1 root root 77 ene 30 10:50 documento1.txt
-rw-r--r-- 1 root root 22 ene 30 09:43 timestamp.txt
-rw-r--r-- 1 root root  0 ene 30 09:43 usuarios.csv

cat ~/proyectos/backup-s3/datos/documento1.txt
root@santiago-VirtualBox:~/proyectos/backup-s3# cat ~/proyectos/backup-s3/datos/documen
to1.txt
Documento importante v1
Documento actualizado - vie 30 ene 2026 10:50:53 CET
```

### Esperar la siguiente ejecución

Espera 3 minutos y verifica:

```
~/proyectos/backup-s3/scripts/monitor.sh
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/monitor.sh
===== echo MONITOR DE BACKUPS AUTOMÁTICOS
===== echo
Últimos 5 logs generados:

Contenido del último backup:

Archivos actuales en S3:

Tamaño total del bucket:
    Total Size: 0

=====
```

## Ver que los nuevos archivos están en S3

```
aws s3 ls s3://backup-automatico-[tu-nombre]/backups/ --recursive
```

## PARTE 10: Dashboard de estado completo

Crea dashboard.sh:

```
cat > ~/proyectos/backup-s3/scripts/dashboard.sh <<'EOF'
```

Y dentro:

```
#!/bin/bash
```

BUCKET NAME="backup-automatico-santiago"

LOG DIR="\$HOME/proyectos/backup-s3/logs"

clear

echo

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

echo "|| DASHBOARD DE BACKUPS AUTOMÁTICOS S3 ||" echo

"I'm not going to let you go," she said, her voice low and steady. "You're mine now."

```
echo ""
```

```
echo Estado actual: $(date)"
```

```
echo ""
```

```
echo "└─ CONFIGURACIÓN CRON
```

\_\_\_\_\_

```
crontab -l | grep backup-cron | sed 's/^/ /'
```

```
if [ $? -ne 0 ]; then
```

```
echo " No hay tareas configuradas"
```

fi

echo

```

" |
"
echo ""

echo " └─ ÚLTIMOS BACKUPS EJECUTADOS
" | if ls $LOG_DIR/cron_backup_*.log 1> /dev/null
2>&1; then
    ls -lht $LOG_DIR/cron_backup_*.log 2>/dev/null | head -3 | \
    awk '{print " | " $6, $7, $8, $9}'
else
    echo " | Esperando primera ejecución..."
fi
echo
" |
"
echo ""

echo " └─ ESTADÍSTICAS S3 "
TOTAL=$(aws s3 ls s3://$BUCKET_NAME/backups/ --recursive 2>/dev/null | wc -l) echo " |
Total de archivos en S3: $TOTAL"
echo
" |
"
echo ""

echo " └─ ARCHIVOS LOCALES
" |
LOCAL_COUNT=$(ls -1 $HOME/proyectos/backup-s3/datos/ | wc -l)
echo " | Total de archivos locales: $LOCAL_COUNT"
ls -lh $HOME/proyectos/backup-s3/datos/ | tail -n +2 | awk '{print " | " $9 " (" $5
")"}' echo
" |
"
echo ""

echo " └─ ÚLTIMO LOG COMPLETO
" |
if ls $LOG_DIR/cron_backup_*.log 1> /dev/null 2>&1; then
    tail -15 $(ls -t $LOG_DIR/cron_backup_*.log 2>/dev/null | head -1) 2>/dev/null |
    \ sed 's/^/ | /'
else
    echo " | Sin logs disponibles aún"
fi
echo
" |
"
EOF

```

```

root@stantiago-VirtualBox:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/dashboard.sh <<'EOF'
> #!/bin/bash
BUCKET_NAME="backup-automatico-santiago"
LOG_DIR="$HOME/proyectos/backup-s3/logs"
clear
echo
"===== 1"
echo "|| DASHBOARD DE BACKUPS AUTOMÁTICOS S3 ||" echo
"===== 2"
echo ""
echo Estado actual: $(date)"
echo ""
echo "└─ CONFIGURACIÓN CRON"
"
crontab -l | grep backup-cron | sed 's/^/| /'
if [ $? -ne 0 ]; then
echo "| No hay tareas configuradas"
fi
echo
"|"
echo ""
echo "└─ ÚLTIMOS BACKUPS EJECUTADOS" if ls $LOG_DIR/cron_backup_*.log 1> /dev/null 2>&1; then
ls -lht $LOG_DIR/cron_backup_*.log 2>/dev/null | head -3 | \
awk '{print "| " $6, $7, $8, $9}'
else
echo "| Esperando primera ejecución..."
fi
echo
"|"
echo ""
echo "└─ ESTADÍSTICAS S3" TOTAL=$(aws s3 ls s3:/
/$BUCKET_NAME/backups/ --recursive 2>/dev/null | wc -l) echo "| Total de archivos en S3
: $TOTAL"
echo

```

Edita el script para reemplazar [tu-nombre]:

```
nano ~/proyectos/backup-s3/scripts/dashboard.sh
```

```

GNU nano 7.2 /root/proyectos/backup-s3/scripts/dashboard.sh
#!/bin/bash
BUCKET_NAME="backup-automático-santiago"
LOG_DIR="$HOME/proyectos/backup-s3/logs"
clear
echo
"===== 1"
echo "|| DASHBOARD DE BACKUPS AUTOMÁTICOS S3 ||" echo
"===== 2"
echo ""
echo Estado actual: $(date)"
echo ""
echo "└─ CONFIGURACIÓN CRON"
"
crontab -l | grep backup-cron | sed 's/^/ /'
if [ $? -ne 0 ]; then
echo "| No hay tareas configuradas"
fi
echo
"===== 3"
echo ""
echo "└─ ÚLTIMOS BACKUPS EJECUTADOS" if ls $LOG_DIR/cron_b>
ls -lht $LOG_DIR/cron_backup_*.log 2>/dev/null | head -3 | \
awk '{print "| " $6, $7, $8, $9}'
else
echo "| Esperando primera ejecución..."
fi
echo
"===== 4"
echo ""
echo "└─ ESTADÍSTICAS S3" TOTAL=$(aws s3 ls s3:>
echo
"===== 5"
echo ""
echo "└─ ARCHIVOS LOCALES"
"
[ 49 líneas leídas ]

```

Dar permisos y ejecutar:

```
chmod +x ~/proyectos/backup-s3/scripts/dashboard.sh
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# chmod +x ~/proyectos/backup-s3/scripts/dashboard.sh
```

```
~/proyectos/backup-s3/scripts/dashboard.sh
```

```

/root/proyectos/backup-s3/scripts/dashboard.sh: línea 6:
||: orden no encontrada
|| DASHBOARD DE BACKUPS AUTOMÁTICOS S3 || echo
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 8:
||: orden no encontrada

Estado actual: vie 30 ene 2026 10:58:49 CET
echo
echo └─ CONFIGURACIÓN CRON
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 16:
crontab -l | grep backup-cron | sed 's/^/| /'
if [ 0 -ne 0 ]; then
echo |: No existe el archivo o el directorio
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 38: error sintáctico cerca del el
emento inesperado '('
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 38: `ls -lh $HOME/proyectos/backu
p-s3/datos/ | tail -n +2 | awk '{print "| " $9 " (" $5 ")"}' echo '
root@santiago-VirtualBox:~/proyectos/backup-s3#

```

## PARTE 11: Script de limpieza de S3

Crea limpiar-s3.sh:

```
cat > ~/proyectos/backup-s3/scripts/limpiar-s3.sh <<'EOF'
```

Y dentro:

```
#!/bin/bash
```

```
BUCKET_NAME="backup-automatico-santiago"
```

```
LOG_FILE="$HOME/proyectos/backup-s3/logs/limpieza_$(date
+%Y%m%d_%H%M%S).log"
```

```
echo "=== Limpieza de backups en S3 ===" | tee -a $LOG_FILE
```

```
echo "Fecha: $(date)" | tee -a $LOG_FILE
```

```
echo "" | tee -a $LOG_FILE
```

```
echo "Listando todas las versiones en S3:" | tee -a $LOG_FILE aws s3 ls
```

```
s3://$BUCKET_NAME/backups/ --recursive | tee -a $LOG_FILE
```

```
echo "" | tee -a $LOG_FILE
```

```
echo "Tamaño total:" | tee -a $LOG_FILE
```

```
aws s3 ls s3://$BUCKET_NAME --recursive --summarize | grep "Total Size" | tee -a
$LOG_FILE
```

```
echo "" | tee -a $LOG_FILE
```

```
echo "Para eliminar archivos antiguos manualmente, usa:" | tee -a $LOG_FILE
```

```
echo "aws s3 rm s3://$BUCKET_NAME/backups/version_YYYYMMDD_HHMMSS/
--recursive" \
```

```
| tee -a $LOG_FILE
```

```
echo "" | tee -a $LOG_FILE
```

```
echo "=== Análisis completado ===" | tee -a $LOG_FILE
```

```
EOF
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/limpiar-s3.sh <<'EOF'
> #!/bin/bash
BUCKET_NAME="backup-automatico-santiago"
LOG_FILE="$HOME/proyectos/backup-s3/logs/limpieza_$(date +%Y%m%d_%H%M%S).log"
echo "=== Limpieza de backups en S3 ===" | tee -a $LOG_FILE echo "Fecha: $(date)" | tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "Listando todas las versiones en S3:" | tee -a $LOG_FILE aws s3 ls s3://$BUCKET_NAME/backups/ --recursive | tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "Tamaño total:" | tee -a $LOG_FILE
aws s3 ls s3://$BUCKET_NAME --recursive --summarize | grep "Total Size" | tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "Para eliminar archivos antiguos manualmente, usa:" | tee -a $LOG_FILE echo "aws s3 rm s3://$BUCKET_NAME/backups/version_YYYYMMDD_HHMMSS/ --recursive" \
| tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "=== Análisis completado ===" | tee -a $LOG_FILE
> EOF
```

Edita el script para reemplazar [tu-nombre]:

```
nano ~/proyectos/backup-s3/scripts/limpiar-s3.sh
```

```
GNU nano 7.2 /root/proyectos/backup-s3/scripts/limpiar-s3.sh
#!/bin/bash
BUCKET_NAME="backup-automatico-santiago"
LOG_FILE="$HOME/proyectos/backup-s3/logs/limpieza_$(date +%Y%m%d_%H%M%S).log"
echo "=== Limpieza de backups en S3 ===" | tee -a $LOG_FILE echo "Fecha: $(date)" | tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "Listando todas las versiones en S3:" | tee -a $LOG_FILE aws s3 ls s3://$BUCKET_NAME/backups/ --recursive | tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "Tamaño total:" | tee -a $LOG_FILE
aws s3 ls s3://$BUCKET_NAME --recursive --summarize | grep "Total Size" | tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "Para eliminar archivos antiguos manualmente, usa:" | tee -a $LOG_FILE echo "aws s3 rm s3://$BUCKET_NAME/backups/version_YYYYMMDD_HHMMSS/ --recursive" \
| tee -a $LOG_FILE
echo "" | tee -a $LOG_FILE
echo "=== Análisis completado ===" | tee -a $LOG_FILE
```

Dar permisos y ejecutar:

```
chmod +x ~/proyectos/backup-s3/scripts/limpiar-s3.sh
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# chmod +x ~/proyectos/backup-s3/scripts/limpiar-s3.sh
```

```
~/proyectos/backup-s3/scripts/limpiar-s3.sh
```

```

root@santiago-VirtualBox:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/limpiar-s3.sh
=== Limpieza de backups en S3 ===

tee: unrecognized option '--recursive'
Pruebe 'tee --help' para más información.

Tamaño total:
  Total Size: 0

tee: 'aws s3 rm s3://backup-automatico-santiago/backups/version_YYYYMMDD_HHMMSS/ --recursive': No existe el archivo o el directorio
Para eliminar archivos antiguos manualmente, usa:
/root/proyectos/backup-s3/scripts/limpiar-s3.sh: línea 12: error sintáctico cerca del elemento inesperado `|'
/root/proyectos/backup-s3/scripts/limpiar-s3.sh: línea 12: `| tee -a $LOG_FILE '

```

## PARTE 12: Prueba de restauración completa

### Simular pérdida de datos

```
echo "Simulando pérdida de datos..."
```

```
rm ~/proyectos/backup-s3/datos/*.txt
```

```

root@santiago-VirtualBox:~/proyectos/backup-s3# echo "Simulando pérdida de datos..."
rm ~/proyectos/backup-s3/datos/*.txt
Simulando pérdida de datos...

```

```
ls ~/proyectos/backup-s3/datos/
```

```

root@santiago-VirtualBox:~/proyectos/backup-s3# ls ~/proyectos/backup-s3/datos/
config.json  usuarios.csv

```

### Restaurar desde S3

```
~/proyectos/backup-s3/scripts/restaurar.sh
```

```

root@santiago-VirtualBox:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/restaurar.sh
=== Iniciando restauración: vie 30 ene 2026 11:02:02 CET ===
Versiones disponibles:

Unknown options:
tee: opción incorrecta -- «l»
Pruebe 'tee --help' para más información.
=== Restauración completada ===

```

### Copiar archivos restaurados al directorio original `cp`

```
~/proyectos/backup-s3/restaurados/* ~/proyectos/backup-s3/datos/
```

```
ls -lh ~/proyectos/backup-s3/datos/
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# ls -lh ~/proyectos/backup-s3/datos/
total 4,0K
-rw-r--r-- 1 root root 30 ene 30 09:40 config.json
-rw-r--r-- 1 root root  0 ene 30 09:43 usuarios.csv
```

```
cat ~/proyectos/backup-s3/datos/documento1.txt
```

## PARTE 13: Gestión del cron

### Ver configuración actual

```
crontab -l
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
```

## Pausar temporalmente el cron

`crontab -e`

Añade # al inicio de la línea:

```
# */3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
```

```
GNU nano 7.2 /tmp/crontab.aV5n4z/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
#*/3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
```

Guarda y sal.

## Reactivar el cron

```
crontab -e
```

Elimina el # del inicio:

```
*/3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
GNU nano 7.2 /tmp/crontab.SsGcrd/crontab
## Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
```

## Eliminar completamente el cron

```
crontab -r
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# crontab -r
```

Verifica que se eliminó:

```
crontab -l
```

## PARTE 14: Verificación final completa

### Ejecutar todos los verificadores

```
echo "=== VERIFICACIÓN COMPLETA DEL SISTEMA ==="
echo ""
echo "1. Configuración de cron:"
crontab -l
echo ""
echo "2. Logs generados:"
ls -lh ~/proyectos/backup-s3/logs/
echo ""
echo "3. Archivos en S3:"
aws s3 ls s3://backup-automatico-[tu-nombre]/backups/ --recursive
echo ""
echo "4. Dashboard completo:"
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# echo "=== VERIFICACIÓN COMPLETA DEL SIS
TEMA ===" echo ""
echo "1. Configuración de cron:"
crontab -l
echo ""
echo "2. Logs generados:"
ls -lh ~/proyectos/backup-s3/logs/
echo ""
echo "3. Archivos en S3:"
aws s3 ls s3://backup-automatico-santiago/backups/ --recursive echo ""
echo "4. Dashboard completo:"
=== VERIFICACIÓN COMPLETA DEL SISTEMA === echo
1. Configuración de cron:
no crontab for root

2. Logs generados:
total 4,0K
-rw-r--r-- 1 root root 153 ene 30 11:00 limpieza_20260130_110037.log

3. Archivos en S3:

Unknown options: echo,
4. Dashboard completo:
```

~/proyectos/backup-s3/scripts/[dashboard.sh](#)

```
root@santiago-VirtualBox: ~/proyectos/backup-s3

/root/proyectos/backup-s3/scripts/dashboard.sh: línea 6:
=====
||: orden no encontrada
|| DASHBOARD DE BACKUPS AUTOMÁTICOS S3 || echo
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 8:
=====
||: orden no encontrada

Estado actual: vie 30 ene 2026 11:06:48 CET
echo
echo └ CONFIGURACIÓN CRON
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 16:
=====
crontab -l | grep backup-cron | sed 's/^/| /'
if [ 0 -ne 0 ]; then
echo |: No existe el archivo o el directorio
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 38: error sintáctico cerca del el
emento inesperado `('
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 38: `ls -lh $HOME/proyectos/backu
p-s3/datos/ | tail -n +2 | awk '{print "| " $9 " (" $5 ")"}' echo '
root@santiago-VirtualBox:~/proyectos/backup-s3#
```

## ENTREGABLES

Captura de cada proceso que realizáis con una breve explicación, sobre todo las más importantes:

### 1. Captura de la estructura de directorios

tree ~/proyectos/backup-s3/

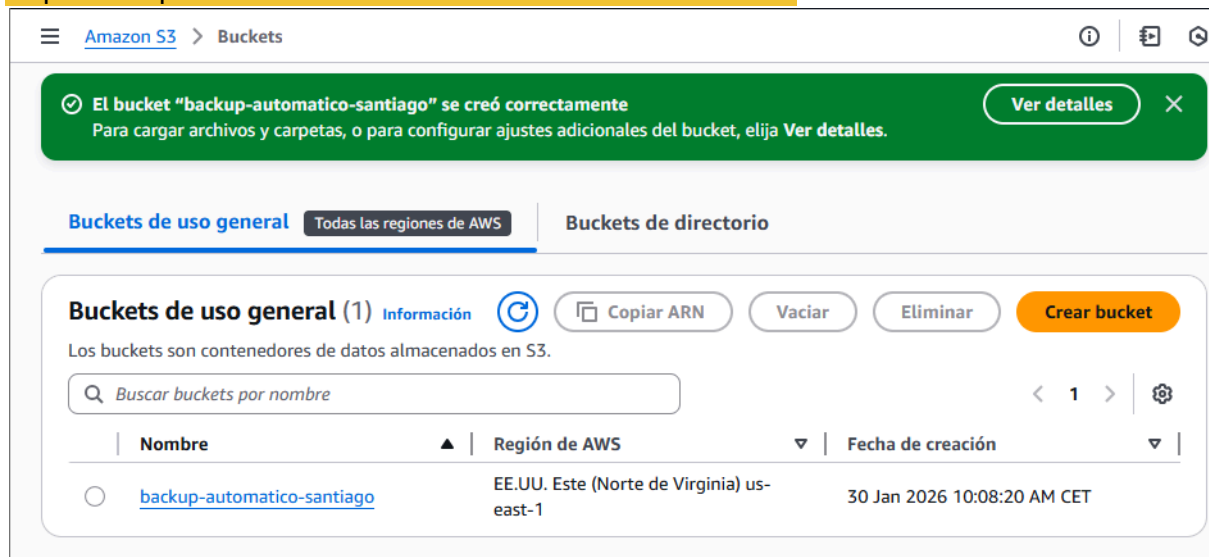
```
root@santiago-VirtualBox:~/proyectos/backup-s3# tree ~/proyectos/backup-s3/
/root/proyectos/backup-s3/
├── datos
│   ├── config.json
│   └── usuarios.csv
├── echo
├── else
├── Fecha: vie 30 ene 2026 11:00:37 CET
├── logs
│   └── limpieza_20260130_110037.log
├── restaurados
├── scripts
│   ├── backup-cron.sh
│   ├── backup-manual.sh
│   ├── backup-versionado.sh
│   ├── dashboard.sh
│   ├── limpiar-s3.sh
│   ├── monitor.sh
│   └── restaurar.sh
5 directories, 14 files
```

O si no tienes tree:

```
find ~/proyectos/backup-s3/ -type f
```

## 2. Evidencia del bucket S3 creado

Captura de pantalla de la consola AWS mostrando el bucket.



## 3. Configuración de crontab

```
crontab -l > ~/proyectos/backup-s3/crontab_config.txt  
cat ~/proyectos/backup-s3/crontab_config.txt
```

## 4. Logs de ejecuciones automáticas

```
ls -lh ~/proyectos/backup-s3/logs/cron_backup_*.log
```

Captura mostrando al menos 3 logs diferentes.

## 5. Contenido de un log completo

```
cat ~/proyectos/backup-s3/logs/cron_backup_*.log | head -50
```

## 6. Verificación de archivos en S3

```
aws s3 ls s3://backup-automatico-[tu-nombre]/ --recursive --human-readable  
root@santiago-VirtualBox:~/proyectos/backup-s3# aws s3 ls s3://backup-automatico-santiago/ --recursive --human-readable
```

## 7. Dashboard de estado

```
~/proyectos/backup-s3/scripts/dashboard.sh
```

```
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 6:   
||: orden no encontrada  
|| DASHBOARD DE BACKUPS AUTOMÁTICOS S3 || echo  
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 8:   
||: orden no encontrada  
  
Estado actual: vie 30 ene 2026 11:10:02 CET  
echo  
echo └─ CONFIGURACIÓN CRON  
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 16:   
└─  
crontab -l | grep backup-cron | sed 's/^/| /'  
if [ 0 -ne 0 ]; then  
echo |: No existe el archivo o el directorio  
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 38: error sintáctico cerca del el  
emento inesperado `(`  
/root/proyectos/backup-s3/scripts/dashboard.sh: línea 38: `ls -lh $HOME/proyectos/backu  
p-s3/datos/ | tail -n +2 | awk '{print "| " $9 " (" $5 ")"}' echo '  
|
```

Captura de pantalla del resultado.

## 8. Prueba de restauración

```
ls -lh ~/proyectos/backup-s3/restaurados/
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# ls -lh ~/proyectos/backup-s3/restaurado  
s/  
total 0
```

```
cat ~/proyectos/backup-s3/restaurados/documento1.txt
```

## 9. Logs del sistema cron

```
grep CRON /var/log/syslog | grep backup-cron | tail -10
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# grep CRON /var/log/syslog | grep backup  
-cron | tail -10  
2026-01-30T10:42:01.018724+01:00 santiago-VirtualBox CRON[5473]: (root) CMD ($HOME/proy  
ectos/backup-s3/scripts/backup-cron.sh)  
2026-01-30T10:45:01.057326+01:00 santiago-VirtualBox CRON[5564]: (root) CMD ($HOME/proy  
ectos/backup-s3/scripts/backup-cron.sh)  
2026-01-30T10:48:01.083433+01:00 santiago-VirtualBox CRON[5786]: (root) CMD ($HOME/proy  
ectos/backup-s3/scripts/backup-cron.sh)  
2026-01-30T10:51:01.108841+01:00 santiago-VirtualBox CRON[5934]: (root) CMD ($HOME/proy  
ectos/backup-s3/scripts/backup-cron.sh)  
2026-01-30T10:54:01.134696+01:00 santiago-VirtualBox CRON[5945]: (root) CMD ($HOME/proy  
ectos/backup-s3/scripts/backup-cron.sh)  
2026-01-30T10:57:01.159451+01:00 santiago-VirtualBox CRON[6055]: (root) CMD ($HOME/proy  
ectos/backup-s3/scripts/backup-cron.sh)  
2026-01-30T11:00:01.180555+01:00 santiago-VirtualBox CRON[6121]: (root) CMD ($HOME/proy  
ectos/backup-s3/scripts/backup-cron.sh)  
2026-01-30T11:03:01.193542+01:00 santiago-VirtualBox CRON[6230]: (root) CMD ($HOME/proy  
ectos/backup-s3/scripts/backup-cron.sh)
```

# Limpieza final

## Al terminar la práctica, elimina los recursos

# Detener el cron

```
crontab -r
```

# Eliminar archivos de S3

```
aws s3 rm s3://backup-automatico-[tu-nombre]/ --recursive
```

# Verificar que S3 está vacío

```
aws s3 ls s3://backup-automatico-[tu-nombre]/
```

```
root@santiago-VirtualBox:~/proyectos/backup-s3# aws s3 ls s3://backup-automatico-santiago/
```