

## PARTE 0: Preparación del entorno

Verificar AWS CLI;

```
root@UbuntuPL:/home/juan# aws --version
aws-cli/2.32.15 Python/3.13.11 Linux/6.14.0-37-generic exe/x86_64/ubuntu.24
root@UbuntuPL:/home/juan# aws configure list
NAME      : VALUE              : TYPE      : LOCATION
profile   : <not set>          : None     : None
access_key : ****DJBG          : shared-credentials-file :
secret_key : *****eDdA          : shared-credentials-file :
region    : us-east-1          : config-file : ~/.aws/config
root@UbuntuPL:/home/juan#
```

Crear estructura de directorios;

```
root@UbuntuPL:/home/juan# mkdir -p ~/proyectos/backup-s3
root@UbuntuPL:/home/juan# mkdir -p ~/proyectos/backup-s3/datos
root@UbuntuPL:/home/juan# mkdir -p ~/proyectos/backup-s3/logs
root@UbuntuPL:/home/juan# mkdir -p ~/proyectos/backup-s3/scripts
root@UbuntuPL:/home/juan# cd ~/proyectos/backup-s3
root@UbuntuPL:~/proyectos/backup-s3# ls
datos  logs  scripts
root@UbuntuPL:~/proyectos/backup-s3#
```

Crear archivos de prueba;

```
root@UbuntuPL:~/proyectos/backup-s3# echo "Documento importante v1" > ~/proyectos/backup-s3/datos/documento1.txt
root@UbuntuPL:~/proyectos/backup-s3# echo "Configuracion de aplicacion" > ~/proyectos/backup-s3/datos/config.json
root@UbuntuPL:~/proyectos/backup-s3# echo "Datos de usuario" > ~/proyectos/backup-s3/datos/usuarios.csv
root@UbuntuPL:~/proyectos/backup-s3# date > ~/proyectos/backup-s3/datos/timestamp.txt
root@UbuntuPL:~/proyectos/backup-s3#
```

Verificar estructura creada;

```

root@UbuntuPL:~/proyectos/backup-s3# ls -la ~/proyectos/backup-s3
total 20
drwxr-xr-x 5 root root 4096 Jan 30 08:10 .
drwxr-xr-x 3 root root 4096 Jan 30 08:10 ..
drwxr-xr-x 2 root root 4096 Jan 30 08:16 datos
drwxr-xr-x 2 root root 4096 Jan 30 08:10 logs
drwxr-xr-x 2 root root 4096 Jan 30 08:10 scripts
root@UbuntuPL:~/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
root@UbuntuPL:~/proyectos/backup-s3#

```

## PARTE 1: Crear el bucket S3 desde AWS

Bucket name		Info	
	Name	AWS Region	Creation date
<input checked="" type="radio"/>	<a href="#">backup-automatico-juan</a>	US East (N. Virginia) us-east-1	January 30, 2026, 08:35:21 (UTC+00:00)

Verificar acceso desde la VM;

```

root@UbuntuPL:~/proyectos/backup-s3# aws configure
AWS Access Key ID [*****]: ASIARV3UIQR6GELDTNLT
AWS Secret Access Key [*****]: dN/3yeB0SEp9m1CVruSxVwBaGdVzeuoMv2922iF4
AWS Session Token [None]: I0oJb3JpZluX2VjENH//////////wEaCXVzLXd1c3QtMiJHMEUCIBXAAKCJX1A32geW97c+CEwQoQcxBynGEwr1Jh2bpA
<Ahzm33QuPyWUJjBkWggSRrOXV2ASKBz8Cxzu4WGUMe8KAi+NYYxvAFSNiJjfMPPzFXEr7pwStFW7Tw==
Default region name [juanz]: us-east-1
Default output format [juanz]: json
root@UbuntuPL:~/proyectos/backup-s3#

```

```

root@UbuntuPL:~/proyectos/backup-s3# aws s3 ls s3://backup-automatico-juan

```

## PARTE 2: Script de backup manual

Crea el script backup-manual.sh:

```

root@UbuntuPL:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/backup-manual.sh <<'EOF'
>#!/bin/bash
# Configuración
BUCKET_NAME="backup-automatico-[tu-nombre]"
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
LOG_FILE="$HOME/proyectos/backup-s3/logs/backup_$TIMESTAMP.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

```

Edita el script y reemplaza [tu-nombre] con tu nombre real;

```

GNU nano 7.2                                     /root/proyectos/backup-s3/scripts/b
#!/bin/bash
# Configuración
BUCKET_NAME="backup-automatico-juan"
LOCAL_DIR="$HOME/proyectos/backup-s3/datos"
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
LOG_FILE="$HOME/proyectos/backup-s3/logs/backup_$TIMESTAMP.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

```

Dar permisos de ejecución;

```

root@UbuntuPL:~/proyectos/backup-s3# chmod +x ~/proyectos/backup-s3/scripts/backup-manual.sh

```

Ejecutar el backup manual;

```
root@UbuntuPL:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/backup-manual.sh
== Inicio del backup: Friday, January 30, 2026 AM09:09:53 UTC ==
upload: datos/documento1.txt to s3://backup-automatico-juan/backups/documento1.txt
upload: datos/usuarios.csv to s3://backup-automatico-juan/backups/usuarios.csv
upload: datos/config.json to s3://backup-automatico-juan/backups/config.json
upload: datos/timestamp.txt to s3://backup-automatico-juan/backups/timestamp.txt
✓ Backup completado exitosamente

--- Archivos en S3 ---
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
== Fin del backup: Friday, January 30, 2026 AM09:09:55 UTC ==
root@UbuntuPL:~/proyectos/backup-s3#
```

Verificar el resultado;

```
root@UbuntuPL:~/proyectos/backup-s3# cat ~/proyectos/backup-s3/logs/backup_*.log
== Inicio del backup: Friday, January 30, 2026 AM08:55:13 UTC ==
fatal error: Could not connect to the endpoint URL: "https://backup-automatico-juan.s3.juanz.amazonaws.com/?list-type=2&prefix=backups%2F&encoding-type=url"
✓ Backup completado exitosamente

--- Archivos en S3 ---
== Fin del backup: Friday, January 30, 2026 AM08:55:18 UTC ==
== Inicio del backup: Friday, January 30, 2026 AM09:09:53 UTC ==
upload: datos/documento1.txt to s3://backup-automatico-juan/backups/documento1.txt
upload: datos/usuarios.csv to s3://backup-automatico-juan/backups/usuarios.csv
upload: datos/config.json to s3://backup-automatico-juan/backups/config.json
upload: datos/timestamp.txt to s3://backup-automatico-juan/backups/timestamp.txt
✓ Backup completado exitosamente

--- Archivos en S3 ---
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
== Fin del backup: Friday, January 30, 2026 AM09:09:55 UTC ==
root@UbuntuPL:~/proyectos/backup-s3#
```

```
root@UbuntuPL:~/proyectos/backup-s3# aws s3 ls s3://backup-automatico-juan/backups/ --recursive
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
```

PARTE 3: Script de backup con versionado

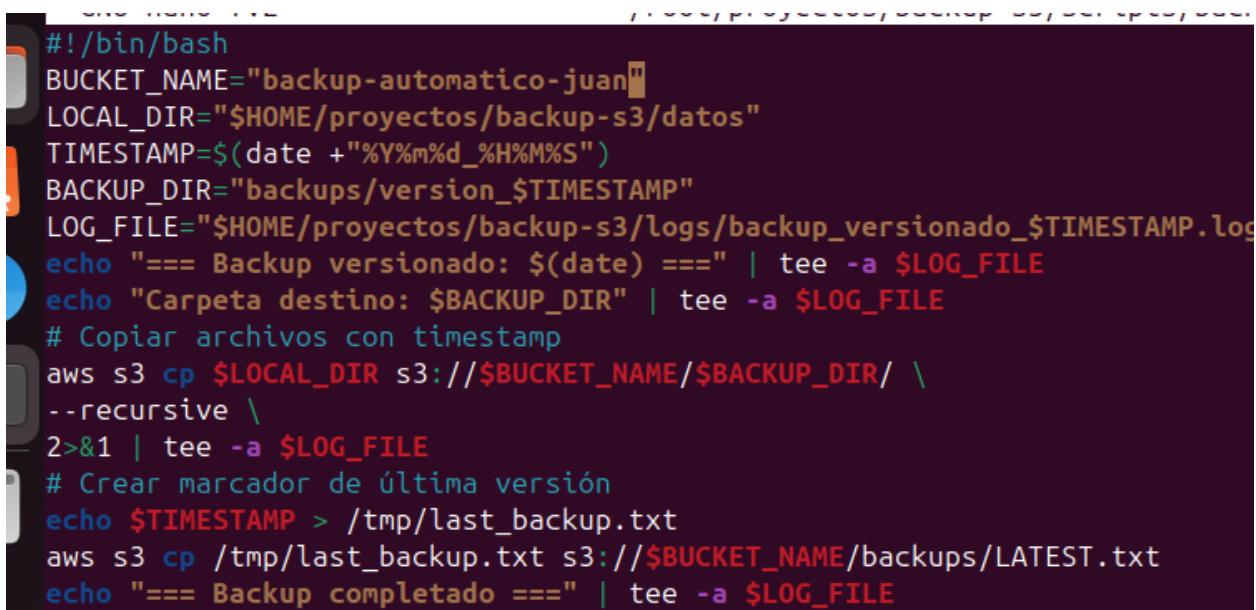
Crea backup-versionado.sh;

```

root@UbuntuPL:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/backup-versionado.sh <<'EOF'
>#!/bin/bash
BUCKET_NAME="backup-automatico-[tu-nombre]"
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];



```

#!/bin/bash
BUCKET_NAME="backup-automatico-juan"
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;

```

root@UbuntuPL:~/proyectos/backup-s3# chmod +x ~/proyectos/backup-s3/scripts/backup-versionado.sh
root@UbuntuPL:~/proyectos/backup-s3# ./proyectos/backup-s3/scripts/backup-versionado.sh
==> Backup versionado: Friday, January 30, 2026 AM09:15:39 UTC ==
Carpeta destino: backups/version_20260130_091539
upload: datos/documento1.txt to s3://backup-automatico-juan/backups/version_20260130_091539/documento1.txt
upload: datos/timestamp.txt to s3://backup-automatico-juan/backups/version_20260130_091539/timestamp.txt
upload: datos/config.json to s3://backup-automatico-juan/backups/version_20260130_091539/config.json
upload: datos/usuarios.csv to s3://backup-automatico-juan/backups/version_20260130_091539/usuarios.csv
upload: ../../tmp/last_backup.txt to s3://backup-automatico-juan/backups/LATEST.txt
==> Backup completado ==

```

Verificar versiones creadas;

```
root@UbuntuPL:~/proyectos/backup-s3# aws s3 ls s3://backup-automatico-juan/backups/
                           PRE version_20260130_091539/
2026-01-30 09:15:41      16 LATEST.txt
2026-01-30 09:09:55      28 config.json
2026-01-30 09:09:55      24 documento1.txt
2026-01-30 09:09:55      40 timestamp.txt
2026-01-30 09:09:55      17 usuarios.csv
root@UbuntuPL:~/proyectos/backup-s3#
```

## PARTE 4: Script de restauración

Crea restaurar.sh;

```
root@UbuntuPL:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/restaurar.sh <<'EOF'
>#!/bin/bash
BUCKET_NAME="backup-automatico-[tu-nombre]"
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@UbuntuPL:~/proyectos/backup-s3#
```

Edita el script para reemplazar [tu-nombre];

```
GNU nano 7.2                                     /root/proyectos/backup-s3/scripts/restaurar.sh *
#!/bin/bash
BUCKET_NAME="backup-automatico-juan"
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
```

Dar permisos;

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

Probar restauración;

```
root@UbuntuPL:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/restaurar.sh
== Iniciando restauración: Friday, January 30, 2026 AM09:19:00 UTC ==
Versiones disponibles:
PRE version_20260130_091539/
2026-01-30 09:15:41      16 LATEST.txt
2026-01-30 09:09:55      28 config.json
2026-01-30 09:09:55      24 documento1.txt
2026-01-30 09:09:55      40 timestamp.txt
2026-01-30 09:09:55      17 usuarios.csv
download: s3://backup-automatico-juan/backups/documento1.txt to restaurados/documento1.txt
download: s3://backup-automatico-juan/backups/version_20260130_091539/config.json to restaurados/version_20260130_091539/config.json
download: s3://backup-automatico-juan/backups/config.json to restaurados/config.json
download: s3://backup-automatico-juan/backups/timestamp.txt to restaurados/timestamp.txt
download: s3://backup-automatico-juan/backups/usuarios.csv to restaurados/usuarios.csv
download: s3://backup-automatico-juan/backups/LATEST.txt to restaurados/LATEST.txt
download: s3://backup-automatico-juan/backups/version_20260130_091539/documento1.txt to restaurados/version_20260130_091539/documento1.txt
download: s3://backup-automatico-juan/backups/version_20260130_091539/timestamp.txt to restaurados/version_20260130_091539/timestamp.txt
download: s3://backup-automatico-juan/backups/version_20260130_091539/usuarios.csv to restaurados/version_20260130_091539/usuarios.csv
Archivos restaurados en: /root/proyectos/backup-s3/restaurados
total 24K
-rw-r--r-- 1 root root  28 Jan 30 09:09 config.json
-rw-r--r-- 1 root root  24 Jan 30 09:09 documento1.txt
-rw-r--r-- 1 root root  16 Jan 30 09:15 LATEST.txt
-rw-r--r-- 1 root root  40 Jan 30 09:09 timestamp.txt
-rw-r--r-- 1 root root  17 Jan 30 09:09 usuarios.csv
drwxr-xr-x 2 root root 4.0K Jan 30 09:19 version_20260130_091539
== Restauración completada ==
root@UbuntuPL:~/proyectos/backup-s3#
```

```
root@UbuntuPL:~/proyectos/backup-s3# ls -la ~/proyectos/backup-s3/restaurados/
total 32
drwxr-xr-x 3 root root 4096 Jan 30 09:19 .
drwxr-xr-x 6 root root 4096 Jan 30 09:19 ..
-rw-r--r-- 1 root root  28 Jan 30 09:09 config.json
-rw-r--r-- 1 root root  24 Jan 30 09:09 documento1.txt
-rw-r--r-- 1 root root  16 Jan 30 09:15 LATEST.txt
-rw-r--r-- 1 root root  40 Jan 30 09:09 timestamp.txt
-rw-r--r-- 1 root root  17 Jan 30 09:09 usuarios.csv
drwxr-xr-x 2 root root 4096 Jan 30 09:19 version_20260130_091539
root@UbuntuPL:~/proyectos/backup-s3#
```

```
root@UbuntuPL:~/proyectos/backup-s3# cat ~/proyectos/backup-s3/restaurados/documento1.txt
Documento importante v1
root@UbuntuPL:~/proyectos/backup-s3#
```

## PARTE 5: Script optimizado para cron

Crea backup-cron.sh;

```
root@UbuntuPL:~/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-automatico-[tu-nombre]"
> 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@UbuntuPL:~/proyectos/backup-s3#
```

Edita el script para reemplazar [tu-nombre];

```
GNU nano 7.2                                     /root/proyectos/backup-s3/scr
#!/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-juan"
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)
```

Dar permisos de ejecución;

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

Probar el script manualmente;

```
root@UbuntuPL:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/backup-cron.sh
root@UbuntuPL:~/proyectos/backup-s3# cat ~/proyectos/backup-s3/logs/cron_backup_*.log
=====
Backup automático iniciado
Fecha: Friday, January 30, 2026 AM09:22:40 UTC
=====
delete: s3://backup-automatico-juan/backups/LATEST.txt
delete: s3://backup-automatico-juan/backups/version_20260130_091539/config.json
delete: s3://backup-automatico-juan/backups/version_20260130_091539/timestamp.txt
delete: s3://backup-automatico-juan/backups/version_20260130_091539/usuarios.csv
delete: s3://backup-automatico-juan/backups/version_20260130_091539/documento1.txt
✓ Backup exitoso
Archivos sincronizados:
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
=====
Backup finalizado: Friday, January 30, 2026 AM09:22:43 UTC
=====
=====
Backup automático iniciado
Fecha: Friday, January 30, 2026 AM09:23:04 UTC
=====
✓ Backup exitoso
Archivos sincronizados:
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
=====
Backup finalizado: Friday, January 30, 2026 AM09:23:06 UTC
```

## PARTE 6: Configurar crontab para ejecución automática

Editar el crontab del usuario;

```
root@UbuntuPL:~/proyectos/backup-s3# crontab -e
```

Añadir la tarea programada;

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

Verificar que cron está configurado;

```
root@UbuntuPL:~/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
# 15 2 * * * /usr/local/bin/backup_web.sh
# 0 3 * * * /home/juan/script_backup.sh
# */3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
root@UbuntuPL:~/proyectos/backup-s3#
```

## PARTE 7: Monitorización en tiempo real

Crea monitor.sh;

```

root@UbuntuPL:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/monitor.sh <<'EOF'
>#!/bin/bash
BUCKET_NAME="backup-automatico-[tu-nombre]"
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];

```

GNU nano 7.2                                     /root/proyectos/backup-s3/scripts/monitor.sh
#!/bin/bash
BUCKET_NAME="backup-automatico-juan"
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 "=====

```

Dar permisos;

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

Ejecutar el monitor;

```
root@UbuntuPL:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/monitor.sh
=====
MONITOR DE BACKUPS AUTOMÁTICOS
=====

Últimos 5 logs generados:
-rw-r--r-- 1 root root 551 Jan 30 09:23 /root/proyectos/backup-s3/logs/cron_backup_20260130_092304.log
-rw-r--r-- 1 root root 1.2K Jan 30 09:22 /root/proyectos/backup-s3/logs/cron_backup_20260130_092240.log

Contenido del último backup:
=====
Backup automático iniciado
Fecha: Friday, January 30, 2026 AM09:23:04 UTC
=====
✓ Backup exitoso
Archivos sincronizados:
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
=====
Backup finalizado: Friday, January 30, 2026 AM09:23:06 UTC
=====

Archivos actuales en S3:
2026-01-30 09:09:55  28 Bytes backups/config.json
2026-01-30 09:09:55  24 Bytes backups/documento1.txt
2026-01-30 09:09:55  40 Bytes backups/timestamp.txt
2026-01-30 09:09:55  17 Bytes backups/usuarios.csv

Tamaño total del bucket:
  Total Size: 109
```

Monitorización continua;

```

Every 30.0s: /root/proyectos/backup-s3/scripts/monitor.sh                                         UbuntuPL: Fri Jan 30 09:31:26 2026

=====
MONITOR DE BACKUPS AUTOMÁTICOS
=====

Últimos 5 logs generados:
-rw-r--r-- 1 root root 551 Jan 30 09:23 /root/proyectos/backup-s3/logs/cron_backup_20260130_092304.log
-rw-r--r-- 1 root root 1.2K Jan 30 09:22 /root/proyectos/backup-s3/logs/cron_backup_20260130_092240.log

Contenido del último backup:
=====
Backup automático iniciado
Fecha: Friday, January 30, 2026 AM09:23:04 UTC
=====
✓ Backup exitoso
Archivos sincronizados:
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
=====
Backup finalizado: Friday, January 30, 2026 AM09:23:06 UTC
=====

Archivos actuales en S3:
2026-01-30 09:09:55  28 Bytes backups/config.json
2026-01-30 09:09:55  24 Bytes backups/documento1.txt
2026-01-30 09:09:55  40 Bytes backups/timestamp.txt
2026-01-30 09:09:55  17 Bytes backups/usuarios.csv

Tamaño total del bucket:

```

## PARTE 8: Verificar ejecuciones de cron

Ver los logs del sistema cron;

```

root@UbuntuPL:~/proyectos/backup-s3# grep CRON /var/log/syslog | tail -20
2026-01-30T08:15:02.057131+00:00 UbuntuPL CRON[4001]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T08:17:01.192100+00:00 UbuntuPL CRON[4017]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
2026-01-30T08:25:01.287222+00:00 UbuntuPL CRON[5642]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T08:30:01.321404+00:00 UbuntuPL CRON[6417]: (root) CMD ([ -x /etc/init.d/anacron ] && if [ ! -d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron start >/dev/null; fi)
2026-01-30T08:35:01.010102+00:00 UbuntuPL CRON[6543]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T08:45:02.507611+00:00 UbuntuPL CRON[8247]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T08:55:01.033287+00:00 UbuntuPL CRON[9328]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T09:05:01.052365+00:00 UbuntuPL CRON[9826]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T09:15:01.090430+00:00 UbuntuPL CRON[10268]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T09:17:01.100144+00:00 UbuntuPL CRON[10399]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
2026-01-30T09:25:01.691210+00:00 UbuntuPL CRON[10789]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2026-01-30T09:30:01.645164+00:00 UbuntuPL CRON[10886]: (root) CMD ([ -x /etc/init.d/anacron ] && if [ ! -d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron start >/dev/null; fi)
root@UbuntuPL:~/proyectos/backup-s3#

```

Listar todos los logs generados;

```
root@UbuntuPL:~/proyectos/backup-s3# ls -lht ~/proyectos/backup-s3/logs/
total 24K
-rw-r--r-- 1 root root 551 Jan 30 09:23 cron_backup_20260130_092304.log
-rw-r--r-- 1 root root 1.2K Jan 30 09:22 cron_backup_20260130_092240.log
-rw-r--r-- 1 root root 2.9K Jan 30 09:19 restore_20260130_091900.log
-rw-r--r-- 1 root root 1.1K Jan 30 09:15 backup_versionado_20260130_091539.log
-rw-r--r-- 1 root root 1.2K Jan 30 09:09 backup_20260130_090953.log
-rw-r--r-- 1 root root 347 Jan 30 08:55 backup_20260130_085513.log
root@UbuntuPL:~/proyectos/backup-s3#
```

Ver contenido de un log específico;

```
=====
Backup automático iniciado
Fecha: Friday, January 30, 2026 AM09:22:40 UTC
=====
delete: s3://backup-automatico-juan/backups/LATEST.txt
delete: s3://backup-automatico-juan/backups/version_20260130_091539/config.json
delete: s3://backup-automatico-juan/backups/version_20260130_091539/timestamp.txt
delete: s3://backup-automatico-juan/backups/version_20260130_091539/usuarios.csv
delete: s3://backup-automatico-juan/backups/version_20260130_091539/documento1.txt
✓ Backup exitoso
Archivos sincronizados:
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
=====
Backup finalizado: Friday, January 30, 2026 AM09:22:43 UTC
=====

=====

Backup automático iniciado
Fecha: Friday, January 30, 2026 AM09:23:04 UTC
=====
✓ Backup exitoso
Archivos sincronizados:
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
=====
Backup finalizado: Friday, January 30, 2026 AM09:23:06 UTC
=====
```

Contar cuántos backups se han ejecutado;

```
root@UbuntuPL:~/proyectos/backup-s3# ls ~/proyectos/backup-s3/logs/cron_backup_*.log | wc -l
2
root@UbuntuPL:~/proyectos/backup-s3#
```

PARTE 9: Simulación de cambios y verificación

Modificar archivos locales;

```
root@UbuntuPL:~/proyectos/backup-s3# echo "Documento actualizado - $(date)" >> ~/proyectos/backup-s3/datos/documento1.txt
root@UbuntuPL:~/proyectos/backup-s3# echo "Nuevo archivo creado automáticamente" > ~/proyectos/backup-s3/datos/nuevo_$(date +%H%M%S).txt
root@UbuntuPL:~/proyectos/backup-s3# echo "Datos adicionales" > ~/proyectos/backup-s3/datos/datos_extra.txt
root@UbuntuPL:~/proyectos/backup-s3#
```

Verificar los cambios localmente;

```
root@UbuntuPL:~/proyectos/backup-s3# ls -lh ~/proyectos/backup-s3/datos/
total 24K
-rw-r--r-- 1 root root 28 Jan 30 08:15 config.json
-rw-r--r-- 1 root root 18 Jan 30 09:35 datos_extra.txt
-rw-r--r-- 1 root root 152 Jan 30 09:34 documento1.txt
-rw-r--r-- 1 root root 38 Jan 30 09:35 nuevo_093500.txt
-rw-r--r-- 1 root root 40 Jan 30 08:16 timestamp.txt
-rw-r--r-- 1 root root 17 Jan 30 08:16 usuarios.csv
root@UbuntuPL:~/proyectos/backup-s3#
```

```
root@UbuntuPL:~/proyectos/backup-s3# cat ~/proyectos/backup-s3/datos/documento1.txt
Documento importante v1
Documento actualizado - Friday, January 30, 2026 AM09:34:05 UTC
Documento actualizado - Friday, January 30, 2026 AM09:34:45 UTC
root@UbuntuPL:~/proyectos/backup-s3#
```

Esperar la siguiente ejecución;

```
=====
MONITOR DE BACKUPS AUTOMÁTICOS
=====

Últimos 5 logs generados:
-rw-r--r-- 1 root root 551 Jan 30 09:23 /root/proyectos/backup-s3/logs/cron_backup_20260130_092304.log
-rw-r--r-- 1 root root 1.2K Jan 30 09:22 /root/proyectos/backup-s3/logs/cron_backup_20260130_092240.log

Contenido del último backup:
=====
Backup automático iniciado
Fecha: Friday, January 30, 2026 AM09:23:04 UTC
=====
✓ Backup exitoso
Archivos sincronizados:
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
=====
Backup finalizado: Friday, January 30, 2026 AM09:23:06 UTC
=====

Archivos actuales en S3:
2026-01-30 09:09:55    28 Bytes backups/config.json
2026-01-30 09:09:55    24 Bytes backups/documento1.txt
2026-01-30 09:09:55    40 Bytes backups/timestamp.txt
2026-01-30 09:09:55    17 Bytes backups/usuarios.csv

Tamaño total del bucket:
Total Size: 109
```

Ver que los nuevos archivos están en S3;

```
root@UbuntuPL:~/proyectos/backup-s3# aws s3 ls s3://backup-automatico-juan/backups/ --recursive
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv
root@UbuntuPL:~/proyectos/backup-s3#
```

## PARTE 10: Dashboard de estado completo

Crea dashboard.sh;

```
root@UbuntuPL:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/dashboard.sh <<'EOF'
>#!/bin/bash
>BUCKET_NAME="backup-automatico-[tu-nombre]"
>LOG_DIR="$HOME/proyectos/backup-s3/logs"
>clear
>echo
>"["
>""
>echo "|| DASHBOARD DE BACKUPS AUTOMÁTICOS S3 ||"
>echo
>""
>""
>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
EOF
```

Edita el script para reemplazar [tu-nombre];

```

GNU nano 7.2                               /root/proyectos/backup-s3/scripts/dashboard.sh
#!/bin/bash
BUCKET_NAME="backup-automatico-juan"
LOG_DIR="$HOME/proyectos/backup-s3/logs"
clear
echo
"[
"
echo "|| DASHBOARD DE BACKUPS AUTOMÁTICOS S3 ||"
echo
"["
]
"
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 "└ 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}'"

```

Dar permisos y ejecutar; (error)

```

|| DASHBOARD DE BACKUPS AUTOMÁTICOS S3 ||
/root/proyectos/backup-s3/scripts/dashboard.sh: line 34: syntax error near unexpected token `('
/root/proyectos/backup-s3/scripts/dashboard.sh: line 34: `ls -lh $HOME/proyectos/backup-s3/datos/ | tail -n +2 | awk '{p
rint "| " $9 " (" $5 ")"}'''
root@UbuntuPL:~/proyectos/backup-s3# 

```

## PARTE 11: Script de limpieza de S3

Crea limpiar-s3.sh;

```

root@UbuntuPL:~/proyectos/backup-s3# cat > ~/proyectos/backup-s3/scripts/limpiar-s3.sh <<'EOF'
>#!/bin/bash
BUCKET_NAME="backup-automatico-[tu-nombre]"
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@UbuntuPL:~/proyectos/backup-s3# 
```

Edita el script para reemplazar [tu-nombre];

```

uno nano ./2
#!/bin/bash
BUCKET_NAME="backup-automatico-juan"
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;

```
root@UbuntuPL:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/limpiar-s3.sh
== Limpieza de backups en S3 ==
Fecha: Friday, January 30, 2026 AM09:51:35 UTC

Listando todas las versiones en S3:
2026-01-30 09:09:55      28 backups/config.json
2026-01-30 09:09:55      24 backups/documento1.txt
2026-01-30 09:09:55      40 backups/timestamp.txt
2026-01-30 09:09:55      17 backups/usuarios.csv

Tamaño total:
 Total Size: 109
/root/proyectos/backup-s3/scripts/limpiar-s3.sh: line 12: /root/proyectos/backup-s3/logs/limpieza_20260130_095135.log: permission denied

Para eliminar archivos antiguos manualmente, usa:
aws s3 rm s3://backup-automatico-juan/backups/version_YYYYMMDD_HHMMSS/
--recursive

== Análisis completado ==
root@UbuntuPL:~/proyectos/backup-s3#
```

## PARTE 12: Prueba de restauración completa

Simular pérdida de datos;

```
root@UbuntuPL:~/proyectos/backup-s3# echo "Simulando pérdida de datos..."
Simulando pérdida de datos...
root@UbuntuPL:~/proyectos/backup-s3# rm ~/proyectos/backup-s3/datos/*.txt
root@UbuntuPL:~/proyectos/backup-s3# ls ~/proyectos/backup-s3/datos/
config.json  usuarios.csv
root@UbuntuPL:~/proyectos/backup-s3#
```

Restaurar desde S3;

```
root@UbuntuPL:~/proyectos/backup-s3# ~/proyectos/backup-s3/scripts/restaurar.sh
== Iniciando restauración: Friday, January 30, 2026 AM09:53:03 UTC ==
Versiones disponibles:
2026-01-30 09:09:55      28 config.json
2026-01-30 09:09:55      24 documento1.txt
2026-01-30 09:09:55      40 timestamp.txt
2026-01-30 09:09:55      17 usuarios.csv
Archivos restaurados en: /root/proyectos/backup-s3/restaurados
total 24K
-rw-r--r-- 1 root root  28 Jan 30 09:09 config.json
-rw-r--r-- 1 root root  24 Jan 30 09:09 documento1.txt
-rw-r--r-- 1 root root  16 Jan 30 09:15 LATEST.txt
-rw-r--r-- 1 root root  40 Jan 30 09:09 timestamp.txt
-rw-r--r-- 1 root root  17 Jan 30 09:09 usuarios.csv
drwxr-xr-x 2 root root 4.0K Jan 30 09:19 version_20260130_091539
== Restauración completada ==
root@UbuntuPL:~/proyectos/backup-s3#
```

Copiar archivos restaurados al directorio original;

```
root@UbuntuPL:~/proyectos/backup-s3# cp ~/proyectos/backup-s3/restaurados/* ~/proyectos/backup-s3/datos/
cp: -r not specified; omitting directory '/root/proyectos/backup-s3/restaurados/version_20260130_091539'
root@UbuntuPL:~/proyectos/backup-s3# ls -lh ~/proyectos/backup-s3/datos/
total 20K
-rw-r--r-- 1 root root 28 Jan 30 09:55 config.json
-rw-r--r-- 1 root root 24 Jan 30 09:55 documento1.txt
-rw-r--r-- 1 root root 16 Jan 30 09:55 LATEST.txt
-rw-r--r-- 1 root root 40 Jan 30 09:55 timestamp.txt
-rw-r--r-- 1 root root 17 Jan 30 09:55 usuarios.csv
root@UbuntuPL:~/proyectos/backup-s3#
```

```
root@UbuntuPL:~/proyectos/backup-s3# cat ~/proyectos/backup-s3/datos/documento1.txt
Documento importante v1
```

## PARTE 13: Gestión del cron

Ver configuración actual;

```
root@UbuntuPL:~/proyectos/backup-s3# cat ~/proyectos/backup-s3/datos/documento1.txt
Documento importante v1
root@UbuntuPL:~/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
# 15 2 * * * /usr/local/bin/backup_web.sh
# 0 3 * * * /home/juan/script_backup.sh
# */3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
root@UbuntuPL:~/proyectos/backup-s3#
```

Pausar temporalmente el cron;

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

Reactivar el cron;

```
# 0 3 * * * /home/juan/script_backup.sh
*/3 * * * * $HOME/proyectos/backup-s3/scripts/backup-cron.sh
```

Eliminar completamente el cron;

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

```
root@UbuntuPL:~/proyectos/backup-s3# crontab -l
```

```
no crontab for root
```

```
root@UbuntuPL:~/proyectos/backup-s3# █
```

## PARTE 14: Verificación final completa

Ejecutar todos los verificadores;

```
root@UbuntuPL:~/proyectos/backup-s3# 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:"  
==> VERIFICACIÓN COMPLETA DEL SISTEMA ==>  
  
1. Configuración de cron:  
no crontab for root  
  
2. Logs generados:  
total 40K  
-rw-r--r-- 1 root root 347 Jan 30 08:55 backup_20260130_085513.log  
-rw-r--r-- 1 root root 1.2K Jan 30 09:09 backup_20260130_090953.log  
-rw-r--r-- 1 root root 1.1K Jan 30 09:15 backup_versionado_20260130_091539.log  
-rw-r--r-- 1 root root 1.2K Jan 30 09:22 cron_backup_20260130_092240.log  
-rw-r--r-- 1 root root 551 Jan 30 09:23 cron_backup_20260130_092304.log  
-rw-r--r-- 1 root root 508 Jan 30 09:51 limpieza_20260130_095135.log  
-rw-r--r-- 1 root root 508 Jan 30 09:50 limpieza_Friday,  
-rw-r--r-- 1 root root 508 Jan 30 09:51 limpieza_.log  
-rw-r--r-- 1 root root 2.9K Jan 30 09:19 restore_20260130_091900.log  
-rw-r--r-- 1 root root 715 Jan 30 09:53 restore_20260130_095303.log  
  
3. Archivos en S3:
```