

Proyecto: servidor de backups incrementales y automatizados

Resumen

Este proyecto trata sobre montar un sistema de respaldo seguro, cifrado y automatizado usando herramientas comunes en Linux. La idea es hacer backups incrementales, comprimidos y deduplicados de directorios importantes del sistema, y que el resultado se notifique automáticamente por correo. Todo sobre un ordenador con Ubuntu Zorin OS, y está pensado para ejecutarse a diario de forma programada usando cron.

Objetivos

Configurar un sistema de backups cifrados e incrementales.

Automatizar los respaldos con cron y bash scripting.

Comprimir y deduplicar la información respaldada con borgbackup.

Implementar notificaciones por correo electrónico usando msmtplib.

Probar la recuperación de archivos respaldados ante fallos o pérdida de datos.

Tecnologías Utilizadas

Sistema Operativo: Ubuntu Zorin OS

Herramientas: borgbackup, cron, bash, msmtplib

Seguridad: Repositorio cifrado con repokey

Notificaciones: Envío de correos mediante servidor SMTP

1. Preparación del sistema

1. Actualización de paquetes:

```
sudo apt update && sudo apt upgrade -y
```

2. Instalación de herramientas

1. Instalación de BorgBackup y msmtplib:

```
sudo apt install borgbackup msmtplib msmtplib-mta -y
```

3. Inicialización del repositorio de backup

1. (Opcional) Crear un usuario exclusivo para backups:

```
sudo adduser backup
```

2. Crear el directorio del repositorio y asignar permisos:

```
sudo mkdir -p /backup/repositorio  
sudo chown backup:backup /backup/repositorio
```

3. Inicializar el repositorio cifrado:

```
sudo -u backup borg init --encryption=repokey /backup/repositorio
```

Aquí perditá la contraseña con la que cifraremos el repositorio.

4. Script de backup automatizado

1. Crear el script en /usr/local/bin/backup_borg.sh:

```
#!/bin/bash  
export BORG_REPO=/backup/repositorio  
export BORG_PASSPHRASE='la contraseña de antes'  
EMAIL='tucorreo@dominio.com'  
fecha=$(date +%Y-%m-%d_%H-%M)  
DIRECTORIOS="/etc /home /var/log"  
  
borg create --verbose --stats --compression lz4 ::backup-$fecha $DIRECTORIOS  
borg prune -v --list --keep-daily=7 --keep-weekly=4 --keep-monthly=3  
  
if [ $? -eq 0 ]; then  
    echo "Backup OK: $fecha" | msmtp $EMAIL  
else  
    echo "Backup FAILED: $fecha" | msmtp $EMAIL  
fi
```

2. Dar permisos de ejecución:

```
sudo chmod +x /usr/local/bin/backup_borg.sh
```

5. Automatización con cron

Programar el backup diario a las 02:00 AM:

```
sudo crontab -e
```

Agregar la línea:

```
0 2 * * * /usr/local/bin/backup_borg.sh
```

6. Configuración de msmtp

1. Crear y editar /etc/msmtprc:

```
defaults
```

```
auth on
tls on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
logfile ~/.msmtp.log
```

```
account gmail
host smtp.gmail.com
port 587
from tucorreo@gmail.com
user tucorreo@gmail.com
password 'contraseña de aplicación previamente creada en Gmail'
```

```
account default : gmail
```

2. Establecer permisos adecuados:

```
sudo chmod 600 /etc/msmtpc
```

7. Verificación y pruebas

Ejecutar manualmente el script:

```
sudo /usr/local/bin/backup_borg.sh
```

Verificar:

Que se cree un nuevo backup en /backup/repositorio

Que llegue un correo con el estado del backup

Revisar logs si hay errores

8. Recuperación de archivos

1. Listar backups:

```
borg list /backup/repositorio
```

2. Recuperar archivos específicos:

```
borg extract /backup/repositorio::backup-2025-05-10_02-00 etc/passwd
```

Lecciones aprendidas

Automatización y seguridad: Aprendí a implementar un sistema de respaldo robusto y automatizado, con foco en la seguridad y la eficiencia del almacenamiento.

Scripting y cron: Reforcé habilidades en bash scripting y programación de tareas en cron

Notificación y monitoreo: Integré notificaciones vía correo como monitoreo.

Gestión de backups: Experimenté con deduplicación, compresión y rotación de backups, optimizando el uso de espacio en disco.