

Servicio personalizado con systemd

Objetivo

- Crear un **script de respaldo automático**.
- Convertirlo en un **servicio systemd**.
- Configurar **logs y reinicios automáticos**.

1. Preparar el script de respaldo

Creemos un script en `/usr/local/bin/respaldo.sh`:

```
sudo nano /usr/local/bin/respaldo.sh
```

```
#!/bin/bash
# Script simple de respaldo
FECHA=$(date +%Y-%m-%d_%H-%M-%S)
ORIGEN="/home"
DESTINO="/var/backups/home_${FECHA}.tar.gz"

echo "[$(date)] Iniciando respaldo de $ORIGEN a $DESTINO" >>
/var/log/respaldo.log
tar -czf $DESTINO $ORIGEN 2>> /var/log/respaldo.log

if [ $? -eq 0 ]; then
    echo "[$(date)] Respaldo completado con éxito." >>
    /var/log/respaldo.log
else
    echo "[$(date)] Error durante el respaldo." >>
    /var/log/respaldo.log
fi
```

Dar permisos de ejecución:

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

2. Crear la unidad systemd

Archivo de servicio: `/etc/systemd/system/respaldo.service`

```
sudo nano /etc/systemd/system/respaldo.service
```

```
[Unit]
Description=Servicio de respaldo automático
After=network.target
```

```
[Service]
Type=simple
ExecStart=/usr/local/bin/respaldo.sh
Restart=on-failure
RestartSec=30
StandardOutput=append:/var/log/respaldo.log
StandardError=append:/var/log/respaldo.log
```

```
[Install]
WantedBy=multi-user.target
```

Explicación:

- **ExecStart:** comando que se ejecuta (nuestro script).
- **Restart=on-failure:** reinicia el servicio si falla.
- **RestartSec=30:** espera 30s antes de reiniciar.
- **StandardOutput/Error=append:** redirige logs al archivo /var/log/respaldo.log.

3. Recargar systemd y probar

```
sudo systemctl daemon-reload
sudo systemctl start respaldo.service
sudo systemctl status respaldo.service
```

```
jose@jose-VirtualBox:~$ sudo systemctl status respaldo.service
● respaldo.service - Servicio de respaldo automático
   Loaded: loaded (/etc/systemd/system/respaldo.service; disabled; preset: enabled)
   Active: active (running) since Wed 2025-10-01 11:26:37 CEST; 16s ago
     Main PID: 22016 (respaldo.sh)
        Tasks: 4 (limit: 4604)
      Memory: 162.3M (peak: 162.5M)
         CPU: 12.726s
    CGroup: /system.slice/respaldo.service
            └─22016 /bin/bash /usr/local/bin/respaldo.sh
               └─22019 tar -czf /var/backups/home_2025-10-01_11-26-37.tar.gz /home
                  └─22020 /bin/sh -c gzip
                     └─22021 gzip

oct 01 11:26:37 jose-VirtualBox systemd[1]: Started respaldo.service - Servicio de respaldo automático.
```

Para habilitar en arranque:

```
sudo systemctl enable respaldo.service
```

```
jose@jose-VirtualBox:~$ sudo systemctl enable respaldo.service
Created symlink /etc/systemd/system/multi-user.target.wants/respaldo.service → /etc/systemd/system/respaldo.service.
jose@jose-VirtualBox:~$
```

4. Automatizar con un timer (opcional)

En lugar de arrancar el servicio manualmente, se puede usar un **timer** (como cron pero integrado en systemd).

Crear /etc/systemd/system/respaldo.timer:

```
[Unit]
Description=Ejecutar respaldo automáticamente cada día

[Timer]
OnCalendar=daily
Persistent=true

[Install]
WantedBy=timers.target
```

Activar el timer:

```
sudo systemctl enable --now respaldo.timer
```

Comprobar timers activos:

```
systemctl list-timers
```

5. Verificación de logs

Logs vía journal:

```
journalctl -u respaldo.service
```

O bien revisar el archivo:

```
tail -f /var/log/respaldo.log
```

En Resumen:

Con todo lo que hemos hecho, hemos conseguido **implementar un sistema completo de respaldo automático bajo systemd**, con varias ventajas clave:

1 Script de respaldo funcional

- Has creado `/usr/local/bin/respaldo.sh` que **comprime la carpeta `/home`** y la guarda en `/var/backups` con **nombre basado en la fecha**.
- El script también registra cada paso en `/var/log/respaldo.log` (inicio, éxito, errores).
- Resultado: un script que puede ejecutar respaldos de manera autónoma y documentar su actividad.

2 Servicio systemd (`respaldo.service`)

- Convertiste el script en un **servicio systemd**, lo que significa que ahora **systemd controla su ejecución**.
- Configuración lograda:
 - `ExecStart` → ejecuta el script.
 - `Restart=on-failure` → reinicia automáticamente si falla.
 - `StandardOutput/Error=append` → todos los logs van al archivo de log.
 - `After=network.target` → se asegura de que arranque después de la red (útil si más adelante incluyes respaldos remotos).
- Resultado: respaldo confiable, con reinicios automáticos y logging centralizado.

3 Habilitación en arranque

- Con `systemctl enable respaldo.service` → el servicio se ejecuta automáticamente cada vez que se inicia el sistema.
- Resultado: **no necesitas iniciar manualmente el script**; funciona desde el arranque.

4 Timer systemd (`respaldo.timer`)

- Crear un timer permite que el servicio **se ejecute automáticamente a intervalos programados**, en este caso **diario** (`OnCalendar=daily`).

- `Persistent=true` asegura que, si el sistema estaba apagado, al arrancar se ejecutará el respaldo pendiente.
- Resultado: automatización completa sin depender de cron, control total desde systemd.

5 Logs y verificación

- Toda la actividad queda registrada en `/var/log/respaldo.log` y también es accesible desde `journalctl -u respaldo.service`.
- Esto permite:
 - Revisar rápidamente errores.
 - Confirmar que los respaldos se están ejecutando correctamente.
 - Auditar historial de respaldos.

Conclusión

Con todo esto, hemos conseguido:

1. **Un respaldo automático y confiable** de `/home`.
2. **Servicio controlado por systemd**, con reinicios automáticos si falla.
3. **Automatización diaria** con `respaldo.timer`.
4. **Registro centralizado de logs**, fácil de auditar.
5. **Integración completa en Linux moderno**, sin depender de cron, usando únicamente herramientas de systemd.