

Referencia Rápida de systemd

Gestión de servicios, unidades, temporizadores y journalctl

Gestión de Servicios

Comandos Básicos de Servicio

```
systemctl start nginx
systemctl stop nginx
systemctl restart nginx
systemctl reload nginx # recargar configuración
systemctl status nginx
```

Activar / Desactivar

```
systemctl enable nginx # iniciar al arranque
systemctl disable nginx # quitar del arranque
systemctl enable --now nginx # activar + iniciar
systemctl is-enabled nginx
```

Estados del Servicio

active (running) El servicio está ejecutándose normalmente
active (exited) Se ejecutó una vez y salió correctamente
inactive (dead) El servicio está detenido
failed El servicio falló o salió con error
activating El servicio está iniciando

Archivos de Unidad

Ubicación de Archivos de Unidad

/etc/systemd/system/ Unidades creadas por el administrador (mayor prioridad)
/run/systemd/system/ Unidades generadas en tiempo de ejecución
/usr/lib/systemd/system/ Unidades instaladas por paquetes
~/.config/systemd/user/ Unidades a nivel de usuario

Unidad de Servicio Básica

```
[Unit]
Description=My Application
After=network.target
[Service]
ExecStart=/usr/bin/myapp --config /etc/myapp.conf
Restart=on-failure
User=appuser
[Install]
WantedBy=multi-user.target
```

Aplicar Cambios

```
systemctl daemon-reload # recargar archivos de unidad
systemctl restart myapp # aplicar cambios
```

Temporizadores

Unidad de Temporizador

```
[Unit]
Description=Ejecutar copia de seguridad diariamente
[Timer]
OnCalendar=*-*-* 02:00:00
Persistent=true
[Install]
WantedBy=timers.target
```

Sintaxis de OnCalendar

***-*-* 02:00:00** Diariamente a las 2:00 AM
Mon *-*-* 09:00:00 Todos los lunes a las 9 AM
***-*-*01 00:00:00** Primer día de cada mes
hourly / daily / weekly Programaciones abreviadas

Gestión de Temporizadores

```
systemctl list-timers --all
systemctl start backup.timer
systemctl enable backup.timer
systemd-analyze calendar "daily"
```

Targets

Targets Comunes

multi-user.target Arranque normal, multiusuario, sin GUI
graphical.target Escritorio GUI completo
rescue.target Modo de rescate monousuario
emergency.target Shell mínima, solo root
network-online.target Red completamente configurada
timers.target Todas las unidades de temporizador listas

Comandos de Target

```
systemctl get-default
systemctl set-default multi-user.target
systemctl isolate rescue.target
systemctl list-dependencies graphical.target
```

Journalctl

Ver Registros

```
journalctl -u nginx # registros de la unidad
journalctl -u nginx -f # seguir (tail)
journalctl -u nginx --no-pager
journalctl -b # solo arranque actual
```

Filtrar Registros

```
journalctl --since "2026-03-01"
journalctl --since "1 hour ago"
journalctl -p err # errores y superior
journalctl _PID=1234
```

Niveles de Prioridad

emerg (0) El sistema no puede utilizar
alert (1) Se requiere acción inmediata
crit (2) Condición crítica
err (3) Condición de error
warning (4) Condición de advertencia
info (6) Informativo
debug (7) Mensajes de nivel debug

Mantenimiento de Registros

```
journalctl --disk-usage
journalctl --vacuum-size=500M
journalctl --vacuum-time=30d
```

Red

networkctl

```
networkctl list
networkctl status eth0
networkctl up eth0
networkctl down eth0
```

systemd-resolve

```
resolvectl status
resolvectl query example.com
resolvectl flush-caches
resolvectl statistics
```

Esperar Red

```
# En la sección [Unit] del archivo de unidad:
After=network-online.target
Wants=network-online.target
```

Montajes

Unidad de Montaje

```
[Unit]
Description=Montar volumen de datos
[Mount]
What=/dev/sdb1
Where=/mnt/data
Type=ext4
Options=defaults,noatime
[Install]
WantedBy=multi-user.target
```

Unidad de Automontaje

```
[Unit]
Description=Automontaje de datos al acceder
[Automount]
Where=/mnt/data
TimeoutIdleSec=300
[Install]
WantedBy=multi-user.target
```

Convención de Nombres

/mnt/data Archivo de unidad: **mnt-data.mount**
/var/lib/app Archivo de unidad: **var-lib-app.mount**

Ruta de montaje con `/' reemplazado por `-', guión inicial eliminado

Variables de Entorno

Establecer Variables de Entorno

```
[Service]
Environment=APP_ENV=production
Environment=PORT=8080
EnvironmentFile=/etc/myapp/env
```

Formato de Archivo de Entorno

```
# /etc/myapp/env
APP_ENV=production
DATABASE_URL=postgres://localhost/db
SECRET_KEY=changeme
```

Endurecimiento del Servicio

ProtectSystem=strict Sistema de archivos de solo lectura excepto rutas permitidas
ProtectHome=true Ocultar /home, /root, /run/user
NoNewPrivileges=true Evitar escalada de privilegios
PrivateTmp=true /tmp aislado para el servicio
ReadWritePaths=/var/lib/myapp Permitir escritura en rutas específicas

Dependencias

Directivas de Ordenamiento y Requerimiento

After=b.service Iniciar después de b (solo ordenamiento)
Before=b.service Iniciar antes de b (solo ordenamiento)
Requires=b.service Dependencia fuerte; fallar si b falla
Wants=b.service Dependencia débil; no fallar si b falla
BindsTo=b.service Detener cuando b se detiene
Conflicts=b.service No puede ejecutarse al mismo tiempo que b

Referencia Rápida de systemd

Inspeccionar Dependencias

```
systemctl list-dependencies nginx
systemctl list-dependencies --reverse nginx
systemd-analyze dot nginx.service | dot -Tsvg > deps.svg
```

Patrones Comunes

Políticas de Reinicio

Restart=no	Nunca reiniciar (por defecto)
Restart=on-failure	Reiniciar si sale con código distinto de cero
Restart=always	Reiniciar siempre (para demonios)
RestartSec=5	Esperar 5 segundos antes de reiniciar
StartLimitBurst=3	Máximo de reinicios en el intervalo
StartLimitIntervalSec=60	Intervalo para el conteo de reinicios

Sobrescribir sin Editar

```
systemctl edit nginx # crea drop-in
# /etc/systemd/system/nginx.service.d/override.conf
systemctl cat nginx # mostrar configuración efectiva
systemctl revert nginx # eliminar sobrescrituras
```

Análisis del Sistema

```
systemd-analyze # tiempo de arranque
systemd-analyze blame # tiempo por unidad
systemd-analyze critical-chain
systemctl list-units --failed
```