

# Referência Rápida do systemd

Gerenciamento de serviços, units, timers e journalctl

## Gerenciamento de Serviços

### Comandos Básicos de Serviço

```
systemctl start nginx
systemctl stop nginx
systemctl restart nginx
systemctl reload nginx # recarregar configuração
systemctl status nginx
```

### Habilitar / Desabilitar

```
systemctl enable nginx # iniciar no boot
systemctl disable nginx # remover do boot
systemctl enable --now nginx # habilitar + iniciar
systemctl is-enabled nginx
```

### Estados do Serviço

<b>active (running)</b>	Serviço está em execução normalmente
<b>active (exited)</b>	Executou e encerrou com sucesso
<b>inactive (dead)</b>	Serviço está parado
<b>failed</b>	Serviço travou ou encerrou com erro
<b>activating</b>	Serviço está iniciando

## Arquivos de Unit

### Localização do Arquivo de Unit

<b>/etc/systemd/system/</b>	Units criados pelo administrador (maior prioridade)
<b>/run/systemd/system/</b>	Units gerados em tempo de execução
<b>/usr/lib/systemd/system/</b>	Units instalados por pacotes
<b>~/.config/systemd/user/</b>	Units de nível de usuário

### Unit de Serviço Básico

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

### Aplicar Alterações

```
systemctl daemon-reload # recarregar arquivos de unit
systemctl restart myapp # aplicar alterações
```

## Timers

### Unit de Timer

```
[Unit]
Description=Executar backup diário
[Timer]
OnCalendar=*-*-* 02:00:00
Persistent=true
[Install]
WantedBy=timers.target
```

### Sintaxe do OnCalendar

<b>*-*-* 02:00:00</b>	Diariamente às 2:00
<b>Mon *-*-* 09:00:00</b>	Toda segunda-feira às 9h
<b>*-*-01 00:00:00</b>	Primeiro dia de cada mês
<b>hourly / daily / weekly</b>	Agendamentos abreviados

## Gerenciamento de Timers

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

## Targets

### Targets Comuns

<b>multi-user.target</b>	Boot normal, multi-usuário, sem GUI
<b>graphical.target</b>	Desktop com GUI completa
<b>rescue.target</b>	Modo de recuperação de usuário único
<b>emergency.target</b>	Shell mínimo, somente root
<b>network-online.target</b>	Rede totalmente configurada
<b>timers.target</b>	Todos os units de timer prontos

### Comandos de Target

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

## Journalctl

### Visualizando Logs

```
journalctl -u nginx # logs para o unit
journalctl -u nginx -f # acompanhar (tail)
journalctl -u nginx --no-pager
journalctl -b # apenas o boot atual
```

### Filtrando Logs

```
journalctl --since "2026-03-01"
journalctl --since "1 hour ago"
journalctl -p err # erros e acima
journalctl _PID=1234
```

### Níveis de Prioridade

<b>emerg (0)</b>	Sistema inutilizável
<b>alert (1)</b>	Ação imediata necessária
<b>crit (2)</b>	Condição crítica
<b>err (3)</b>	Condição de erro
<b>warning (4)</b>	Condição de aviso
<b>info (6)</b>	Informativo
<b>debug (7)</b>	Mensagens de depuração

### Manutenção de Logs

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

## Rede

### 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
```

## Aguardar Rede

```
# Na seção [Unit] do arquivo de unit:
After=network-online.target
Wants=network-online.target
```

## Montagens

### Unit de Montagem

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

### Unit de Montagem Automática

```
[Unit]
Description=Montar dados automaticamente no acesso
[Automount]
Where=/mnt/data
TimeoutIdleSec=300
[Install]
WantedBy=multi-user.target
```

### Convenção de Nomenclatura

<b>/mnt/data</b>	Arquivo de unit: <b>mnt-data.mount</b>
<b>/var/lib/app</b>	Arquivo de unit: <b>var-lib-app.mount</b>

Caminho de montagem com `/' substituído por `-', traço inicial removido

## Ambiente

### Definindo Variáveis de Ambiente

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

### Formato do Arquivo de Ambiente

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

### Endurecimento do Serviço

<b>ProtectSystem=strict</b>	Sistema de arquivos somente leitura exceto caminhos permitidos
<b>ProtectHome=true</b>	Ocultar /home, /root, /run/user
<b>NoNewPrivileges=true</b>	Impedir escalada de privilégios
<b>PrivateTmp=true</b>	/tmp isolado para o serviço
<b>ReadWritePaths=/var/lib/myapp</b>	Permitir gravações em caminhos específicos

## Dependências

### Diretivas de Ordenação e Requisito

<b>After=b.service</b>	Iniciar após b (apenas ordenação)
<b>Before=b.service</b>	Iniciar antes de b (apenas ordenação)
<b>Requires=b.service</b>	Dependência forte; falha se b falhar
<b>Wants=b.service</b>	Dependência fraca; não falha se b falhar
<b>BindsTo=b.service</b>	Para quando b para
<b>Conflicts=b.service</b>	Não pode ser executado ao mesmo tempo que b

# Referência Rápida do systemd

---

## Inspecionando Dependências

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

## Padrões Comuns

### Políticas de Reinício

<b>Restart=no</b>	Nunca reiniciar (padrão)
<b>Restart=on-failure</b>	Reiniciar em saída com código diferente de zero
<b>Restart=always</b>	Sempre reiniciar (para daemons)
<b>RestartSec=5</b>	Aguardar 5 segundos antes de reiniciar
<b>StartLimitBurst=3</b>	Máximo de reinícios no intervalo
<b>StartLimitIntervalSec=60</b>	Intervalo para contagem de burst

### Substituir Sem Editar

```
systemctl edit nginx # cria drop-in
# /etc/systemd/system/nginx.service.d/override.conf
systemctl cat nginx # exibir configuração efetiva
systemctl revert nginx # remover substituições
```

### Análise do Sistema

```
systemd-analyze # tempo de boot
systemd-analyze blame # tempo por unit
systemd-analyze critical-chain
systemctl list-units --failed
```