systemd Services and logs

CONCEPTS

A unit is anything systemd manages: a service, timer, socket, target or mount. The .service suffix is optional on the command line.

Unit file /etc/systemd/system
↓ `daemon-reload`
Loaded known to systemd
↓ `start`
Active running
↓ `enable`
Boot target starts at boot

enable only controls start at boot; start only controls now. Use enable --now for both.

CONTROL SERVICES

`systemctl start <unit>`
Start a unit now

`systemctl stop <unit>`
Stop a unit

`systemctl restart <unit>`
Stop and start again

`systemctl reload <unit>`
Re-read config without restart

`systemctl reset-failed <unit>`
Clear the failed state

`systemctl --user <command>`
Manage your own user units

INSPECT

`systemctl status <unit>`
State, PID and last log lines

`systemctl list-units --type=service`
Loaded services

`systemctl --failed`
List failed units

`systemctl cat <unit>`
Show unit file plus drop-ins

BOOT AND UNIT FILES

`systemctl enable <unit>`
Start at boot

`systemctl disable <unit>`
Do not start at boot

`systemctl enable --now <unit>`
Enable and start in one go

`systemctl daemon-reload`
Re-read changed unit files

`systemctl edit <unit>`
Override settings in a drop-in

LOGS (JOURNALCTL)

`journalctl -u <unit>`
Logs of one unit

`journalctl -u <unit> -f`
Follow new lines

`journalctl -n 100`
Last 100 entries

`journalctl -b`
Current boot only

`journalctl -p err`
Errors and worse

UNIT FILE BASICS

Minimal service

```
# /etc/systemd/system/myapp.service
[Unit]
Description=My app
After=network-online.target
Wants=network-online.target

[Service]
ExecStart=/usr/local/bin/myapp --serve
Restart=on-failure
User=myapp

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

TIMERS AND ONE-OFFS

`systemctl list-timers`
Timers and next/last run