

systemd Quick Reference

Services, Logs and Boot Targets on One Page

Read-only commands usually work without root privileges. Changes normally require `sudo`.
Use complete unit names, for example `ssh.service`, `NetworkManager.service`, `my-app.service`.

1. Control Services

Status and State

<code>systemctl status ssh.service</code>	Status, process and recent logs
<code>systemctl is-active ssh.service</code>	Is the service currently running?
<code>systemctl is-enabled ssh.service</code>	Is it enabled at boot?
<code>systemctl show ssh.service</code>	Show all properties

Start, Stop and Restart

<code>sudo systemctl start ssh.service</code>	Start now
<code>sudo systemctl stop ssh.service</code>	Stop now
<code>sudo systemctl restart NetworkManager.service</code>	Stop and start again
<code>sudo systemctl reload ssh.service</code>	Reload configuration if supported

Enable and Disable

<code>sudo systemctl enable ssh.service</code>	Start during boot
<code>sudo systemctl disable ssh.service</code>	Do not start automatically
<code>sudo systemctl enable --now my-app.service</code>	Enable and start immediately
<code>sudo systemctl disable --now my-app.service</code>	Disable and stop immediately

Failed Units

<code>systemctl --failed</code>	List all failed units
<code>sudo systemctl reset-failed my-app.service</code>	Clear the failed state

2. Logs and Troubleshooting

View the Journal

<code>sudo journalctl -u ssh.service</code>	Logs for one service
<code>sudo journalctl -b</code>	Logs since the current boot
<code>sudo journalctl -b -1</code>	Logs from the previous boot
<code>sudo journalctl -u NetworkManager.service -b</code>	Service logs since boot
<code>sudo journalctl -u my-app.service -f</code>	Follow logs live
<code>sudo journalctl -p warning..alert -b</code>	Warnings and errors from this boot

Typical Troubleshooting

<code>systemctl status my-app.service</code>	Error, exit code and recent messages
<code>sudo journalctl -x -u my-app.service</code>	Jump to the end and show explanations
<code>systemctl cat my-app.service</code>	Loaded unit file including drop-ins
<code>systemctl show my-app.service -p Result -p ExecMainStatus</code>	Result and exit status
<code>systemd-analyze verify /etc/systemd/system/my-app.service</code>	Check the unit file for errors
Reliable order: read the status, inspect the journal, check the unit file, then fix the issue and restart.	

Useful Journal Options

<code>-n 50</code>	Only the last 50 entries
<code>-since today</code>	Only entries from today
<code>-no-pager</code>	Output without a pager
<code>-o short-iso</code>	Show ISO timestamps

3. Units, Files and Boot

Units and Unit Files

<code>systemctl list-units --type=service</code>	Loaded service units
<code>systemctl list-units --type=service --all</code>	Include inactive loaded units
<code>systemctl list-unit-files --type=service</code>	Installed service files and enablement state
<code>systemctl cat NetworkManager.service</code>	Show the effective unit file

Edit a Service File

<code>sudoedit /etc/systemd/system/my-app.service</code>	Edit a custom unit file
<code>sudo systemctl daemon-reload</code>	Reload unit files
<code>sudo systemctl restart my-app.service</code>	Apply the change
<code>sudo systemctl edit my-app.service</code>	Create a drop-in; reloads automatically
Do not edit packaged files below <code>/usr/lib/systemd/system/</code> directly. Put custom units and overrides below <code>/etc/systemd/system/</code> .	

Targets and Boot Modes

<code>systemctl get-default</code>	Show the default boot target
<code>systemctl list-units --type=target</code>	List loaded targets
<code>sudo systemctl set-default multi-user.target</code>	Set console mode as default
<code>sudo systemctl set-default graphical.target</code>	Set graphical mode as default
<code>sudo systemctl isolate rescue.target</code>	Switch to rescue mode immediately

Timers

<code>systemctl list-timers --all</code>	All timers and next activation
<code>systemctl status my-app.timer</code>	Status of a specific timer

Important: Runtime State and Boot Configuration Are Separate

start / stop

Changes only the runtime state, not startup at boot.
enable --now
Enables startup at boot and starts the service now.

enable / disable

Changes startup at boot, not the current runtime state.
daemon-reload
Reloads changed unit files; restart the service afterwards.