

Ubuntu Systemctl and Journalctl Cheat Sheet

Get more Linux command cheat sheets at [UbuntuFree.com](https://ubuntufree.com)

Full guide with explanations: www.ubuntufree.com/systemctl-journalctl-commands-in-ubuntu-with-examples/

Updated for Ubuntu 26.04 LTS (2026)

Ubuntu 26.04 LTS and Ubuntu 26.10 use systemd to start and supervise services. systemctl controls those services and journalctl reads the logs they write. This guide walks through the commands you need to check, service, enable and debug services, with copy-ready examples and a free cheat sheet.

Check services with systemctl

Service names end in .service, but you can leave the suffix off. On Ubuntu 24.04 and newer, the OpenSSH server uses socket activation, so you may also see an ssh.socket unit listening on port 22.

1. Show service status

Shows whether the service is loaded, enabled and running, plus the latest log lines.

```
systemctl status ssh
```

Press q to leave the pager, or add --no-pager.

2. Quick yes or no checks

Return a single word and an exit code, which makes them handy in scripts.

```
systemctl is-active ssh
systemctl is-enabled ssh
systemctl is-failed ssh
```

3. List running services

Shows every service that is currently running.

```
systemctl list-units --type=service --state=running
```

4. List failed units

The first command to run when something on the system seems broken.

```
systemctl --failed
```

5. List services enabled at boot

Shows unit files and whether they start automatically.

```
systemctl list-unit-files --type=service --state=enabled
```

Start, stop and enable services

These commands change system state, so they need sudo.

6. Start and stop a service

Runs or stops the service right now. It does not change what happens at boot.

```
sudo systemctl start ssh
sudo systemctl stop ssh
```

7. Restart or reload

restart stops and starts the process. reload asks it to re-read its config without dropping connections, if the service supports it.

```
sudo systemctl restart ssh
sudo systemctl reload ssh
```

8. Enable at boot and start now

Creates the boot-time links and starts the service in one step.

```
sudo systemctl enable --now ssh
```

10. Mask a service

Blocks a service from being started at all, even by other units. unmask reverses it.

```
sudo systemctl mask bluetooth  
sudo systemctl unmask bluetooth
```

9. Disable at boot and stop now

Removes the boot-time links and stops the service.

```
sudo systemctl disable --now ssh
```

11. Reboot or power off

systemctl also handles power actions.

```
sudo systemctl reboot  
sudo systemctl poweroff
```

Inspect and edit unit files

Package unit files live in `/usr/lib/systemd/system`. Never edit those directly. Use drop-in overrides in `/etc/systemd/system` so upgrades do not wipe your changes.

12. Show a unit file

Prints the unit file and any drop-in overrides that apply to it.

```
systemctl cat ssh
```

13. Override settings safely

Opens an editor for a drop-in file such as `/etc/systemd/system/ssh.service.d/override.conf`.

```
sudo systemctl edit ssh
```

14. Reload systemd after manual edits

Needed when you create or change unit files without `systemctl edit`.

```
sudo systemctl daemon-reload
```

15. Show dependencies

Displays the units this service needs or pulls in.

```
systemctl list-dependencies ssh
```

16. List timers

Shows scheduled systemd timers, when they last ran and when they run next.

```
systemctl list-timers
```

17. Find slow boot services

`systemd-analyze` reports total boot time. `blame` sorts units by how long they took.

```
systemd-analyze  
systemd-analyze blame | head
```

Read logs with journalctl

Ubuntu keeps a persistent journal in `/var/log/journal`, so logs survive reboots. Members of the `adm` and `sudo` groups can read all system logs. Other users see only their own.

18. Logs for one service

Shows all journal entries for a unit, oldest first.

```
journalctl -u ssh
```

19. Follow logs live

Streams new lines as they arrive, like `tail -f`. Press `Ctrl+C` to stop.

```
journalctl -u ssh -f
```

20. Jump to the newest lines

-e opens the pager at the end. -n limits output to the last N lines.

```
journalctl -u ssh -e
journalctl -n 50 --no-pager
```

22. Errors only

Filters by priority. err includes err, crit, alert and emerg.

```
journalctl -p err -b
```

24. Kernel messages

The journal version of dmesg, limited to the current boot.

```
journalctl -k -b
```

26. Check and trim journal size

Shows disk usage, then deletes entries older than two weeks or beyond a size cap.

```
journalctl --disk-usage
sudo journalctl --vacuum-time=2weeks
sudo journalctl --vacuum-size=500M
```

21. Current and previous boots

-b shows the current boot, -b -1 the one before. --list-boots lists them all.

```
journalctl -b
journalctl -b -1
journalctl --list-boots
```

23. Filter by time

Accepts relative times and full timestamps.

```
journalctl --since "1 hour ago"
journalctl --since "2026-10-19 09:00" --until
"2026-10-19 10:00"
```

25. Different output formats

Useful for scripts or for precise timestamps.

```
journalctl -u ssh -o short-iso -n 20
journalctl -u ssh -o json-pretty -n 1
```

Systemctl and journalctl quick reference

Goal	Command
Is it running?	systemctl status name
Start now / stop now	sudo systemctl start name / stop name
Start at boot	sudo systemctl enable --now name
Apply config change	sudo systemctl reload name or restart name
What failed?	systemctl --failed
Service logs	journalctl -u name -e
Live logs	journalctl -u name -f
Errors this boot	journalctl -p err -b
Last boot logs	journalctl -b -1

Tips and best practices

When a service fails, run `systemctl status name` first, then `journalctl -u name -e` for the full error. Fix the config, run `sudo systemctl daemon-reload` if you edited a unit, and restart.

Many daemons can check their own config before a restart, for example `sudo sshd -t` for OpenSSH. Testing first avoids locking yourself out of a remote server.