systemd

Here's a practical reference of common `systemctl` examples, covering the most useful day-to-day operations:

Service Control

bash

```
systemctl start nginx
systemctl stop nginx
systemctl restart nginx
systemctl reload nginx          # reload config without full
restart
systemctl try-restart nginx     # restart only if already running
systemctl try-reload-or-restart nginx
```

Enable / Disable at Boot

bash

```
systemctl enable nginx
systemctl disable nginx
systemctl enable --now nginx    # enable + start immediately
systemctl disable --now nginx   # disable + stop immediately
systemctl reenable nginx        # disable + re-enable (useful
after unit file changes)
```

Status & Inspection

bash

```
systemctl status nginx
systemctl is-active nginx       # returns active/inactive, exit
code 0/1
systemctl is-enabled nginx      # returns enabled/disabled
systemctl is-failed nginx
systemctl show nginx            # all properties (key=value)
```

```
systemctl show -p MainPID nginx # single property
systemctl cat nginx             # display the unit file(s) in use
systemctl edit nginx            # open drop-in override (creates
/etc/systemd/system/nginx.d/)
systemctl edit --full nginx     # edit full unit file copy
```

Listing Units

bash

```
systemctl list-units
systemctl list-units --type=service
systemctl list-units --state=failed
systemctl list-units --state=running
systemctl list-unit-files
systemctl list-unit-files --type=service
systemctl list-unit-files --state=enabled
```

Failed Services

bash

```
systemctl --failed
systemctl reset-failed          # clear all failed flags
systemctl reset-failed nginx    # clear failed flag for one unit
```

Targets (Runlevels)

bash

```
systemctl get-default
systemctl set-default multi-user.target
systemctl isolate rescue.target         # switch target
immediately
systemctl isolate graphical.target
```

Legacy runlevel	Systemd target
1	rescue.target
3	multi-user.target
5	graphical.target

Dependencies & Order

bash

```
systemctl list-dependencies nginx
systemctl list-dependencies --reverse nginx   # who depends on nginx
systemctl list-dependencies --after nginx     # ordered after
systemctl list-dependencies --before nginx
```

Daemon & Unit File Management

bash

```
systemctl daemon-reload         # required after modifying unit files
systemctl daemon-reexec         # re-execute systemd itself
```

System State

bash

```
systemctl poweroff
systemctl reboot
systemctl suspend
systemctl hibernate
```

```
systemctl emergency             # minimal shell, no services
systemctl rescue
```

Masking

bash

```
systemctl mask nginx            # prevent starting entirely (even manually)
systemctl unmask nginx
```

Masking symlinks the unit to `/dev/null` — useful when you want to ensure a service is never accidentally started (e.g., blocking `firewalld` when using `iptables` directly).

Timers (systemd equivalent of cron)

bash

```
systemctl list-timers           # all active timers with next trigger
systemctl list-timers --all     # include inactive
systemctl start mytask.timer
systemctl enable --now mytask.timer
```

Environment & Overrides

bash

```
systemctl show-environment
systemctl set-environment VAR=value
systemctl unset-environment VAR
```

Drop-in overrides go in `/etc/systemd/system/<unit>.d/*.conf` and survive package upgrades — always prefer these over editing the original unit file.