Systemd is the init system used in modern Linux distributions for managing services, processes, logs, and boot configurations. Below is a cheat sheet with essential systemd commands.

Table of Contents

Managing Services

Command	Description
systemctl start <service>	Start a service.
systemctl stop <service>	Stop a service.
systemctl restart <service>	Restart a service.
systemctl reload <service>	Reload configuration without restarting.
systemctl status <service>	Check service status.
systemctl enable <service>	Enable service to start at boot.
systemctl disable <service>	Disable service from starting at boot.
systemctl is-enabled <service>	Check if a service is enabled.
systemctl mask <service>	Prevent a service from starting.
systemctl unmask <service>	Unmask (re-enable) a masked service.

Example: Restart and Enable Apache

```
sudo systemctl restart apache2
sudo systemctl enable apache2
```

Checking System Boot and Performance

Command	Description
systemctl list-units –type=service	List all running services.
systemctl list-units –failed	Show failed services.
systemctl list-timers	Show scheduled system timers (cron alternative).
systemd-analyze	Show boot time analysis.
systemd-analyze blame	List services causing slow boot.

Example: Check Slow Boot Services

```
systemd-analyze blame
```

Managing System Startup (Targets)

Systemd uses “targets” instead of traditional runlevels.

Command	Description
systemctl get-default	Show current default target.
systemctl set-default multi-user.target	Set system to boot into multi-user mode (no GUI).
systemctl set-default graphical.target	Set system to boot into GUI mode.
systemctl isolate rescue.target	Switch to rescue mode (single-user mode).

Example: Boot into Text Mode

```
sudo systemctl set-default multi-user.target
```

Managing Systemd Unit Files

Systemd services are defined in unit files stored in:

- /etc/systemd/system/ (custom services)
- /lib/systemd/system/ (system-installed services)

Command	Description
systemctl cat <service>	Show service unit file.
systemctl edit –full <service>	Edit service unit file.
systemctl daemon-reload	Reload systemd after editing unit files.

Example: Edit and Reload a Service

```
sudo systemctl edit --full myservice
sudo systemctl daemon-reload
sudo systemctl restart myservice
```

Viewing System Logs (Journalctl)

Systemd uses journald to store logs.

Command	Description
journalctl -xe	Show recent logs with errors.
journalctl -u <service>	View logs for a specific service.
journalctl –since “1 hour ago”	View logs from the last hour.
journalctl -b	Show logs from the last boot.
journalctl –disk-usage	Check journal log size.

Example: View Apache Logs

```
journalctl -u apache2 --since "yesterday"
```

Example: Create a Custom Systemd Service

Create a new service file

```
sudo nano /etc/systemd/system/myapp.service
```

Add the following configuration

```
[Unit]
Description=My Custom Application
After=network.target

[Service]
ExecStart=/usr/bin/python3 /home/user/myapp.py
Restart=always
User=user
Group=user

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

Enable and Start the Service

```
sudo systemctl daemon-reload
sudo systemctl enable myapp.service
sudo systemctl start myapp.service
sudo systemctl status myapp.service
```

Other Useful Commands

Command	Description
hostnamectl	Show system hostname and OS details.
timedatectl	Show and set system time & timezones.
localectl	Display system locale settings.
loginctl	Manage user sessions.

Summary

- systemctl manages services and targets.
- journalctl handles logs.
- Systemd replaces cron for scheduled tasks (using timers).
- Custom unit files allow custom services.

Systemd is a powerful tool—this cheat sheet helps simplify managing your Linux system!