# Day 12-Managing Services

## systemd Overview

systemd is the system and service manager used by most modern Linux distributions.
It manages system state, service startup, shutdown, and logging.

Key Features:

- Run programs in the background as services
- Manage service dependencies
- Handle startup order and service recovery
- Centralized logging with journald

## Creating a Simple systemd Service

- Service unit files are typically stored in:
  - `/etc/systemd/system/<service_name>.service`
- A minimal service unit file looks like this:

```
[Unit]
Description=Simple Background Service

[Service]
ExecStart=/bin/bash /path/to/your/script.sh

[Install]
WantedBy=graphical.target
```

- Explanation:
  - [Unit] Section: Description of the service
  - [Service] Section: Command to start the service
  - [Install] Section: Target at which service should start

## Running a Service

- Start a service:
  - `systemctl start <service_name>.service`
- Check service status:
  - `systemctl status <service_name>.service`

## Enabling Service at Boot

- Enable service to start automatically at boot:
  - `systemctl enable <service_name>.service`
- Disable a service:
  - `systemctl disable <service_name>.service`

## Setting Service Options

- Run the service as a specific user:

`[Service]`
`User=<service_account_name>`

- Restart service automatically on failure:

  `Restart=on-failure`
  `RestartSec=5`

## Managing Dependencies

- To start a service **after** another service:

  `[Unit]`
  `After=network.target`
  `Requires=network.target`

## Daemon Management

- Reload systemd configuration after editing a service file:
  - `systemctl daemon-reload`
- Edit a unit file safely:
  - `systemctl edit <service_name>.service --full`

## Service State Meanings

- active : Service is running
- inactive : Service is stopped
- failed : Service crashed or timed out

## System Targets

- View current default target:
  - `systemctl get-default`
- Set a new default target:
  systemctl set-default multi-user.target
- List all loaded units:
  - `systemctl list-units --all`

## systemctl Common Commands

- Start a service:
  - `systemctl start <service>`
- Stop a service:
  - `systemctl stop <service>`
- Restart a service:
  - systemctl restart <service>
- Reload service configuration:
  - `systemctl reload <service>`

## systemd Logging with journalctl

- View all logs (oldest to newest):
  - `journalctl`

- View logs from current boot:
  - `journalctl -b`
- View logs for a specific unit:
  - `journalctl -u <unit_name>`

## Examples:

- View SSH service logs:
  - `journalctl -u sshd`
- Restart Apache server:
  - `systemctl restart httpd`
- Check status of Docker:
  - `systemctl status docker`