

systemd 서비스 관리 완벽 가이드

재부팅 후에도 서비스가 자동으로 올라온다

● 새벽장애

● **systemd**

○ unit 파일

○ enable

○ 복구

○ 로그

○ 의존성

systemd 란 ?

- PID 1 번 , 시스템 최초 실행 프로세스
- init 을 대체 , 병렬 부팅으로 속도 향상
- 의존성을 스스로 파악해 필요 순서대로 시작
- 우분투 · 데비안 · CentOS 모두 기본 채택

- 새벽장애
- systemd
- **unit 파일**
- enable
- 복구
- 로그
- 의존성

3 / 7

unit 파일 구조 (3 섹션)

```
# /etc/systemd/system/myapp.service

[Unit]
Description=My Python App
After=network.target

[Service]
ExecStart=/usr/bin/python3 /opt/myapp/app.py
WorkingDirectory=/opt/myapp
User=myapp
Restart=always
RestartSec=5

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

● 새벽장애

● systemd

● **unit 파일**

○ enable

○ 복구

○ 로그

○ 의존성

Service 섹션 핵심 옵션

- ExecStart: 서비스 실행 명령어
- WorkingDirectory: 작업 폴더 (상대경로용)
- User: 전용 계정으로 최소 권한 실행
- daemon-reload 로 새 파일 인식 필수

서비스 제어 명령

- 새벽장애
- systemd
- unit 파일
- **enable**
- 복구
- 로그
- 의존성

daemon-reload

설정 다시 읽기

enable

부팅 시 자동 시작 설정

start

지금 당장 서비스 시작

status

running / failed 확인

- 새벽장애
- systemd
- unit 파일
- enable
- **복구**
- 로그
- 의존성

크래시 자동 복구

- Restart=always: 어떤 이유로 종료돼도 재시작
- Restart=on-failure: 비정상 종료시만 재시작
- RestartSec=5: 5 초 후 재시작
- 메모리 부족 주의, 리소스 점검 병행

- 새벽장애
- systemd
- unit 파일
- enable
- 복구
- **로그**
- 의존성

journalctl 통합 로그

```
# 서비스별 로그
journalctl -u myapp

# 실시간 추적
journalctl -u myapp -f

# 오늘 로그만
journalctl -u myapp --since today

# 최근 100 줄
journalctl -u myapp -n 100

# 영속 로그 (재부팅 후에도 유지)
mkdir -p /var/log/journal
systemctl restart systemd-journald
```

- 새벽장애
- systemd
- unit 파일
- enable
- 복구
- 로그
- 의존성

의존성과 시작 순서

- After: 다른 서비스 이후 시작
- Requires: 필수 의존성 명시
- Wants: 느슨한 의존성
- 웹은 DB 이후 시작 → 연결 에러 예방

더 이상 새벽 호출은 없습니다

 [Vultr \\$100 크레딧 \(설명란 \)](#)

 [systemd 전체 설정은 블로그에](#)

 [구독 & 좋아요 부탁드립니다](#)

구독과 좋아요 부탁드립니다! 👍