

# systemd 빠른 참조

서비스 관리, 유닛, 타이머, journalctl

## 서비스 관리

### 기본 서비스 명령어

```
systemctl start nginx
systemctl stop nginx
systemctl restart nginx
systemctl reload nginx # reload config
systemctl status nginx
```

### 활성화 / 비활성화

```
systemctl enable nginx # start at boot
systemctl disable nginx # remove from boot
systemctl enable --now nginx # enable + start
systemctl is-enabled nginx
```

### 서비스 상태

**active (running)** 서비스가 정상 실행 중  
**active (exited)** 한 번 실행 후 정상 종료  
**inactive (dead)** 서비스가 중지됨  
**failed** 서비스가 충돌하거나 오류 종료  
**activating** 서비스가 시작 중

### 유닛 파일

#### 유닛 파일 위치

**/etc/systemd/system/** 관리자 생성 유닛 (최고 우선순위)  
**/run/systemd/system/** 런타임 생성 유닛  
**/usr/lib/systemd/system/** 패키지 설치 유닛  
**~/.config/systemd/user/** 사용자 수준 유닛

### 기본 서비스 유닛

```
[Unit]
Description=My Application
After=network.target
[Service]
ExecStart=/usr/bin/myapp --config /etc/myapp.conf
Restart=on-failure
User=appuser
[Install]
WantedBy=multi-user.target
```

### 변경 사항 적용

```
systemctl daemon-reload # reload unit files
systemctl restart myapp # apply changes
```

## 타이머

### 타이머 유닛

```
[Unit]
Description=Run backup daily
[Timer]
OnCalendar=*-*-* 02:00:00
Persistent=true
[Install]
WantedBy=timers.target
```

### OnCalendar 구문

**\*-\*-\* 02:00:00** 매일 오전 2시  
**Mon \*-\*-\* 09:00:00** 매주 월요일 오전 9시  
**\*-\*-\*01 00:00:00** 매월 1일  
**hourly / daily / weekly** 약식 스케줄

## 타이머 관리

```
systemctl list-timers --all
systemctl start backup.timer
systemctl enable backup.timer
systemd-analyze calendar "daily"
```

## 타겟

### 주요 타겟

**multi-user.target** 일반 부팅, 멀티유저, GUI 없음  
**graphical.target** 풀 GUI 데스크톱  
**rescue.target** 단일 사용자 복구 모드  
**emergency.target** 최소 셸, root만 가능  
**network-online.target** 네트워크가 완전히 구성됨  
**timers.target** 모든 타이머 유닛 준비 완료

### 타겟 명령어

```
systemctl get-default
systemctl set-default multi-user.target
systemctl isolate rescue.target
systemctl list-dependencies graphical.target
```

## Journalctl

### 로그 보기

```
journalctl -u nginx # logs for unit
journalctl -u nginx -f # follow (tail)
journalctl -u nginx --no-pager
journalctl -b # current boot only
```

### 로그 필터링

```
journalctl --since "2026-03-01"
journalctl --since "1 hour ago"
journalctl -p err # errors and above
journalctl _PID=1234
```

### 우선순위 레벨

**emerg (0)** 시스템 사용 불가  
**alert (1)** 즉각 조치 필요  
**crit (2)** 심각한 상태  
**err (3)** 오류 상태  
**warning (4)** 경고 상태  
**info (6)** 정보 메시지  
**debug (7)** 디버그 수준 메시지

### 로그 유지 관리

```
journalctl --disk-usage
journalctl --vacuum-size=500M
journalctl --vacuum-time=30d
```

## 네트워킹

### networkctl

```
networkctl list
networkctl status eth0
networkctl up eth0
networkctl down eth0
```

### systemd-resolve

```
resolvectl status
resolvectl query example.com
resolvectl flush-caches
resolvectl statistics
```

## 네트워크 대기

```
# In unit file [Unit] section:
After=network-online.target
Wants=network-online.target
```

## 마운트

### 마운트 유닛

```
[Unit]
Description=Mount data volume
[Mount]
What=/dev/sdb1
Where=/mnt/data
Type=ext4
Options=defaults,noatime
[Install]
WantedBy=multi-user.target
```

### 자동 마운트 유닛

```
[Unit]
Description=Automount data on access
[Automount]
Where=/mnt/data
TimeoutIdleSec=300
[Install]
WantedBy=multi-user.target
```

### 명명 규칙

**/mnt/data** 유닛 파일: **mnt-data.mount**  
**/var/lib/app** 유닛 파일: **var-lib-app.mount**

마운트 경로의 ` `를 ` `로 대체, 선행 대시 제거

## 환경 변수

### 환경 변수 설정

```
[Service]
Environment=APP_ENV=production
Environment=PORT=8080
EnvironmentFile=/etc/myapp/env
```

### 환경 파일 형식

```
# /etc/myapp/env
APP_ENV=production
DATABASE_URL=postgres://localhost/db
SECRET_KEY=changeme
```

## 서비스 보안 강화

**ProtectSystem=strict** 허용된 경로 외에는 읽기 전용 파일 시스템  
**ProtectHome=true** /home, /root, /run/user 숨김  
**NoNewPrivileges=true** 권한 상승 방지  
**PrivateTmp=true** 서비스 전용 격리된 /tmp  
**ReadWritePaths=/var/lib/myapp** 특정 경로에 쓰기 허용

## 의존성

### 순서 및 요구사항 지시어

**After=b.service** b 다음에 시작 (순서만)  
**Before=b.service** b 이전에 시작 (순서만)  
**Requires=b.service** 강한 의존성; b가 실패하면 같이 실패  
**Wants=b.service** 약한 의존성; b가 실패해도 실패하지 않음  
**BindsTo=b.service** b가 중지되면 같이 중지  
**Conflicts=b.service** b와 동시에 실행 불가

# systemd 빠른 참조

## 의존성 검사

```
systemctl list-dependencies nginx
systemctl list-dependencies --reverse nginx
systemd-analyze dot nginx.service | dot -Tsvg > deps.svg
```

## 자주 쓰는 패턴

### 재시작 정책

<b>Restart=no</b>	재시작 안 함 (기본값)
<b>Restart=on-failure</b>	비정상 종료 시 재시작
<b>Restart=always</b>	항상 재시작 (데몬용)
<b>RestartSec=5</b>	재시작 전 5초 대기
<b>StartLimitBurst=3</b>	인터벌 내 최대 재시작 횟수
<b>StartLimitIntervalSec=60</b>	버스트 계산 인터벌

### 원본 파일 수정 없이 오버라이드

```
systemctl edit nginx # creates drop-in
# /etc/systemd/system/nginx.service.d/override.conf
systemctl cat nginx # show effective config
systemctl revert nginx # remove overrides
```

## 시스템 분석

```
systemd-analyze # boot time
systemd-analyze blame # per-unit time
systemd-analyze critical-chain
systemctl list-units --failed
```