

Cronjobs und systemd-Timer

ITS-Net-Lin

Sebastian Meisel

4. Januar 2025

1 Einleitung

In Linux- und Unix-Systemen sind Cronjobs und systemd-Timer grundlegende Werkzeuge zur Automatisierung wiederkehrender Aufgaben. Während Cronjobs eine traditionelle Methode zur Planung von Aufgaben bieten, stellt systemd mit seinen Timer-Einheiten eine moderne und flexiblere Lösung dar. In diesem Dokument werden sowohl Cronjobs als auch systemd-Timer behandelt, ihre Funktionsweise erklärt und Best Practices zur Nutzung vorgestellt.

2 Cronjobs

Cronjobs sind geplante Aufgaben, die regelmäßig zu festgelegten Zeiten ausgeführt werden. Cron ist ein Daemon, der es ermöglicht, Befehle oder Skripte in regelmäßigen Abständen auszuführen. Die Konfiguration erfolgt durch die Bearbeitung der Datei `crontab`.

2.1 Cronjob erstellen

Mit dem Befehl `crontab -e` wird die Crontab-Datei für den aktuellen Benutzer geöffnet, in der wiederkehrende Aufgaben definiert werden. Ein einfaches Beispiel für einen Cronjob, der täglich um 3 Uhr morgens ein Backup-Skript ausführt, sieht wie folgt aus:

```
1 0 3 * * * /home/user/backup.sh
```

In diesem Beispiel wird der Befehl `/home/user/backup.sh` täglich um 3:00 Uhr ausgeführt.

2.2 Cronjob-Syntax

Die Cronjob-Syntax besteht aus fünf Feldern, die die Ausführungszeit definieren:

```
1 * * * * * Befehl | | | | | | | | | |
2
3   Wochentag (0-7, wobei 0 und 7 Sonntag sind) | | | | |
4   Monat (1-12) | | | | |
5   Tag des Monats (1-31) | | | | |
6   Stunde (0-23) | | | | |
7   Minute (0-59) | | | | |
```

3 systemd-Timer

systemd bietet eine moderne und flexible Methode zur Planung von Aufgaben mit sogenannten Timer-Einheiten. Sie bieten mehr Funktionen und Integration in das systemd-System als Cronjobs und erlauben komplexe Zeitpläne sowie Abhängigkeiten zwischen Aufgaben.

4 systemd-Timer erstellen

Um einen systemd-Timer zu erstellen, müssen zwei Dateien angelegt werden: eine Service-Datei, die die auszuführende Aufgabe beschreibt, und eine Timer-Datei, die die Ausführungszeit festlegt.

4.1 Service-Datei (z.B. /etc/systemd/system/backup.service)

```
1 [Unit]
2 Description=Backup-Skript ausführen
3
4 [Service]
5 ExecStart=/home/user/backup.sh
```

4.2 Timer-Datei (z.B. /etc/systemd/system/backup.timer):

```
1 [Unit]
2 Description=Backup-Timer
3
4 [Timer]
5 OnCalendar=daily
6 Persistent=true
7
8 [Install]
9 WantedBy=timers.target
```

In diesem Beispiel wird das Skript /home/user/backup.sh täglich ausgeführt.

4.3 Timer-Syntax

Die Timer-Syntax von systemd verwendet den Parameter `OnCalendar`, um festzulegen, wann die Aufgabe ausgeführt werden soll. Dabei sind verschiedene Zeitformate möglich, z.B. `daily`, `weekly` oder eine genaue Angabe wie `Mon *--*--03 03:00:00` für eine Ausführung am 3. Montag jedes Monats um 3 Uhr.

5 Vor- und Nachteile

5.1 Cronjobs

- Vorteile:
 - Einfach zu konfigurieren
 - Weit verbreitet und gut dokumentiert
 - Ideal für einfache Zeitpläne
- Nachteile:
 - Eingeschränkte Flexibilität
 - Keine Integration in systemd

5.2 systemd-Timer

- Vorteile:
 - Höhere Flexibilität (z.B. komplexe Zeitpläne, Abhängigkeiten)
 - Bessere Integration in das systemd-System
 - Fehlermanagement und Log-Funktionalitäten
- Nachteile:
 - Etwas komplexer in der Konfiguration
 - Neuere Technologie, weniger verbreitet