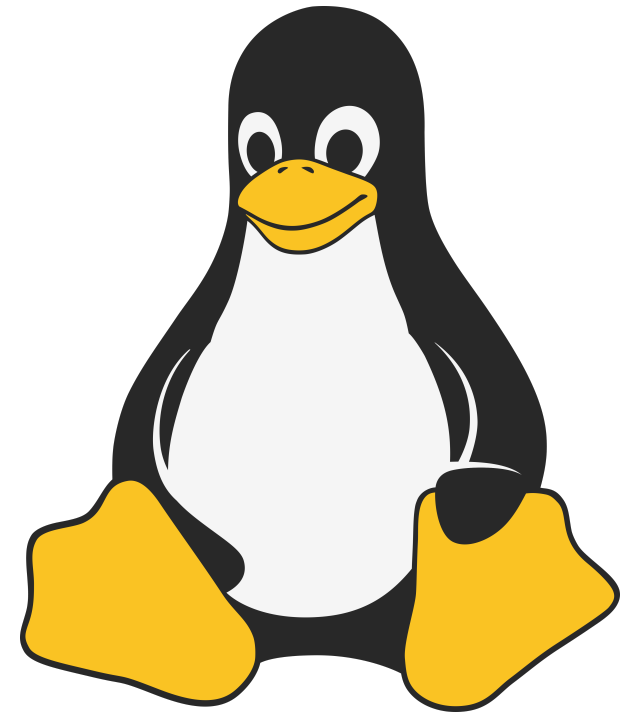


# `systemctl` - Service-Units überwachen und steuern mit Systemd



# Inhaltsverzeichnis

- [Überblick](#)
- [Service-Units auflisten](#)
- [Zustände von Units anzeigen/ändern](#)
- [Status einer Unit anzeigen](#)
- [Unit-Dependencies anzeigen](#)
- [Unit-Files anzeigen](#)

# Überblick

Diese Lektion zeigt, wie man mit `systemctl` Service-Units überwacht und steuert. Dieses Vorgehen gilt nicht nur für Units vom Typ `service`, sondern grundsätzlich auch für alle anderen Unit-Typen.

Wir demonstrieren dies am Beispiel des `cron.service`.

Auch hier gilt: Die Befehle müssen als `root` (`sudo`) ausgeführt werden, wenn Änderungen an Units (Starten, Stoppen etc.) vorgenommen werden sollen. Für reine Abfragen genügen normale Benutzerrechte.

# Service-Units auflisten

Mit `systemctl list-units` lassen sich alle aktuell geladenen Units anzeigen. Mit dem Parameter `-t service` werden nur Units vom Typ `service` angezeigt.

Eine einzelne Unit (hier: einen einzelnen Service) kann man durch Angabe des Unit-Namens anzeigen lassen, z.B.

```
systemctl list-units cron.service.
```

```
hermann@debian:~$ systemctl list-units -t service | head -5; \
> echo "..."; systemctl list-units -t service | tail
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
apache2.service                    loaded active running The Apache HTTP Server
apparmor.service                   loaded active exited Load AppArmor profiles
avahi-daemon.service               loaded active running Avahi mDNS/DNS-SD Stack
connman-wait-online.service         loaded active exited Wait for network to be configured by ConnMan
...
udisks2.service                    loaded active running Disk Manager
user-runtime-dir@1000.service       loaded active exited User Runtime Directory /run/user/1000
user@1000.service                  loaded active running User Manager for UID 1000
wpa_supplicant.service              loaded active running WPA supplicant
```

LOAD = Reflects whether the unit definition was properly loaded.  
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.  
SUB = The low-level unit activation state, values depend on unit type.  
52 loaded units listed. Pass --all to see loaded but inactive units, too.  
To show all installed unit files use '`systemctl list-unit-files`'.

## `cron.service` auflisten

```
hermann@debian:~$ systemctl list-units cron.service
UNIT          LOAD    ACTIVE SUB    DESCRIPTION
cron.service loaded active running Regular background program processing daemon
```

LOAD = Reflects whether the unit definition was properly loaded.  
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.  
SUB = The low-level unit activation state, values depend on unit type.  
1 loaded units listed. Pass --all to see loaded but inactive units, too.  
To show all installed unit files use '`systemctl list-unit-files`'.

# Zustände von Units anzeigen/ändern

- **enabled - disabled:** Gibt an, ob die Unit beim Systemstart automatisch gestartet wird.

Diese Information kann mit `systemctl is-enabled <unit>` abgefragt werden.

Mit `systemctl enable <unit>` bzw. `systemctl disable <unit>` kann der betr. Zustand geändert werden.

```
hermann@debian:~$ systemctl is-enabled cron.service
enabled
hermann@debian:~$ sudo systemctl disable cron.service
Synchronizing state of cron.service with SysV service script with ...
Executing: /lib/systemd/systemd-sysv-install disable cron
Removed "/etc/systemd/system/multi-user.target.wants/cron.service".
hermann@debian:~$ systemctl is-enabled cron.service
disabled
hermann@debian:~$ sudo systemctl enable cron.service
Synchronizing state of cron.service with SysV service script with ...
Executing: /lib/systemd/systemd-sysv-install enable cron
Created symlink /etc/systemd/system/multi-user.target.wants/cron.service
→ /lib/systemd/system/cron.service.
hermann@debian:~$ systemctl is-enabled cron.service
enabled
```



- **active - inactive:** Gibt an, ob die Unit aktuell läuft.

Diese Information kann mit `systemctl is-active <unit>` abgefragt werden.

Mit `systemctl start <unit>` wird die Unit (hier: der Service) gestartet, mit `systemctl stop <unit>` wird sie gestoppt. Mit `systemctl restart <unit>` wird die Unit neu gestartet.

```
hermann@debian:~$ systemctl is-active cron.service
active
hermann@debian:~$ sudo systemctl stop cron.service
hermann@debian:~$ systemctl is-active cron.service
inactive
hermann@debian:~$ sudo systemctl start cron.service
hermann@debian:~$ systemctl is-active cron.service
active
hermann@debian:~$ sudo systemctl restart cron.service
hermann@debian:~$ systemctl is-active cron.service
active
```

# Status einer Unit anzeigen

Mit `systemctl status <unit>` kann der Status einer Unit angezeigt werden. Dazu genügen normale Benutzerrechte.

```
hermann@debian:~$ systemctl status cron.service
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-03-12 09:31:29 CET; 6min ago
     Docs: man:cron(8)
  Main PID: 1371 (cron)
    Tasks: 1 (limit: 2215)
   Memory: 344.0K
      CPU: 5ms
   CGroup: /system.slice/cron.service
           └─1371 /usr/sbin/cron -f
```

## Status von `cron.service` als `root` anzeigen

Bei der Status-Abfrage als `root` werden die Log-Einträge angezeigt.

```
hermann@debian:~$ sudo systemctl status cron.service
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-03-12 09:31:29 CET; 8min ago
     Docs: man:cron(8)
  Main PID: 1371 (cron)
    Tasks: 1 (limit: 2215)
   Memory: 344.0K
      CPU: 6ms
   CGroup: /system.slice/cron.service
           └─1371 /usr/sbin/cron -f

Mär 12 09:31:29 debian systemd[1]: Started cron.service - Regular background program processing daemon.
Mär 12 09:31:29 debian cron[1371]: (CRON) INFO (pidfile fd = 3)
Mär 12 09:31:29 debian cron[1371]: (CRON) INFO (Skipping @reboot jobs -- not system startup)
```

# Status von `ssh.service` als `root` anzeigen

```
hermann@debian:~$ sudo systemctl status ssh.service
```

```
● ssh.service - OpenBSD Secure Shell server
```

```
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
```

```
   Active: active (running) since Wed 2025-03-12 08:27:18 CET; 1h 13min ago
```

```
     Docs: man:sshd(8)
```

```
           man:sshd_config(5)
```

```
 Main PID: 589 (sshd)
```

```
    Tasks: 1 (limit: 2215)
```

```
  Memory: 8.1M
```

```
    CPU: 98ms
```

```
 CGroup: /system.slice/ssh.service
```

```
         └─589 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

```
Mär 12 08:27:18 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
```

```
Mär 12 08:27:18 debian sshd[589]: Server listening on 0.0.0.0 port 22.
```

```
Mär 12 08:27:18 debian sshd[589]: Server listening on :: port 22.
```

```
Mär 12 08:27:18 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

```
Mär 12 08:27:19 debian sshd[696]: Accepted publickey for hermann from 172.22.160.1 port 50120 ssh2: RSA ...
```

```
Mär 12 08:27:19 debian sshd[696]: pam_unix(sshd:session): session opened for user hermann(uid=1000) by (uid=0)
```

```
Mär 12 08:27:20 debian sshd[696]: pam_env(sshd:session): deprecated reading of user environment enabled
```

# Unit-Dependencies anzeigen

Mit `systemctl list-dependencies <unit>` lassen sich die Dependencies einer Unit anzeigen.

```
hermann@debian:~/repo$ systemctl list-dependencies cron.service | head -5; \  
> echo "..."; systemctl list-dependencies cron.service | tail  
cron.service  
● └─system.slice  
● └─sysinit.target  
●   └─apparmor.service  
●   └─dev-hugepages.mount  
...  
●   └─cryptsetup.target  
●   └─integritysetup.target  
●   └─local-fs.target  
●       └─-.mount  
●       └─boot-efi.mount  
○       └─systemd-fsck-root.service  
●       └─systemd-remount-fs.service  
●   └─swap.target  
●       └─dev-disk-by\x2duuid-ff32afed\x2d4797\x2d487a\x2dbecc\x2d271b43edff14.swap  
●   └─veritysetup.target
```

# Unit-Files anzeigen

Mit `systemctl list-unit-files <unit>` lassen sich die Unit-Files einer Unit (hier: eines Service) anzeigen.

```
UNIT FILE    STATE    PRESET
cron.service enabled enabled
```

```
1 unit files listed.
```

In diesem Fall hat der Service `cron` nur ein Unit-File: `cron.service`. Es liegt in einem der Standard-Verzeichnisse für Unit-Files, z.B. in `/lib/systemd/system`.



Mit `systemctl cat <unit>` lässt sich der Inhalt anzeigen.

```
hermann@debian:~/repo$ systemctl cat cron.service
# /lib/systemd/system/cron.service
[Unit]
Description=Regular background program processing daemon
Documentation=man:cron(8)
After=remote-fs.target nss-user-lookup.target
[Service]
EnvironmentFile=-/etc/default/cron
ExecStart=/usr/sbin/cron -f $EXTRA_OPTS
IgnoreSIGPIPE=false
KillMode=process
Restart=on-failure
[Install]
WantedBy=multi-user.target
```

Dies entspricht dem Inhalt des Unit-Files

`/lib/systemd/system/cron.service`.

```
hermann@debian:~$ cat /lib/systemd/system/cron.service
[Unit]
Description=Regular background program processing daemon
Documentation=man:cron(8)
After=remote-fs.target nss-user-lookup.target
[Service]
EnvironmentFile=-/etc/default/cron
ExecStart=/usr/sbin/cron -f $EXTRA_OPTS
IgnoreSIGPIPE=false
KillMode=process
Restart=on-failure
[Install]
WantedBy=multi-user.target
```