Linux

Verschiedene Grundlagen zum Linux Kernel bzw. insbesondere zu diversen Linux Distributionen. SystemD, Autostart, Desktopumgebungen, SSH, ...

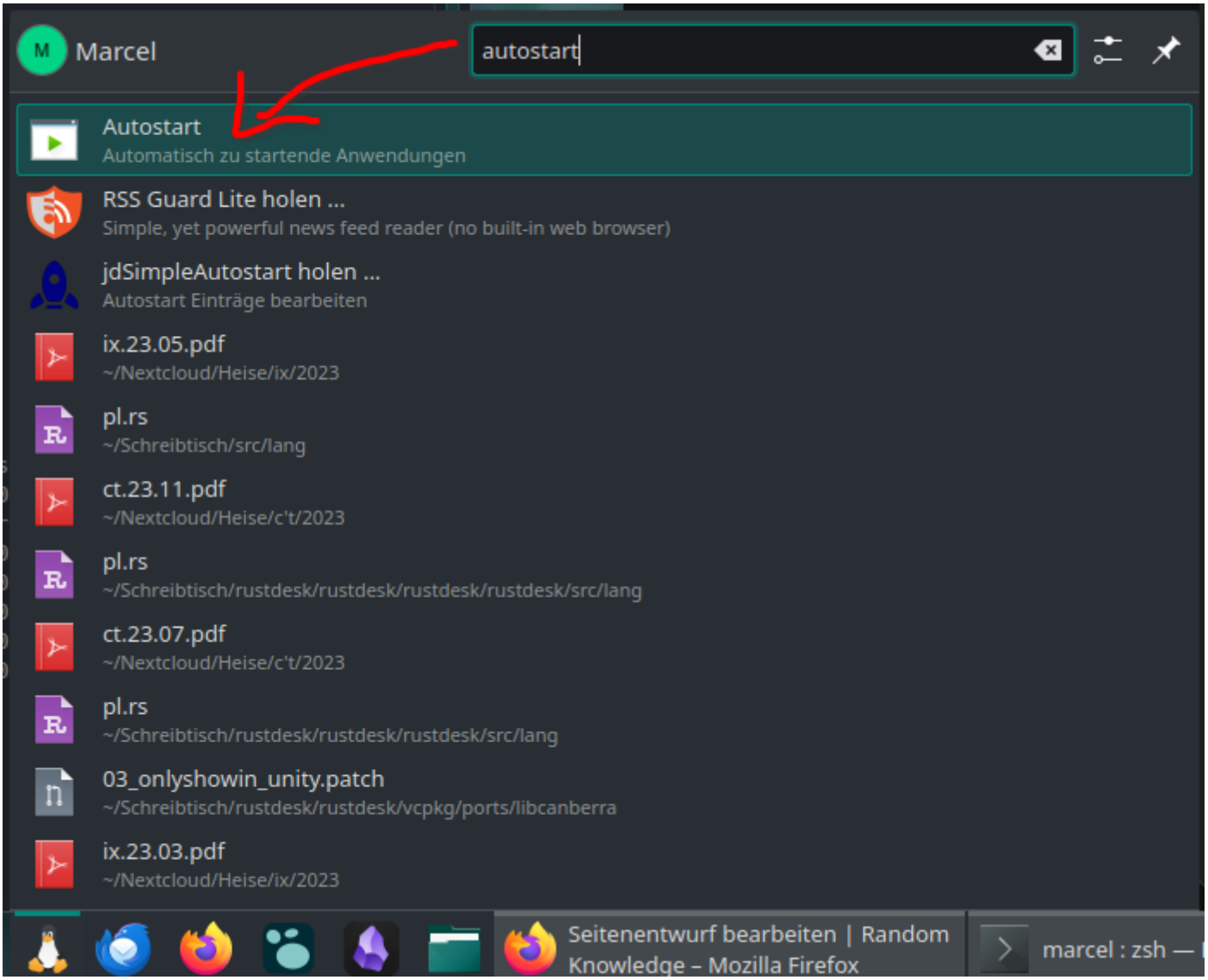
- Autostart
- Dateien live einlesen
- Automatische Updates

Autostart

Je nach Linux Distribution stehen verschiedene Möglichkeiten zur Auswahl, um Programme automatisch zu starten

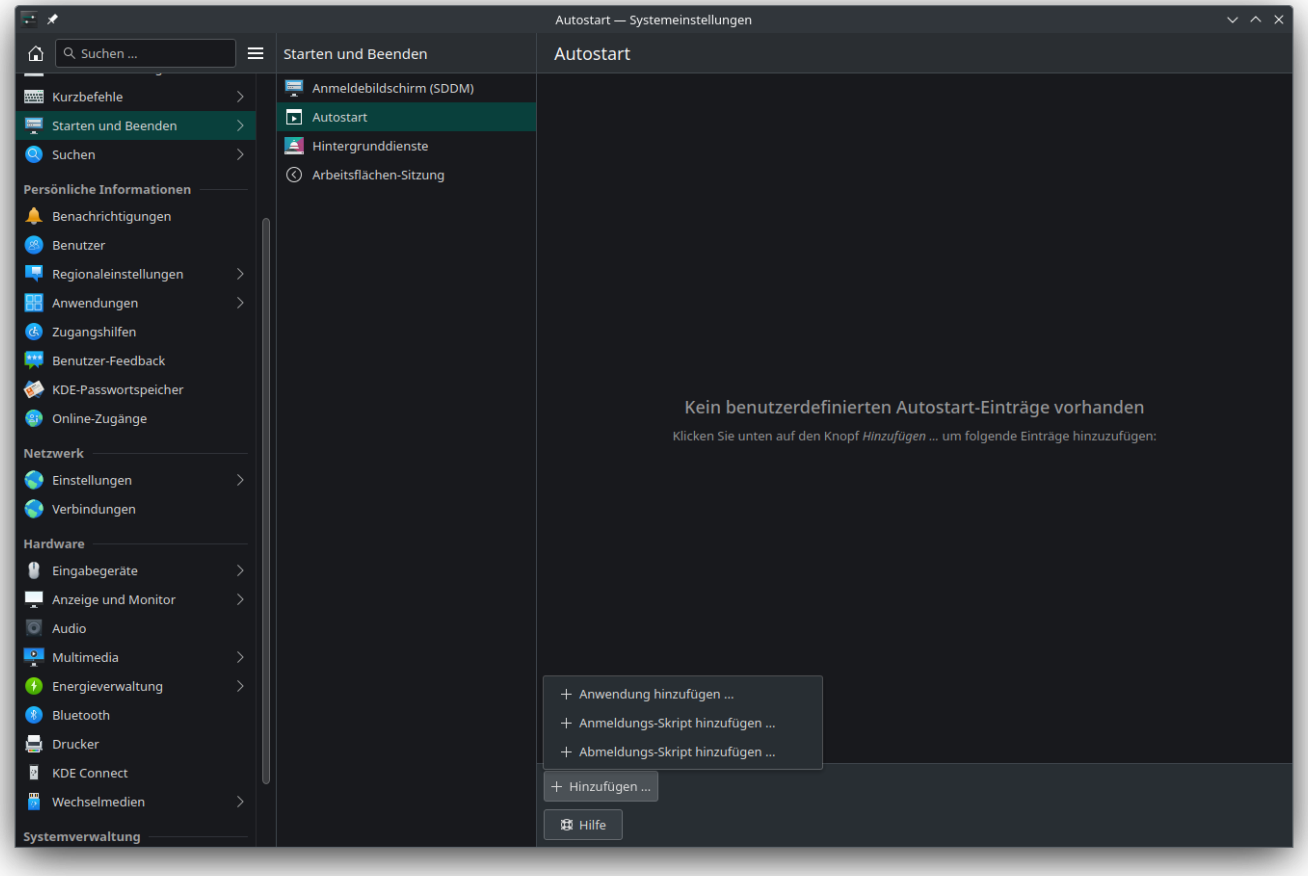
GUI (z. B. KDE, Gnome, Xfce)

Bei Distributionen mit einer grafischen Desktopumgebung wie z. B. Ubuntu (Gnome), Linux Mint (KDE, Xfce, Cinnamon, Mate) oder Debian (KDE) gibt es in der Regel ein Programm mit dem Namen Autostart.

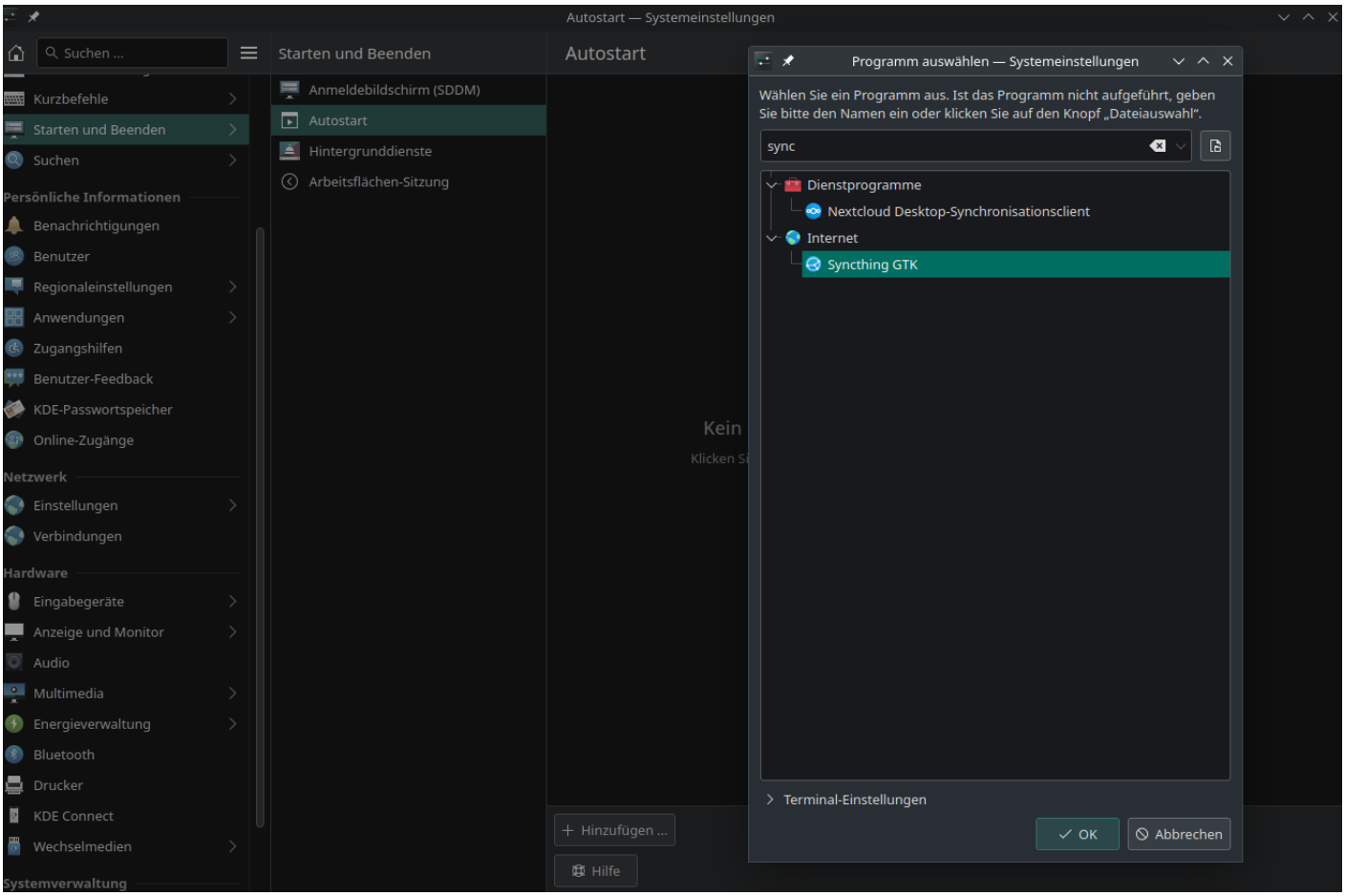


In dem Einstellungsfenster können nun für den aktuellen Benutzer Programme hinterlegt werden, welche beim Anmelden automatisch gestartet werden sollen. Dazu einfach auf das Pluszeichen oder *Hinzufügen* klicken. Es können nicht nur Programme ausgewählt werden, sondern auch An-

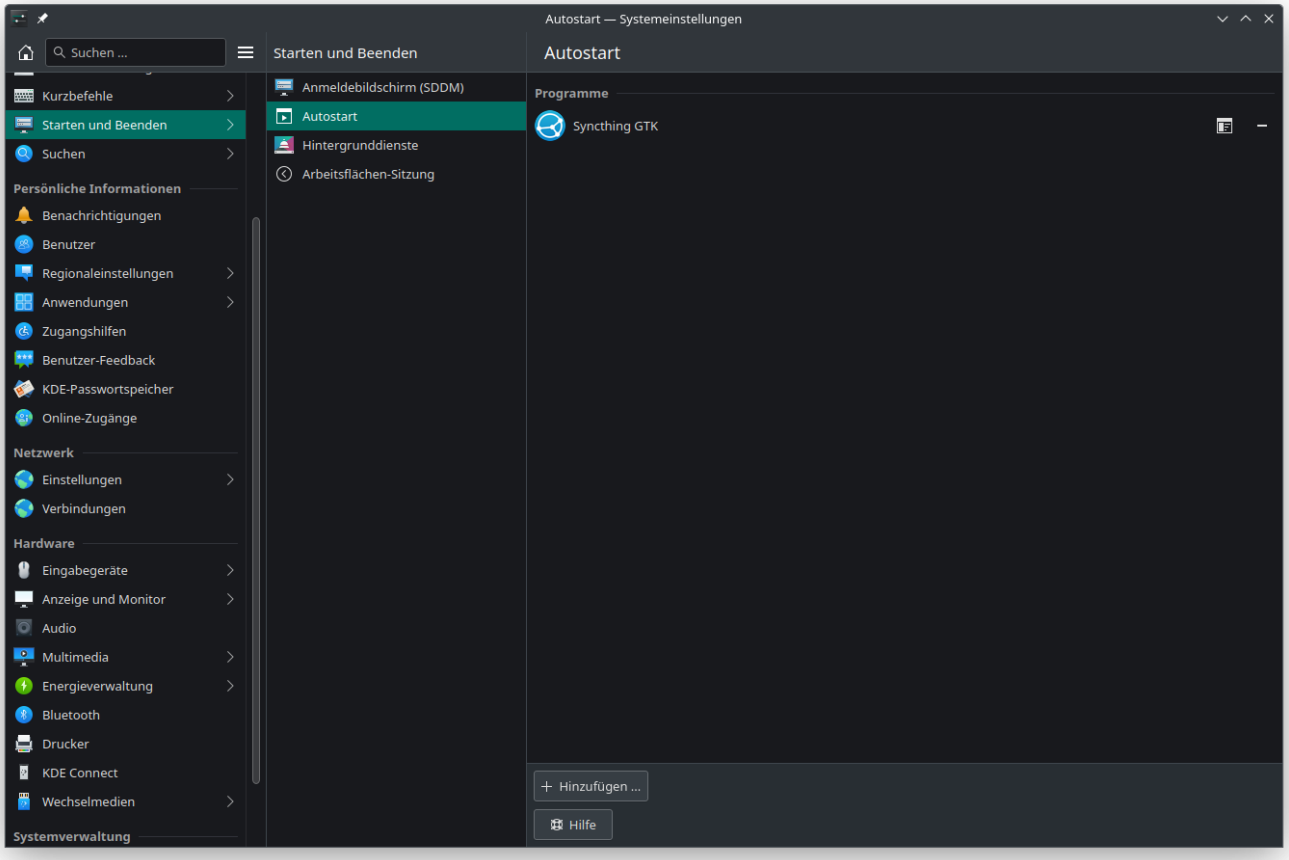
und Abmelde Skripte, die ausgeführt werden sollen. Der folgende Screenshot zeit das Menü Autostart der Distribution Manjaro mit KDE als Desktopumgebung.



Als Beispiel wird nun *Anwendung hinzufügen* ausgewählt, was der häufigste Anwendungsfall sein dürfte. In dem Fall öffnet sich ein neues Fenster, in dem nun das Programm ausgewählt werden kann. Oben wird ein Suchfeld angezeigt, in das der Name des Programms oder ein Teil davon eingegeben werden kann, um die Liste zu durchsuchen.



Wird das Programm mit OK bestätigt, zeigt das Einstellungsfenster es nun in der Liste der Autostart Programme an. Dort kann es jederzeit wieder bearbeitet oder entfernt werden. Natürlich lassen sich noch weitere Programme oder Skripte zum Autostart hinzufügen.


Autostart — Systemeinstellungen
Suchen ...
Kurzbefehle
Starten und Beenden
Suchen
Persönliche Informationen
Benachrichtigungen
Benutzer
Regionaleinstellungen
Anwendungen
Zugangshilfen
Benutzer-Feedback
KDE-Passwortspeicher
Online-Zugänge
Netzwerk
Einstellungen
Verbindungen
Hardware
Eingabegeräte
Anzeige und Monitor
Audio
Multimedia
Energieverwaltung
Bluetooth
Drucker
KDE Connect
Wechselmedien
Systemverwaltung
Starten und Beenden
Anmeldebildschirm (SDDM)
Autostart
Hintergrunddienste
Arbeitsflächen-Sitzung
Autostart
Programme
Syncthing GTK
Hinzufügen ...
Hilfe

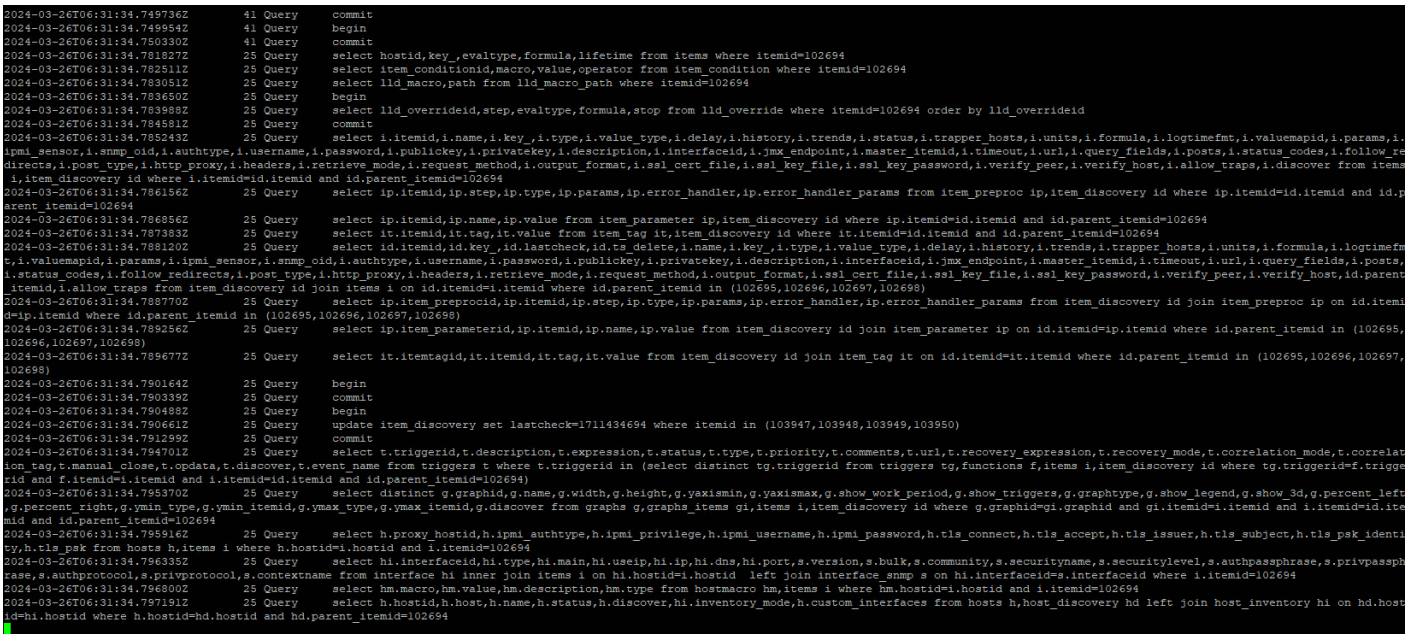
Dateien live einlesen

Bei Logdateien ist es häufig hilfreich, wenn diese konstant eingelesen und dargestellt werden.

Hierfür bietet sich das Tool *tail* an. Angenommen wir haben eine Logdatei */var/log/mysql/abfragen.log*. In diese Logdatei werden alle Abfragen eines MySQL-Servers geschrieben. Wenn nun live im Log gelesen werden soll, welche Abfragen gerade ausgeführt werden, so geht dies mit dem folgenden Befehl.

```
tail -f /var/log/mysql/abfragen.log
```

Wird der Befehl ausgeführt, wird die Datei geöffnet und ans Ende gesprungen. Ab dann werden alle neuen Einträge im Terminal angezeigt.



Automatische Updates

Um Linux Server automatisch auf dem neusten Stand zu halten und somit den besten Schutz gegen Angriffe bieten zu können, bieten sich die automatischen Updates an.

Automatische Sicherheitsupdates aktivieren

Hierfür steht das Paket unattended-upgrades bereit, welches ggf. mit folgenden Befehl nachinstalliert werden muss:

```
sudo apt -y install unattended-upgrades
```

Nach der Installation können die Auto-Updates wie folgt aktiviert werden:

```
sudo dpkg-reconfigure -plow unattended-upgrades
```

Zusätzlich sind noch einige Einstellungen vorzunehmen, damit auch wirklich alle Updates automatisch eingespielt werden.

Zuerst wird der automatische Neustart aktiviert, damit nach der Installation von Updates auch ein Neustart, sofern notwendig, durchgeführt wird. Hierfür ist die Konfigurationsdatei `/etc/apt/apt.conf.d/50unattended-upgrades` anzupassen. Diese enthält diverse Einstellungen, also am besten nach `Automatic-Reboot` suchen, auskommentieren und auf `true` ändern, wie im folgenden Beispiel zu sehen.

```
// Automatically reboot *WITHOUT CONFIRMATION* if
//  the file /var/run/reboot-required is found after the upgrade
Unattended-Upgrade::Automatic-Reboot "true";
```

Sicherheits- und Standard-Pakete aktualisieren

Die Datei noch nicht schließen, denn standardmäßig werden nur Sicherheitsupdates installiert. Also in der selben Konfigurationsdatei nach folgender Zeile suchen und die Kommentarzeichen entfernen.

```
"${distro_id}:${distro_codename}-updates";
```

Zeitpunkt der Suche/Installation anpassen

Nun können wir noch den Zeitpunkt anpassen, wann nach Updates gesucht und wann diese Installiert werden sollen. Dazu dient der folgenden Befehl, mit dessen Hilfe die Konfigurationsdatei bearbeitet wird.

```
sudo systemctl edit apt-daily.timer
```

Daraufhin öffnet sich die Konfiguration für die Update-Suche. Um die Updates z. B. jeden Tag um 01:00 Uhr automatisch suchen zu lassen, wird die Update-Suche wie folgt konfiguriert.

```
### Editing /etc/systemd/system/apt-daily.timer.d/override.conf
### Anything between here and the comment below will become the new contents of the file

[Timer]
OnCalendar=
OnCalendar=01:00
RandomizedDelaySec=0

### Lines below this comment will be discarded

### /lib/systemd/system/apt-daily.timer
# [Unit]
# Description=Daily apt download activities
#
# [Timer]
# OnCalendar=*-*-* 6,18:00
# RandomizedDelaySec=12h
# Persistent=true
#
# [Install]
# WantedBy=timers.target
```

Die Konfiguration für die Installation der Updates könnte z. B. wie folgt gestaltet werden, um diese eine halbe Stunde nach der Suche zu installieren.

```
sudo systemctl edit apt-daily-upgrade.timer
```

Und folgendes in die Konfigurationsdatei.

```
### Editing /etc/systemd/system/apt-daily-upgrade.timer.d/override.conf
### Anything between here and the comment below will become the new contents of the file

[Timer]
OnCalendar=
OnCalendar=01:30
RandomizedDelaySec=0

### Lines below this comment will be discarded

### /lib/systemd/system/apt-daily-upgrade.timer
# [Unit]
# Description=Daily apt upgrade and clean activities
# After=apt-daily.timer
#
# [Timer]
# OnCalendar=*-*-* 6:00
# RandomizedDelaySec=60m
# Persistent=true
#
# [Install]
# WantedBy=timers.target
```

Nun noch die Dienste neu starten und den Status prüfen.

```
sudo systemctl restart apt-daily.timer
sudo systemctl restart apt-daily-upgrade.timer
sudo systemctl status apt-daily.timer
sudo systemctl status apt-daily-upgrade.timer
```

> Unter /var/log/unattended-upgrades/ werden die verschiedenen Logs für Update-Suche, Installation und Neustart abgelegt. Das Log /var/log/unattended-upgrades/unattended-upgrades.log enthält z. B. die wichtigsten Informationen zu installierten Updates und Neustart.

Zusätzliche Pakete aktualisieren

Wenn auf einem System zusätzliche Paketquellen wie z. B. für Docker hinzugefügt wurden, dann müssen diese manuell mit aufgenommen werden.

Mit folgenden Befehl können alle Paketquellen des Systems eingelesen werden.

```
ls -l /var/lib/apt/lists/ | grep "Release"
# Optional kann auch der folgende Befehl genutzt werden, um verschiedene Standard-Paketquellen rauszufiltern
ls -l /var/lib/apt/lists/ | grep "Release" | grep -v "security.ubuntu.com\|asi-fs-n."
```

Die Ausgabe könnte z. B. wie folgt aussehen.

```
-rw-r--r-- 1 root root   108609 Mar 18 19:43 asi-fs-n.contabo.net_ubuntu_dists_jammy-backports_InRelease
-rw-r--r-- 1 root root   270087 Apr 21  2022 asi-fs-n.contabo.net_ubuntu_dists_jammy_InRelease
-rw-r--r-- 1 root root   118761 Mar 27 21:20 asi-fs-n.contabo.net_ubuntu_dists_jammy-updates_InRelease
-rw-r--r-- 1 root root    48847 Mar 22 14:50 download.docker.com_linux_ubuntu_dists_jammy_InRelease
```

Hier sehen wir nun oben die Standard-Pakete von Contabo unter Ubuntu und ganz unten eine Paketquelle von Docker, die zur Zeit nicht aktualisiert wird. Also ermitteln wir zuerst einige Informationen über diese Paketquelle mit folgendem Befehl.

```
cat /var/lib/apt/lists/download.docker.com_linux_ubuntu_dists_jammy_InRelease | grep "Origin\|Suite"
```

In diesem Fall wird nun folgendes ausgegeben.

```
Origin: Docker
Suite: jammy
```

Jetzt können wir das Docker Paket hinzufügen, indem wir wieder die Datei `/etc/apt/apt.conf.d/50unattended-upgrades` bearbeiten. In dieser fügen wir die Zeilen in folgendem Abschnitt zwischen den geschweiften Klammern hinzu. Das Format für die Eingabe ist: `"<Origin>:<Suite>";` und somit in diesem Fall: `"Docker:jammy";`.

```
Unattended-Upgrade::Allowed-Origins {
        "${distro_id}:${distro_codename}";
        "${distro_id}:${distro_codename}-security";
        // Extended Security Maintenance; doesn't necessarily exist for
        // every release and this system may not have it installed, but if
        // available, the policy for updates is such that unattended-upgrades
        // should also install from here by default.
        "${distro_id}ESMApps:${distro_codename}-apps-security";
        "${distro_id}ESM:${distro_codename}-infra-security";
        "${distro_id}:${distro_codename}-updates";
//      "${distro_id}:${distro_codename}-proposed";
```

```
//      "${distro_id}:${distro_codename}-backports";
};
```

Danach sieht der Abschnitt wie folgt aus und wir speichern die Änderung.

```
Unattended-Upgrade::Allowed-Origins {
        "${distro_id}:${distro_codename}";
        "${distro_id}:${distro_codename}-security";
        // Extended Security Maintenance; doesn't necessarily exist for
        // every release and this system may not have it installed, but if
        // available, the policy for updates is such that unattended-upgrades
        // should also install from here by default.
        "${distro_id}ESMApps:${distro_codename}-apps-security";
        "${distro_id}ESM:${distro_codename}-infra-security";
        "${distro_id}:${distro_codename}-updates";
//      "${distro_id}:${distro_codename}-proposed";
//      "${distro_id}:${distro_codename}-backports";
        "Docker:jammy";
};
```

Nun das ganze einmal mit folgendem Befehl testen.

```
sudo unattended-upgrade --dry-run --debug
```

Jetzt sollten auch die Docker Updates angewendet werden. Aufgrund des Parameters `--dry-run` wird der Vorgang nur simuliert und nimmt keine Änderungen am System vor.