Raspberry Pi Spickzettel

Alle wichtigen Befehle auf einer Seite.

SO LIEST DU DAS HIER

Jede Karte hat denselben Aufbau: was der Befehl macht, den Befehl selbst zum Abtippen, eine kurze Erklärung und einen Hinweis aus der Praxis. Alles in **Himbeerrot** musst du durch deine eigenen Werte ersetzen.

BEVOR DU LOSLEGST

Befehle mit `sudo` laufen mit Administratorrechten – lies sie einmal durch, bevor du Enter drückst. Ein falsch gesetztes Leerzeichen kann bei `rm` teuer werden. Alles hier ist auf Raspberry Pi OS in 64 Bit geprüft.

1 System und Grundlagen

Womit jede Sitzung anfängt

System aktualisieren

```
sudo apt update && sudo apt full-upgrade -y
```

apt update holt die aktuellen Paketlisten, **full-upgrade** installiert die Aktualisierungen. Das **-y** beantwortet Rückfragen automatisch mit Ja.

Aus der Praxis: Der erste Befehl auf jedem frischen Pi. Danach einmal neu starten, wenn der Kernel aktualisiert wurde.

Architektur prüfen

```
uname -m
uname -a   # alles auf einmal
```

Zeigt, ob dein System 32 oder 64 Bit ist. Erwartet wird **aarch64**, bei älteren Systemen **armv7l**.

Wichtig: Manche Dienste laufen ausschließlich auf 64 Bit. Meldet der Befehl **armv6l**, ist dein Pi zu alt dafür.

SSH aktivieren

```
sudo raspi-config nonint do_ssh 0
sudo systemctl enable --now ssh
```

Schaltet den Fernzugriff ein, damit du den Pi vom Laptop aus bedienen kannst. Die **0** bedeutet hier „einschalten“.

Achtung: Vorher unbedingt das Standardpasswort ändern. Ein Pi mit SSH und Werkspasswort am Internet ist innerhalb von Stunden übernommen.

Neu starten und herunterfahren

```
sudo reboot
sudo shutdown -h now
```

reboot startet neu, **shutdown -h now** fährt sofort herunter. Das **-h** steht für „halt“.

Achtung: Zieh den Stecker nie ohne Shutdown – das ist der häufigste Grund für zerschossene SD-Karten.

Wie geht es dem Pi?

Temperatur prüfen

```
vcgencmd measure_temp
# dauerhaft beobachten:
watch -n 2 vcgencmd measure_temp
```

Gibt die aktuelle Kerntemperatur aus. Mit **watch -n 2** wird der Wert alle zwei Sekunden aktualisiert.

Faustregel: Unter 60 °C ist entspannt, ab etwa 80 °C drosselt der Pi seine Taktrate. Dann fehlt Kühlung.

Arbeitsspeicher

```
free -h
```

Zeigt belegten und freien Arbeitsspeicher. Das **-h** sorgt für lesbare Einheiten statt roher Kilobytes.

Nicht erschrecken: Ein hoher Wert in der Spalte **buff/cache** ist normal und sogar gut – Linux nutzt freien Speicher als Zwischenspeicher.

Festplatte und SD-Karte

```
df -h
# größte Ordner finden:
sudo du -h / --max-depth=1 | sort -hr | head
```

df zeigt den Füllstand aller Laufwerke, **du** sucht die größten Verzeichnisse.

Wichtig: Bleibt die Systempartition unter zehn Prozent frei, fangen seltsame Fehler an. Räum vorher auf.

Prozessor und laufende Prozesse

```
htop      # komfortabel, ggf. installieren
top       # immer vorhanden
nproc     # Anzahl Kerne
```

Zeigt live, was Rechenzeit verbraucht. Beenden mit **q**.

Tipp: Falls **htop** fehlt: `sudo apt install htop -y`. Deutlich übersichtlicher als top.

Netzwerk

```
hostname -I        # eigene IP-Adresse
ip a               # alle Schnittstellen
ping -c 4 1.1.1.1  # Verbindung testen
```

hostname -I ist der schnellste Weg zur eigenen Adresse im Heimnetz. **ping** prüft, ob überhaupt etwas durchkommt.

Für Dauerbetrieb: Vergib im Router eine feste IP für den Pi. Sonst suchst du ihn nach jedem Neustart neu.

Dienste steuern

```
systemctl status dienstname
sudo systemctl restart dienstname
sudo systemctl enable dienstname
```

status zeigt, ob etwas läuft, **restart** startet neu, **enable** sorgt für automatischen Start nach jedem Boot.

Merkhilfe: start heißt „jetzt“, **enable** heißt „ab jetzt immer“. Zwei verschiedene Dinge.

Docker

Dienste in abgeschotteten Paketen

Was Docker eigentlich macht

Ein Container ist eine Anwendung samt allem, was sie zum Laufen braucht, in einem abgeschotteten Paket. Der praktische Vorteil: Du kannst einen Dienst mit einem einzigen Befehl restlos wieder entfernen, ohne dass Reste auf dem System zurückbleiben. Der Preis dafür: ein zusätzlicher Begriffsapparat, den diese Seite abdeckt.

Docker installieren

```
curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker $USER
newgrp docker
docker --version
```

Zeile 2 erspart dir das **sudo** vor jedem künftigen Docker-Befehl, Zeile 3 übernimmt die neue Gruppe sofort.

Was hier passiert: curl … | sh lädt ein fremdes Skript und führt es sofort mit deinen Rechten aus. Bei get.docker.com ist das der offizielle Weg – bei unbekannten Quellen erst herunterladen und lesen.

Container starten

```
docker run -d \
  --name meinname \
  --restart unless-stopped \
  imagename
```

-d lässt ihn im Hintergrund laufen, **--name** gibt ihm einen wiedererkennbaren Namen, **--restart unless-stopped** startet ihn nach jedem Neustart wieder.

Wichtigster Schalter: Ohne **--restart unless-stopped** ist dein Dienst nach dem nächsten Stromausfall weg – und du merkst es erst Wochen später.

Container prüfen

```
docker ps          # nur laufende
docker ps -a       # auch gestoppte
docker stats       # Last live
```

docker ps ist dein Statusbericht. Fehlt ein Container in der Liste, läuft er nicht.

Zum Suchen: Taucht er nur bei **ps -a** auf, ist er abgestürzt. Die Spalte **STATUS** verrät, warum.

Protokolle anzeigen

```
docker logs meinname
docker logs -f meinname
docker logs --tail 50 meinname
```

Die erste Anlaufstelle, wenn etwas nicht tut. **-f** hängt sich live an, **--tail 50** zeigt nur die letzten 50 Zeilen.

Beenden: Aus dem Live-Modus kommst du mit **Strg + C** heraus. Der Container läuft dabei weiter.

Container stoppen und starten

```
docker stop meinname
docker start meinname
docker restart meinname
```

Stoppen hält den Container an, ohne ihn zu löschen. Alles bleibt erhalten und lässt sich jederzeit wieder starten.

Unterschied: stop ist eine Pause, **rm** ist endgültig. Zum Testen immer erst stoppen.

Container löschen

```
docker rm -f meinname
# auch das Image entfernen:
docker rmi imagename
```

rm -f stoppt und entfernt in einem Schritt. Danach ist der Dienst restlos weg.

Endgültig: Es gibt keine Rückfrage und keinen Papierkorb. Alles, was im Container lag und nicht ausgelagert war, ist weg.

Bei „exec format error“

```
docker run --privileged --rm \
  tonistiigi/binfmt --install all
```

Diese Meldung heißt: Das Image ist für eine andere Prozessorarchitektur gebaut. Der Befehl richtet eine Übersetzungsschicht ein.

Danach: Den Container mit **--platform linux/amd64** starten. Die Emulation kostet etwas Leistung, funktioniert aber.

Aufräumen

```
docker system df
docker image prune -a
docker system prune -a
```

df zeigt, wie viel Platz Docker belegt. **prune** löscht, was nicht mehr gebraucht wird.

Vorsicht bei system prune -a: Entfernt alle nicht laufenden Container und ungenutzten Images. Vorher **docker ps -a** ansehen.

4 screen

Damit nichts abbricht, wenn du dich abmeldest

Wann du screen brauchst – und wann nicht

Programme, die du im Vordergrund startest, sterben mit deiner SSH-Verbindung.

screen

legt eine Sitzung an, die unabhängig davon weiterläuft. Für Docker-Dienste brauchst du das *nicht* – dort erledigt

--restart unless-stopped

dieselbe Aufgabe zuverlässiger.

Die fünf Befehle, die reichen

```
sudo apt install screen -y     # einmalig installieren
screen -S name                 # neue Sitzung anlegen
screen -r name                 # wieder aufschalten
screen -ls                     # alle Sitzungen auflisten
exit                           # Sitzung endgültig schließen
```

Und die beiden Tastenkürzel, die du dir merken musst:

`Strg` + `A` `D` Sitzung verlassen, Prozess läuft weiter

`Strg` + `A` `N` zur nächsten Sitzung wechseln

Die eine Einschränkung: Eine screen-Sitzung überlebt keinen Neustart. Soll etwas dauerhaft laufen, auch nach einem Stromausfall, brauchst du stattdessen einen systemd-Dienst.

5 Wenn etwas klemmt

Die Reihenfolge, die meistens hilft

Was ist überhaupt los?

```
docker ps -a                # läuft es?
docker logs --tail 50 name  # warum nicht?
vcgencmd measure_temp       # zu heiß?
df -h                       # Platte voll?
```

In dieser Reihenfolge abarbeiten. In den allermeisten Fällen steht die Antwort schon in den Protokollzeilen.

Erfahrungswert: Volle SD-Karte, zu schwaches Netzteil und Überhitzung erklären zusammen die meisten Ausfälle.

Stromversorgung prüfen

```
vcgencmd get_throttled
# 0x0 bedeutet: alles in Ordnung
```

Meldet, ob der Pi jemals wegen zu wenig Spannung oder zu hoher Temperatur gedrosselt hat. Jeder Wert außer **0x0** ist ein Hinweis.

Häufigste Ursache: ein Netzteil vom Handy statt des offiziellen. Hier zu sparen kostet Stabilität, nicht Geld.