Raspberry Pi Cheat Sheet

Every command that matters, on one page.

PI NEBENBEI
[YOUR_DOMAIN]
Updated [DATE]

HOW TO READ THIS

Every card has the same structure: what the command does, the command itself to type out, a short explanation and a note from practice. Everything in **raspberry** has to be replaced with your own values.

BEFORE YOU START

Commands with `sudo` run with administrator privileges – read them through once before you press Enter. A misplaced space can get expensive with `rm`. Everything here has been checked on Raspberry Pi OS in 64-bit.

1 System and basics

How every session starts

Update the system

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

apt update fetches the current package lists, **full-upgrade** installs the updates. The **-y** answers prompts automatically with yes.

From practice: The first command on every fresh Pi. Reboot once afterwards if the kernel was updated.

Check the architecture

```
uname -m
uname -a        # everything at once
```

Shows whether your system is 32 or 64 bit. You want to see **aarch64**, on older systems **armv7l**.

Important: Some services only run on 64 bit. If the command reports **armv6l**, your Pi is too old for them.

Enable SSH

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

Switches on remote access so you can operate the Pi from your laptop. The **0** here means “switch on”.

Careful: Change the default password first. A Pi with SSH and the factory password on the internet is taken over within hours.

Reboot and shut down

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

reboot restarts, **shutdown -h now** shuts down immediately. The **-h** stands for “halt”.

Careful: Never pull the plug without a shutdown – it's the most common reason for corrupted SD cards.

2 How is the Pi doing?

Temperature, storage, load

Check the temperature

```
vcgencmd measure_temp
# keep an eye on it:
watch -n 2 vcgencmd measure_temp
```

Prints the current core temperature. With **watch -n 2** the value is refreshed every two seconds.

Rule of thumb: Below 60 °C is relaxed, from around 80 °C the Pi throttles its clock speed. That means cooling is missing.

Memory

```
free -h
```

Shows used and free memory. The **-h** gives readable units instead of raw kilobytes.

Don't panic: A high value in the **buff/cache** column is normal and even good – Linux uses free memory as a cache.

Disk and SD card

```
df -h
# find the biggest folders:
sudo du -h / --max-depth=1 | sort -hr | head
```

df shows how full every drive is, **du** finds the largest directories.

Important: If the system partition drops below ten percent free, strange errors start. Clear up before that.

Processor and running processes

```
htop            # convenient, may need installing
top             # always available
nproc           # number of cores
```

Shows live what is using processing time. Quit with **q**.

Tip: If **htop** is missing: **`sudo apt install htop -y`**. Far clearer than **top**.

Network

```
hostname -I               # your own IP address
ip a                      # all interfaces
ping -c 4 1.1.1.1         # test the connection
```

hostname -I is the quickest way to your own address on the home network. **ping** checks whether anything gets through at all.

For continuous operation: Assign the Pi a fixed IP in your router. Otherwise you'll be hunting for it after every reboot.

Controlling services

```
systemctl status servicename
sudo systemctl restart servicename
sudo systemctl enable servicename
```

status shows whether something is running, **restart** restarts it, **enable** makes it start automatically after every boot.

Memory aid: start means “now”, **enable** means “from now on, always”. Two different things.

3 Docker

Services in sealed-off packages

What Docker actually does

A container is an application together with everything it needs to run, in a sealed-off package. The practical advantage: you can remove a service completely with a single command, without any leftovers staying behind on the system. The price for that: an extra set of terminology, which this page covers.

Install Docker

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

Line 2 saves you the **sudo** before every future Docker command, line 3 applies the new group immediately.

What happens here: `curl … | sh` downloads someone else's script and runs it immediately with your privileges. With get.docker.com this is the official route – with unknown sources, download and read it first.

Start a container

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

-d lets it run in the background, **--name** gives it a recognisable name, **--restart unless-stopped** starts it again after every reboot.

The most important switch: Without **--restart unless-stopped** your service is gone after the next power cut – and you'll only notice weeks later.

Check containers

```
docker ps          # running only
docker ps -a       # stopped ones too
docker stats       # live load
```

docker ps is your status report. If a container is missing from the list, it isn't running.

To investigate: If it only shows up under **ps -a**, it crashed. The **STATUS** column tells you why.

Show the logs

```
docker logs myname
docker logs -f myname
docker logs --tail 50 myname
```

The first place to look when something isn't working. **-f** attaches live, **--tail 50** shows only the last 50 lines.

Quitting: You leave live mode with **Ctrl + C**. The container keeps running.

Stop and start containers

```
docker stop myname
docker start myname
docker restart myname
```

Stopping halts the container without deleting it. Everything is preserved and can be started again at any time.

The difference: stop is a pause, **rm** is final. When testing, always stop first.

Delete containers

```
docker rm -f myname
# remove the image as well:
docker rmi imagename
```

rm -f stops and removes in one step. After that the service is completely gone.

Final: There is no confirmation prompt and no recycle bin. Everything that was inside the container and not stored elsewhere is gone.

On “exec format error”

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

This message means: the image was built for a different processor architecture. The command sets up a translation layer.

Afterwards: Start the container with **--platform linux/amd64**. The emulation costs a little performance, but it works.

Cleaning up

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

df shows how much space Docker is using. **prune** deletes what is no longer needed.

Careful with system prune -a: It removes every container that isn't running and every unused image. Look at **docker ps -a** first.

4 screen

So nothing breaks off when you log out

When you need screen – and when you don't

Programs you start in the foreground die with your SSH connection. **`screen`** creates a session that keeps running independently of it. For the Docker services you don't need this – there, **`--restart unless-stopped`** does the same job more reliably.

The five commands that are enough

```
sudo apt install screen -y                    # install once
screen -S name                                # create a new session
screen -r name                                # reattach to it
screen -ls                                    # list all sessions
exit                                          # close the session for good
```

And the two keyboard shortcuts you have to remember:

Ctrl + A D — leave the session, the process keeps running

Ctrl + A N — switch to the next session

The one limitation: A screen session does not survive a reboot. If something has to run permanently, even after a power cut, you need a systemd service instead.

5 When something's stuck

The order that usually helps

What's going on at all?

```
docker ps -a                      # is it running?
docker logs --tail 50 name        # why not?
vcgencmd measure_temp             # too hot?
df -h                             # disk full?
```

Work through it in this order. In the vast majority of cases the answer is already in the log lines.

From experience: A full SD card, an underpowered PSU and overheating together explain most failures.

Check the power supply

```
vcgencmd get_throttled
# 0x0 means: everything is fine
```

Reports whether the Pi has ever throttled because of low voltage or excessive temperature. Any value other than **0x0** is a hint.

Most common cause: a phone charger instead of the official power supply. Saving money here costs stability, not money.