

# Docker

- [Erstinstallation inkl. Portainer](#)
- [frigate NVR](#)

# Erstinstallation inkl. Portainer

## verwendete Hosts:

- virtual private Server bei IONOS - Debian 12
- raspy3 - raspberry PI5 8GB - Raspberry PI OS - Debian 12

Basis für meine Docker Installationen ist die offizielle Doku:

<https://docs.docker.com/engine/install/debian/>

## Offizielle Repositories einbinden

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

## Docker installieren

... erweitert um "docker-compose"

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin docker-compose
```

# Portainer

Ich möchte für alle meine Docker Container die persistenten Daten an einem Platz finden  
"/opt/docker"

hier der angepasste docker Befehl:

```
docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v  
/var/run/docker.sock:/var/run/docker.sock -v /opt/docker/portainer/data:/data portainer/portainer-ce:2.21.0
```

# frigate NVR

frigate auf dem raspberry

In Portainer:

> Stacks > add Stack

hier das docker compose File von der Original Seite einfügen und anpassen.

<https://docs.frigate.video/frigate/installation/>

Ich nutze das pineboard HatDrive! AI

<https://pineboards.io/blogs/tutorials/installing-frigate-on-raspberry-pi-5-with-a-google-coral-tpu>

## Modifying the config.txt File

We need to make a few changes to the config.txt file to make sure the Coral TPU is able to run, so open /boot/firmware/config.txt with your chosen text editor (with sudo) and add the following lines to the end of your file.

```
kernel=kernel8.img  
dtoverlay=pineboards-hat-ai
```

These enable the PCIe connection on the Raspberry Pi 5 (just in case, it's enabled by default these days), switch the kernel to use 4K block sizes (a requirement of the Coral TPU), and enable the Pineboards AI Hat device tree overlay to tweak things that are needed for the TPU to run as expected.

Now, it's time to reboot and apply the changes. A quick sudo reboot command is enough for this.

Once back up, check with uname -a make sure the Linux kernel version has updated. You should see 6.6.31+rpt-rpi-v8, and it's the v8 we're looking for at the end. If it's there, the change was successful.