This post is not a Frigate camera configuration tutorial. The camera configuration within Frigate is very environment specific, and Frigate has a ton of configuration options. At the time of this post I am using the latest Frigate version (0.12.0) which can leverage built-in go2rtc, which I use in my example. So if you look at Frigate config tutorials on other sites, make sure it's for 0.12 (or later) and uses go2rtc.

Tutorial Sections

This tutorial is pretty long as it covers a lot of ground:

- Promox Host PCIe Passthrough
- Deploy Ubuntu 22.04 VM for Frigate
- Mount Second Drive for Media (Optional)
- Install Docker and QUMU Agent
- Coral PCIe TPU Passthrough (Optional)
- Home Assistant MQTT Setup
- Home Assistant Frigate Integration Install
- Camera Configuration (IPC-T5442T-ZE)
- Frigate Configuration File
- Frigate Docker Compose
- Verify Frigate Functionality
- Home Assistant Frigate Integration

Proxmox Host PCIe Passthrough

```
update-grub
```

4. Let’s modify the kernel modules file.

```
nano /etc/modules
```

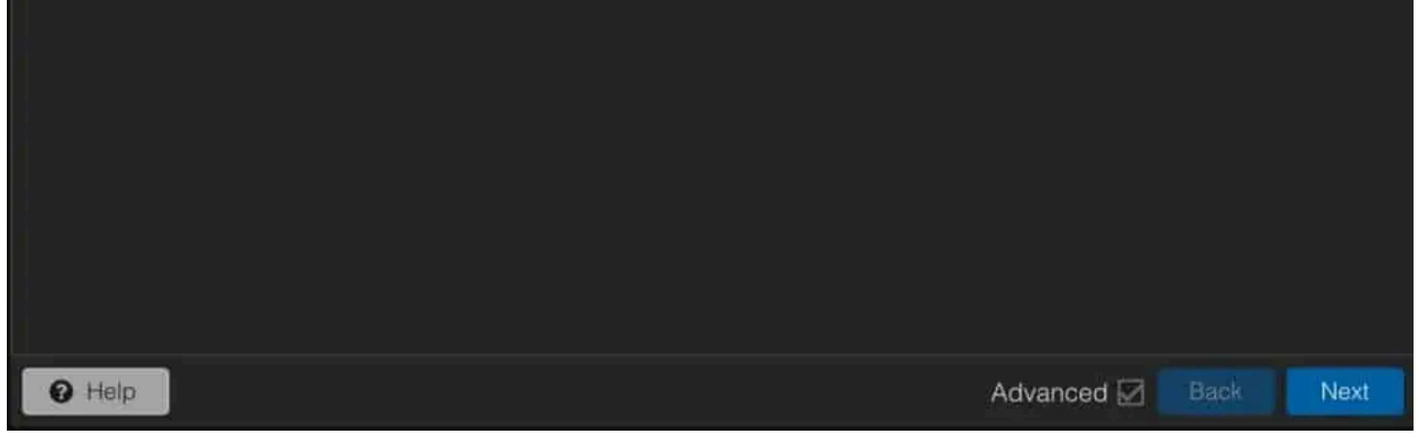
5. Add the following lines to the file, then save and exit nano.

```
vfio
vfio_iommu_type1
vfio_pci
```

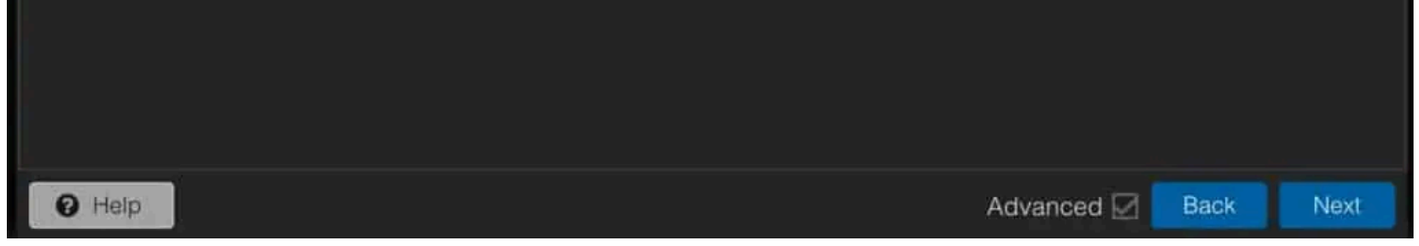
8. Add the following lines to the blacklist file:

```
blacklist gasket
blacklist apex
options vfio-pci ids=1ac1:089a
```

9. Save and exit nano, then run the following command:



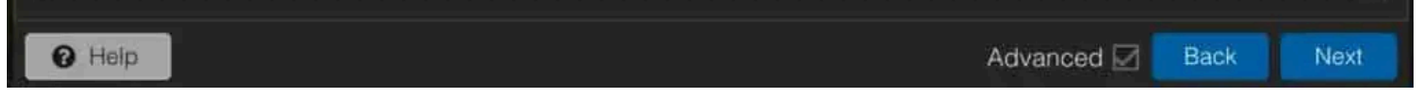
2. Download the Ubuntu 22.04 ISO. Upload it to a Proxmox datastore. On the OS page mount your Ubuntu 22.04 ISO.



4. On Disks configure the following:

- **Disk size:** 16GB (or more if you wish)
- **SSD emulation:** Check
- **Cache:** Write through
- **Discard:** Check
- **IO thread:** Check

Note: If you have separate NVR storage add that virtual disk now. I highly recommend a spinning disk that's designed for video surveillance like Seagate SkyHawk. 2.5″ SkyHawk drives are no longer made (just 3.5″), but Ebay may have factory sealed 2TB 2.5″ SkyHawk drives. I added a 1.7TB virtual drive.

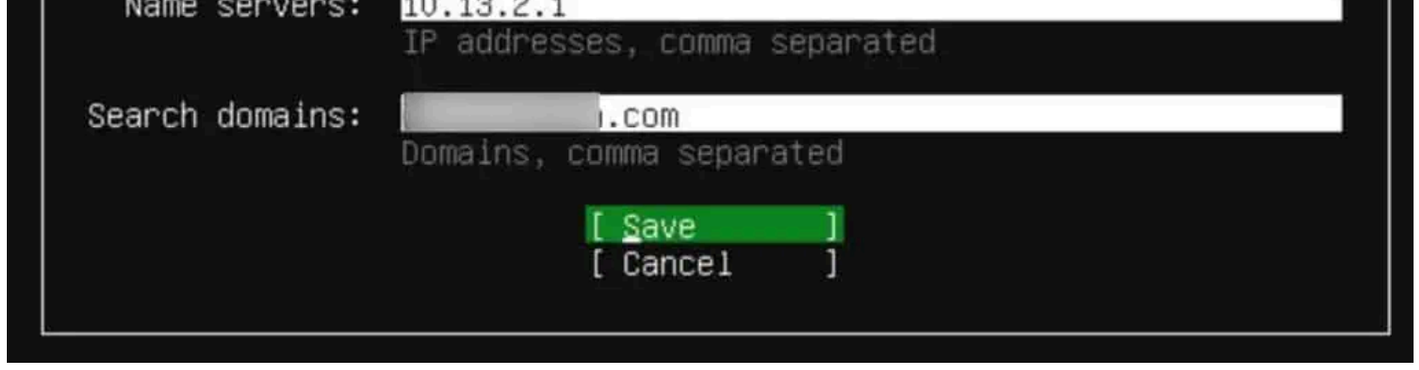


6. On the Memory page configure:

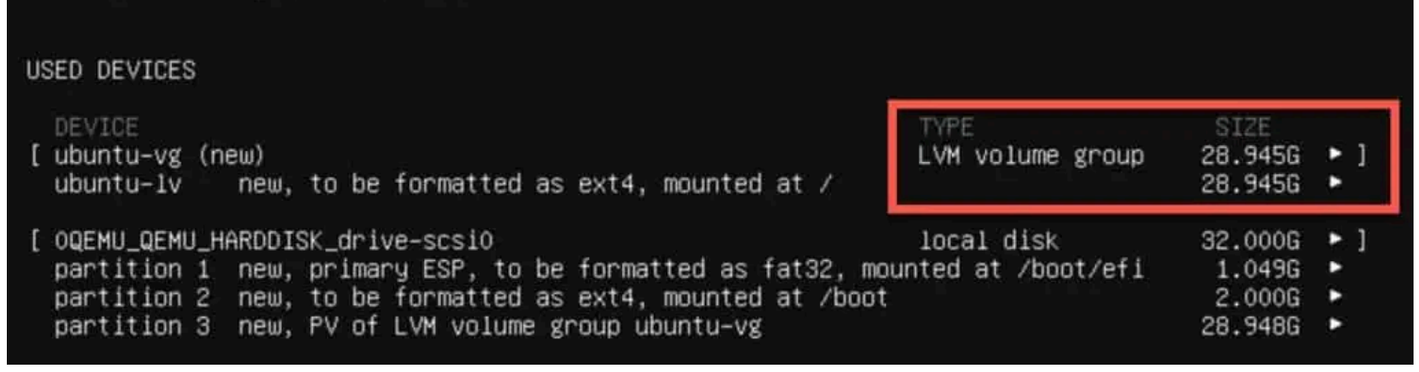
- **Memory:** 2048
- **Ballooning Device:** un-check

8. Check the summary page and tick the box to **Start after created**.

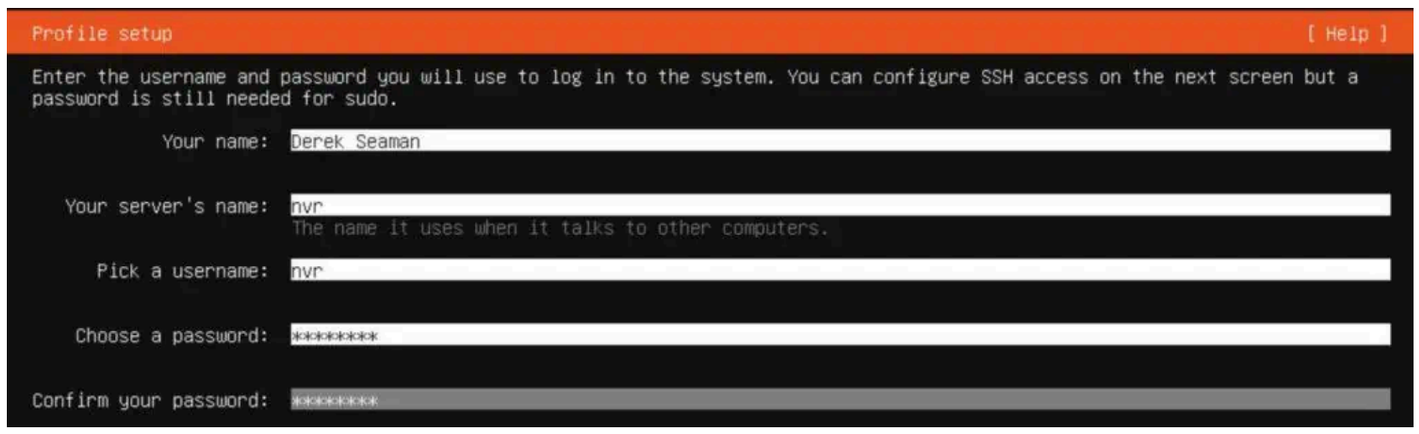
```
agent 1
balloon 0
bios ovmf
cores 1
cpu host
efidisk0 local-lvm:1,efitype=4m
ide2 local:iso/ubuntu-22.04.2-live-server-amd64.iso,media=cdrom
memory 2048
name NVR
net0 virtio,bridge=vmbr0,firewall=1
numa 0
onboot 1
ostype l26
scsi0 NVR:16,discard=on,ssd=on,iothread=on,cache=writethrough
scsihw virtio-scsi-single
sockets 1
start 1
vga std
```



16. On the **Configure Ubuntu archive mirror** press **Enter** on **Done**.
17. On the **Guided Storage configuration** screen tab to **Done** and press **Enter**.
18. On the **Storage configuration** tab to **ubuntu-lv** under **USED DEVICES** and press **ENTER**. Tab to **Edit** and press **ENTER**.



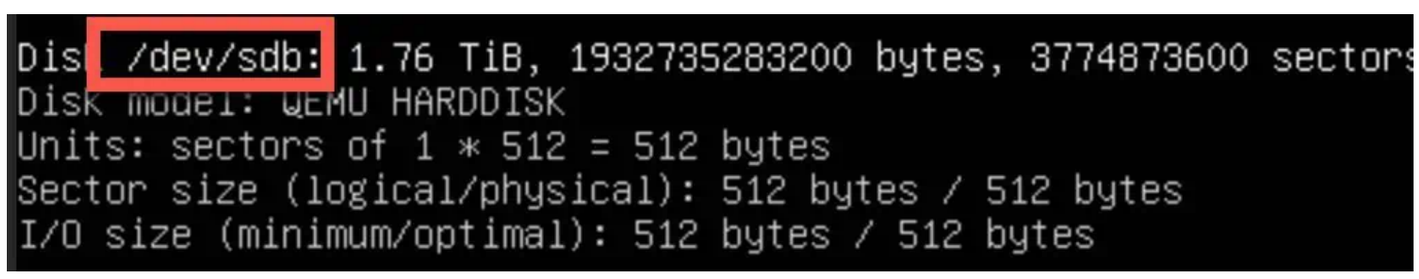
20. Tab to **Done** on the **Storage configuration** page.
21. On the Confirm **Destructive action** windfow tab to **Continue** and press **ENTER.**
22. On the **Profile setup** enter the appropriate information.



23. **Skip** the **Upgrade to Ubuntu Pro**.
24. On the SSH Setup screen **tick** the box to **install OpenSSH server**.

2. Run the following command to list all disks mounted to the Frigate VM. Locate the device name for the new disk (**/dev/sdb** in my case).

```
sudo fdisk -l
```



3. Now we need to partition the raw disk by running the following command:

```
sudo cgdisk /dev/sdb
```

```
sudo nano /etc/fstab
```

7. At the end of the file add the following line, save and exit nano.

```
/dev/sdb1 /media/frigate ext4 defaults 0 0
```

8. Mount the drive:

```
sudo mount /media/frigate
```

```
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

```
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.g
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

3. After the repos are setup let's install Docker. The hello-world test will verify that Docker is running and pulled down an image.

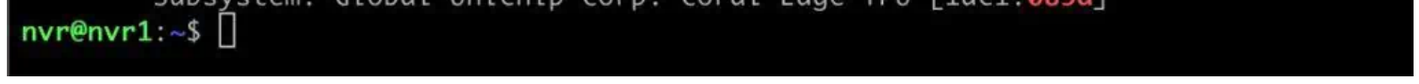
```
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

4. I like setting up Docker to store the app config files under **/home/docker**.

```
sudo mkdir -p  /home/docker/frigate
```

5. Install the QEMU agent and update all the packages:

```
sudo apt-get upgrade
sudo apt install qemu-guest-agent
sudo systemctl enable qemu-guest-agent
sudo reboot
```



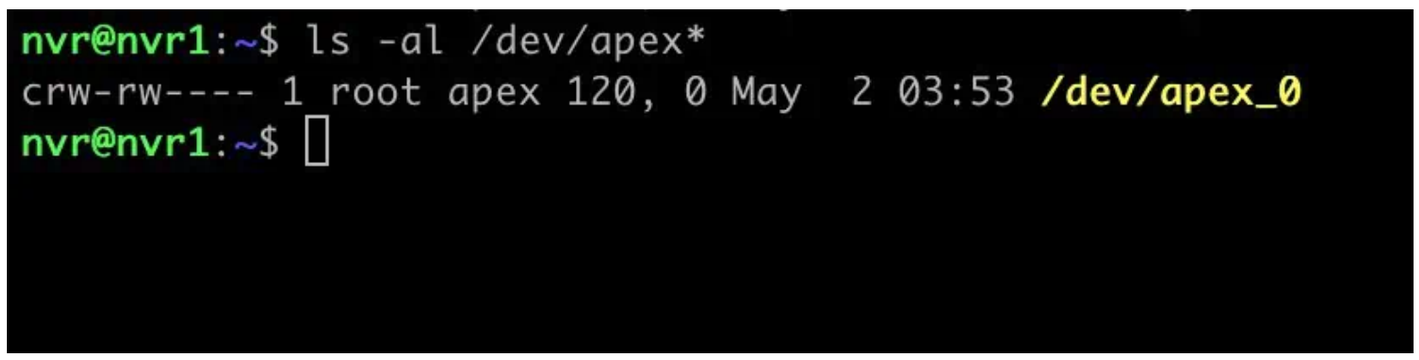
3. Let's start the driver installation:

```
echo "deb https://packages.cloud.google.com/apt coral-edgetpu-stable main" | sud
curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
sudo apt-get update
```

4. Install the PCIe drivers and TPU runtime:

```
sudo apt-get install gasket-dkms libedgetpu1-std
```

```
ls -al /dev/apex*
```



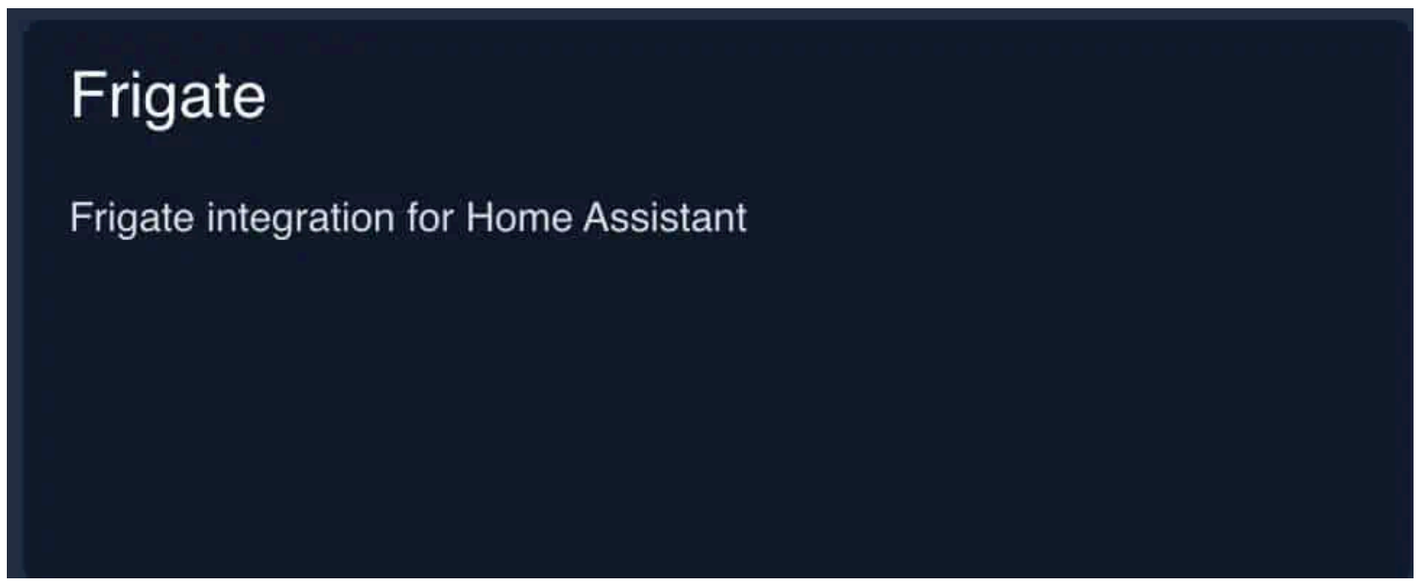
Home Assistant MQTT Setup

If you have already installed and configured a MQTT broker in Home Assistant, you can skip this section.

1. In Home Assistant go to **Settings** -> **Add-ons**.
2. Click on **Add-on Stor**e. Search for **MQTT** and click on **Mosquito broker**.
3. Click on **Install**.
4. After it installs, enable **Watchdog**.
5. Click on **Start**.
6. Click on **Settings** in the lower left, then **Devices & Services**.
7. **MQTT** should be auto-discovered. Click **CONFIGURE**.

```
10
11  lovelace:
12    resources:
13      - url: /hacsfiles/frigate-hass-card/frigate-hass-card.js
14        type: module
15
```

7. Restart Home Assistant.



Camera Configuration (IPC-T5442T-ZE)

Every camera will have their own unique optimal settings. In my case the following settings for my Dahua work very well. Use the best practices for your particular camera. Just for reference, my IP camera is at

```
# Only enable the detector if you are using coral PCIe passthrough
detectors:
  coral:
    type: edgetpu
    device: pci

logger:
  default: info

rtmp:
  enabled: false

live:
  height: 720
  quality: 1

birdseye:
  enabled: True
  restream: false
  width: 1280
  height: 720
  quality: 1
  mode: continuous

go2rtc:
  streams:
    garage:
      - rtsp://feed:password1234@10.13.2.231:554/cam/realmonitor?channel=1&subty
    garage_sub:
      - rtsp://feed:password1234@10.13.2.231:554/cam/realmonitor?channel=1&subty
    webrtc:
      candidates:
        - 10.13.2.217:8555
        #- stun:8555

cameras:
  garage:
    ffmpeg:
      inputs:
        - path: rtsp://127.0.0.1:8554/garage_sub
```

you prefer Docker compose via CLI, just save the docker compose file as **frigate-compose.yml** in **/home/docker/frigate.**

Note: If you aren't using the Coral PCIe TPU, remove the devices section in the compose file (lines 11-12).

```
# ~/docker/frigate/frigate-compose.yml

version: "3.9"
services:
  frigate:
    container_name: frigate
    privileged: true
    restart: unless-stopped
    image: ghcr.io/blakeblackshear/frigate:stable
    shm_size: "384mb" # update for your cameras based on calculation
    devices:
      - /dev/apex_0:/dev/apex_0 # passes a PCIe Coral
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - /home/docker/frigate:/config
      - /media/frigate:/media/frigate
      - type: tmpfs # Optional: 1GB of memory, reduces SSD/SD Card wear
        target: /tmp/cache
        tmpfs:
          size: 1000000000
    ports:
      - "5000:5000"
      - "8554:8554" # RTSP feeds
      - "8555:8555/tcp" # WebRTC over tcp
      - "8555:8555/udp" # WebRTC over udp
    environment:
      FRIGATE_RTSP_PASSWORD: "rtsppassword"
```

4. Now you can add the Frigate card to your dashboard. It has a huge amount of configuration options. So you can really go to town on customizing your camera dashboard views.

SMART HOME

Home Assistant: The Ultimate Light Automation Blueprints

June 8, 2024

RUCKUS

6 GHz Wi-Fi: Are Your Devices ready for Standard Power?

April 23, 2024

Subscribe

1024

Join the discussion

22 COMMENTS

Oldest

Anders June 24, 2023 10:09 am

Raj September 4, 2023 2:34 pm

Does these steps apply to the mini PCIE version of the TPU as well. I installed it with pcie adapter and it is detected by proxmox.

0 Reply

Author

Derek Seaman September 12, 2023 3:11 am

> *Reply to Raj*

It should be the same as I think the adapter is just physical.

0 Reply

Another Derek October 12, 2023 8:40 am

First, thank you for making this detailed guide. You stated “I prefer to run it [Frigate] in a separate dedicated VM on Proxmox”. Is this for performance and/or flexibility? I’m just getting started with smart home security and looking to setup something similar to what you have detailed.

0 Reply

Steve October 24, 2023 8:08 pm

Love the guide — wondering how you update go2rtc.

0 Reply

Ruslan November 6, 2023 5:05 am

Thanks for the guidance. Very well and described in detail. My server runs on a Ryzen 5600u processor. Has anyone been able to make it work with AMD drivers.

0 Reply

gasket-dkms. (Reading database … 130369 files and directories currently installed.)
Preparing… Read more »

+ 0 — Reply

Michael McGuinness July 26, 2024 7:56 pm

Deprecated feature: REMAKE_INITRD (/var/lib/dkms/gasket/1.0/source/dkms.conf)
ERROR: Cannot create report: [Errno 17] File exists: '/var/crash/gasket-dkms.0.crash'
Error! Bad return status for module build on kernel: 6.8.0-39-generic (x86_64)
Consult /var/lib/dkms/gasket/1.0/build/make.log for more information.
dpkg: error processing package gasket-dkms (–configure):
 installed gasket-dkms package post-installation script subprocess returned error exit status 10
Errors were encountered while processing:
 gasket-dkms
needrestart is being skipped since dpkg has failed
E: Sub-process /usr/bin/dpkg returned an error code (1)

+ 0 — Reply

Michael McGuinness July 26, 2024 7:58 pm

The following outputs work:

nvr@nvr:~$ lspci -nnk | grep 089a
00:10.0 System peripheral [0880]: Global Unichip Corp. Coral Edge TPU [1ac1:089a]
Subsystem: Global Unichip Corp. Coral Edge TPU [1ac1:089a]

nvr@nvr:~$ lspci -nn | grep 089a
00:10.0 System peripheral [0880]: Global Unichip Corp. Coral Edge TPU [1ac1:089a]

Though apex is not being created
nvr@nvr:~$ ls -al /dev/apex*
ls: cannot access '/dev/apex*': No such file or directory