

AI - Debian 12 Docker környezet beállítása NVIDIA RTX 4070 GPU-val Ollama számára

Ez a lépésenkénti útmutató bemutatja, hogyan állíthatsz be egy Debian 12 rendszeren futó Docker környezetet, amely az NVIDIA RTX 4070 GPU-t használja az Ollama AI modell futtatásához. A leírás tartalmazza az NVIDIA driver, a CUDA Toolkit, az NVIDIA Container Toolkit telepítését, a Docker Compose konfigurációt, valamint a tesztelési és hibaelhárítási lépéseket.

🚧 Előfeltételek

- Debian 12 rendszer.
- NVIDIA RTX 4070 videokártya.
- Docker és Docker Compose telepítve.
- Internetkapcsolat a csomagok letöltéséhez.
- Adminisztrátori (sudo) jogosultságok.

🔧 Fizikai beszerelés

1. Kapcsold ki a gépet, húzd ki a tápot.
2. Távolítsd el az előző videokártyát (ha volt).
3. Tedd be az RTX 4070-et a PCIe x16 foglalatba.
4. Csatlakoztasd a tápkábelt (1 db 8-pines vagy 2 db 6+2, a tápod támogatja).
5. Indítsd újra a szerveret.

1. Rendszer frissítése

A rendszer naprakészen tartása biztosítja, hogy a legfrissebb csomagokat és függőségeket használd.

Parancsok:

```
sudo apt update
sudo apt upgrade -y
```

Magyarázat:

- Az `apt update` frissíti a csomaglistát, az `apt upgrade` pedig telepíti a legújabb csomagverziókat.
- Ez segít elkerülni a kompatibilitási problémákat az NVIDIA driver és más eszközök telepítésekor.

2. NVIDIA driver és firmware telepítése

Az NVIDIA driver szükséges ahhoz, hogy a rendszer felismerje és használja az RTX 4070 GPU-t.

Parancsok:

```
sudo apt install -y nvidia-driver firmware-misc-nonfree nvidia-utils
```

Magyarázat:

- Az `nvidia-driver` biztosítja a GPU működéséhez szükséges drivert.
- A `firmware-misc-nonfree` tartalmazza a nem szabad szoftvereket, amelyek szükségesek lehetnek a GPU teljes funkcionalitásához.
- Az `nvidia-utils` olyan segédprogramokat tartalmaz, mint az `nvidia-smi`, amely a GPU állapotának ellenőrzésére szolgál.

Ellenőrzés:

Indítsd újra a rendszert, hogy a driver betöltődjön:

```
sudo reboot
```

Ellenőrizd a GPU működését:

```
nvidia-smi
```

Várt kimenet:

+-----+ NVIDIA-SMI 535.247.01 Driver Version: 535.247.01 CUDA Version: 12.2 +-----+-----+-----+-----+-----+-----+ GPU Name Persistence-M Bus-Id Disp.A Volatile Uncorr. ECC Fan Temp Perf Pwr:Usage/Cap Memory-Usage GPU-Util Compute M. MIG M. +-----+-----+-----+-----+-----+-----+ 0 NVIDIA GeForce RTX 4070 On 00000000:01:00.0 Off N/A 0% 45C P8 9W / 200W 4MiB / 12282MiB 0% Default N/A +-----+-----+-----+-----+-----+-----+									
---	--	--	--	--	--	--	--	--	--

Ha a fenti kimenetet látod, az NVIDIA driver sikeresen települt, és a GPU felismerésre került.

3. CUDA Toolkit telepítése

A CUDA Toolkit lehetővé teszi a GPU számítási képességeinek kihasználását az Ollama számára.

Parancsok:

```
sudo apt install -y nvidia-cuda-toolkit
```

Magyarázat:

- Az `nvidia-cuda-toolkit` biztosítja a CUDA könyvtárakat és eszközöket, amelyek szükségesek az Ollama GPU-támogatásához.
- Az RTX 4070 támogatja a CUDA 11.x és 12.x verziókat. A Debian 12 alapértelmezett tárolóiban lévő CUDA verzió általában kompatibilis.

Ellenőrzés:

Ellenőrizd a CUDA verziót:

```
nvcc --version
```

Várt kimenet (példa):

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2023 NVIDIA Corporation
Built on Tue_Aug_15_22:02:13_PDT_2023
Cuda compilation tools, release 12.2, V12.2.140
```

Ha a CUDA verzió megjelenik, a telepítés sikeres.

4. NVIDIA Container Toolkit telepítése

Az NVIDIA Container Toolkit lehetővé teszi, hogy a Docker konténerek hozzáférjenek a GPU-hoz.

Parancsok:

```
# Add hozzá az NVIDIA Container Toolkit tárolóját
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-toolkit.list | \
    sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg] https://#g' | \
    sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list

# Frissítsd a csomaglistát és telepítsd a toolkit-et
sudo apt update
sudo apt install -y nvidia-container-toolkit

# Konfiguráld a Dockert az NVIDIA runtime használatára
sudo nvidia-ctk runtime configure --runtime=docker
sudo systemctl restart docker
```

Magyarázat:

- Az első két parancs hozzáadja az NVIDIA Container Toolkit tárolóját a Debian csomagkezelőjéhez, és biztosítja a hitelesítést a GPG kulccsal.
- Az `nvidia-container-toolkit` telepítése lehetővé teszi a Docker számára a GPU-hozzáférést.
- Az `nvidia-ctk runtime configure` beállítja a Docker runtime-ot, hogy az NVIDIA GPU-t használja, és az újraindítás aktiválja a változásokat.

Ellenőrzés:

Teszteld, hogy a Docker látja-e a GPU-t:

```
docker run --rm --gpus all nvidia/cuda:12.0.0-base-ubuntu20.04 nvidia-smi
```

Várt kimenet:

A parancsnak meg kell jelenítenie az RTX 4070 kártyát, hasonlóan az `nvidia-smi` kimenetéhez a hoszt gépen. Ha hibaüzenetet kapsz (pl. "no GPU devices available"), ellenőrizd az NVIDIA driver és a toolkit telepítését.

5. Forráslista módosítása (opcionális)

Bizonyos esetekben szükséges lehet a Debian forráslista módosítása, hogy a `non-free` és `non-free-firmware` tárolók engedélyezve legyenek az NVIDIA driver és firmware telepítéséhez.

Parancsok:

1. Nyisd meg a forráslistát szerkesztésre:

```
sudo nano /etc/apt/sources.list
```

2. Győződj meg róla, hogy minden sor a következőkkel végződik:

Példa sor:

```
main contrib non-free non-free-firmware
```

```
deb http://deb.debian.org/debian bookworm main contrib non-free non-free-firmware
```

3. Mentsd el a fájlt, majd frissítsd a csomaglistát:

```
sudo apt update
```

Magyarázat:

- A `non-free` és `non-free-firmware` tárolók szükségesek lehetnek az NVIDIA proprietary driverek és firmware-ek telepítéséhez.
- Ha az `nvidia-driver` és `firmware-misc-nonfree` csomagok már települtek, ez a lépés valószínűleg nem szükséges, de érdemes ellenőrizni, ha problémák merülnek fel.

6. Opcionális lépés: Alapértelmezett NVIDIA runtime beállítása

A `/etc/docker/daemon.json` módosításának lépései

A megadott parancsok és a módosított `daemon.json` tartalom helyes. Az alábbiakban összefoglalom a lépéseket és magyarázatot adok hozzájuk:

Lépések:

1. **Nyisd meg a `/etc/docker/daemon.json` fájlt szerkesztésre:**

```
sudo nano /etc/docker/daemon.json
```

2. **Módosítsd vagy hozd létre a fájlt az alábbi tartalommal:**

```
{
  "iptables": true,
  "default-runtime": "nvidia",
  "runtimes": {
    "nvidia": {
      "path": "nvidia-container-runtime",
      "runtimeArgs": []
    }
  }
}
```

3. **Mentsd el a fájlt**, majd indítsd újra a Docker szolgáltatást:

```
sudo systemctl restart docker
```

4. **Ellenőrizd, hogy a Docker helyesen működik-e.**Várt kimenet: `nvidia`

```
docker info --format '{{.DefaultRuntime}}'
```

Magyarázat:

Ez a lépés biztosítja, hogy minden konténer alapértelmezés szerint az NVIDIA runtime-ot használja, ami leegyszerűsíti a GPU-támogatás konfigurálását több konténer esetén.

- `"iptables": true`: Ez a beállítás biztosítja, hogy a Docker kezelje az iptables szabályokat, ami a hálózati konfigurációhoz szükséges (ez a te eredeti beállításod).
- `"default-runtime": "nvidia"`: Beállítja az NVIDIA runtime-ot alapértelmezettként minden konténerhez.
- `"runtimes": { "nvidia": ... }`: Definiálja az NVIDIA runtime-ot, és megadja annak elérési útját (`nvidia-container-runtime`).

Fontos megjegyzés:

- A JSON formátumnak szintaktikailag helyesnek kell lennie. A te példád helyes, mivel a vesszőt (iptables után) megfelelően hozzáadtad.
- Ha a `/etc/docker/daemon.json` fájl nem létezett korábban, a fenti tartalom létrehoz egy új, érvényes konfigurációt.

Összegzés

- **Szükséges-e?** Nem, a te esetedben **opcionális**, mert az ollama konténer már expliciten használja az nvidia runtime-ot a Docker Compose-ban.

- **Mikor érdemes?** Ha több GPU-t használó konténert tervezel, vagy egységes GPU-kezelést szeretnél a rendszeredben.
- **Hogyan illeszkedik a leírásodhoz?** Hozzáadható opcionális lépésként a **4. NVIDIA Container Toolkit telepítése** után, hogy hosszú távon egyszerűsítsd a konfigurációt.
- **Ajánlás:** Mivel a rendszered már működik, nem szükséges a módosítás, de megtarthatod a daemon.json beállítást a jövőbeli rugalmasság érdekében. Ha megtartod, ellenőrizd, hogy a Docker és az Ollama továbbra is megfelelően működik.

7. Docker Compose konfiguráció beállítása

Az Ollama konténer konfigurálása a Docker Compose-ban biztosítja a GPU-támogatást és a megfelelő hálózati integrációt a webalkalmazással.

Konfiguráció:

Hozd létre vagy módosítsd a `docker-compose.yml` fájlt az alábbi tartalommal (csak az `ollama` szolgáltatás releváns része szerepel itt, a teljes fájlt lásd a korábbi üzenetekben):

```
services:
  ollama:
    image: ollama/ollama:latest
    container_name: ollama
    restart: unless-stopped
    ports:
      - "11434:11434"
    volumes:
      - ./ollama_data:/root/.ollama
    networks:
      - traefik-proxy
    environment:
      - OLLAMA_HOST=0.0.0.0:11434
      - OLLAMA_MAX_LOADED_MODELS=1
      - OLLAMA_KEEP_ALIVE=5m
      - OLLAMA_NUM_PARALLEL=1
      - NVIDIA_VISIBLE_DEVICES=all
      - NVIDIA_DRIVER_CAPABILITIES=compute,utility
    runtime: nvidia
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: all
              capabilities: [gpu]
    networks:
      traefik-proxy:
        external: true
```

Magyarázat:

- `image: ollama/ollama:latest` : A legfrissebb Ollama Docker képet használja.
- `ports: - "11434:11434"` : Az Ollama API portját elérhetővé teszi a hoszt gépen.
- `volumes: - ./ollama_data:/root/.ollama` : Perzisztens tárhelyet biztosít a letöltött modellek számára.
- `environment` : Környezetváltozók az Ollama konfigurálásához és a GPU-támogatáshoz.
 - `NVIDIA_VISIBLE_DEVICES=all` : Minden GPU-t elérhetővé tesz a konténer számára.
 - `NVIDIA_DRIVER_CAPABILITIES=compute,utility` : Engedélyezi a számítási és segédprogram funkciókat.

- `runtime: nvidia` : Expliciten az NVIDIA runtime-ot használja.
- `deploy.resources.reservations.devices` : Biztosítja, hogy a konténer hozzáférjen a GPU-hoz.

Futtatás:

Navigálj a `docker-compose.yml` fájlt tartalmazó könyvtárba, és indítsd el a konténert:

```
cd /mnt/raid/docker/ai
docker-compose up -d ollama
```

8. Ollama GPU-támogatás ellenőrzése

Ellenőrizd, hogy az Ollama valóban a GPU-t használja.

Parancsok:

1. Nézd meg az Ollama naplóit:

```
docker logs ollama
```

Várt kimenet: Keresd a "NVIDIA GPU detected" vagy hasonló üzenetet, amely megerősíti, hogy a GPU felismerésre került.

2. Lépj be az Ollama konténerbe:

```
docker exec -it ollama bash
```

3. Tölts le egy modellt (pl. Llama 3 8B, amely kompatibilis a 12 GB VRAM-mal):

```
ollama pull llama3:8b
```

4. Futtass egy tesztkérelést:

```
ollama run llama3
```

Írj be egy egyszerű promptot, pl.: `Is GPU enabled?`

5. Ellenőrizd a környezetváltozókat a konténerben:

```
env | grep NVIDIA
```

Várt kimenet:

```
NVIDIA_VISIBLE_DEVICES=all
NVIDIA_DRIVER_CAPABILITIES=compute,utility
```

6. Ellenőrizd a GPU használatot a hoszt gépen:

```
nvidia-smi
```

Amíg az Ollama fut, az `nvidia-smi` kimenetben látnod kell az Ollama folyamatokat (pl. `llama_server`) és a GPU memória használatát.

Magyarázat:

- A `llama3:8b` egy kvantált modell, amely kevesebb memóriát igényel, így az RTX 4070 12 GB VRAM-jával jól működik.
- Ha a naplókban nem látod a GPU használatára utaló jeleket, vagy az `nvidia-smi` nem mutat aktivitást, akkor a GPU-támogatás nem működik megfelelően.

9. Webalkalmazás és Ollama kommunikáció tesztelése

A webalkalmazásod (pl. [hertz-ai](#) szolgáltatás) az Ollama API-t használja. Ellenőrizd, hogy a kommunikáció működik-e.

Parancs:

Teszteld az Ollama API-t közvetlenül:

```
curl http://localhost:11434/api/generate -d '{"model": "llama3", "prompt": "Test GPU", "stream": false}'
```

Magyarázat:

- Ez a parancs egy egyszerű kérést küld az Ollama API-nak, és ellenőrzi, hogy a válasz megérkezik-e.
- Ha a válasz sikeres, az Ollama API megfelelően működik, és a webalkalmazásodnak is kommunikálnia kell vele.

10. Hibaelhárítási tippek

Ha a GPU-támogatás nem működik, vagy más problémák merülnek fel, próbáld meg az alábbi lépéseket:

1. Ellenőrizd a Docker verziót:

```
docker --version
```

Győződj meg róla, hogy a Docker 19.03 vagy újabb verziója van telepítve, mivel ez szükséges a GPU-támogatáshoz.

2. Frissítsd az Ollama képet:

```
docker pull ollama/ollama:latest
```

Biztosítsd, hogy a legfrissebb Ollama kép használd.

3. Naplók elemzése:

Nézd meg a konténerek naplóit hibák kereséséhez:

```
docker logs ollama
docker logs hertz-ai
```

4. Modell kompatibilitás:

Ha nagy modellt használasz (pl. Llama 3 70B), az nem férhet el a 12 GB VRAM-ban, és az Ollama CPU-ra vált. Használj kisebb, kvantált modellt:

```
docker exec -it ollama ollama pull llama3:8b
```

5. SELinux/AppArmor problémák:

Ha Debianon SELinux vagy AppArmor fut, ezek blokkolhatják a GPU hozzáférést. Ideiglenesen kapcsold ki őket teszteléshez:

```
sudo setenforce 0
```

6. Konténer újraindítása:

Ha módosítottad a [docker-compose.yml](#) fájlt, indítsd újra a konténereket:

```
cd /mnt/raid/docker/ai
docker-compose down
docker-compose up -d
```

11. Források és további olvasnivalók

- **NVIDIA Container Toolkit telepítési útmutató:** [NVIDIA Container Toolkit](#)

- **Ollama GPU-támogatás dokumentáció:** [Ollama GPU Docs](#)
 - **Docker Compose dokumentáció:** [Docker Compose](#)
-

Záró gondolatok

Ez az útmutató részletesen bemutatja, hogyan állíthatsz be egy Debian 12 rendszeren futó Docker környezetet az NVIDIA RTX 4070 GPU-val az Ollama AI modell futtatásához. A lépések követésével az Ollama sikeresen használja a GPU-t, ami jelentősen növeli az AI modellek inference sebességét. Ha további problémák merülnek fel, ellenőrizd a naplókat, és használd a hibaelhárítási tippeket. A konfiguráció rugalmas, és könnyen integrálható más webalkalmazásokkal (pl. Traefik-kal és Redis-szel).