Lokale KI mit Ollama + open-WebUI

Quelle: https://www.youtube.com/watch?v=vW29hgdb05I

abgeändert für AMD Ryzen AI Max+ 395 (dockerimage ollama/ollama:rocm und devices)

2026-03-24

als erstes amdgpu installieren mit

https://amdgpu-install.readthedocs.io/en/latest/install-overview.html#install-script

```
amdgpu-install
```

danach

nano /etc/default/grub

```
[...]
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash amdgpu.noretry=0"
[...]
```

```
sudo update-grub
sudo reboot
```

auf diese Änderung basiert die env-var HSA_XNACK: 1 in der folgenden docker-compose, erst damit klappte bei meinem Strix Halo die Erkennung der Graka

```
services:
  ollama:
    image:
        ollama/ollama:0.18.2-rocm
    restart: unless-stopped
    container_name: ollama
    environment:
      #OLLAMA_FLASH_ATTENTION: true
      # setting Context length to 8192 for longer data
      #OLLAMA_CONTEXT_LENGTH: 8192
      #HSA_OVERRIDE_GFX_VERSION: 11.5.1
      #HIP_VISIBLE_DEVICES: 0 # Standard-GPU (0 = erste GPU)
      #ROCR_VISIBLE_DEVICES: "34632"
      OLLAMA_DEBUG: 1
      HSA_XNACK: 1
    volumes:
      - ollama:/root/.ollama
    devices:
      - "/dev/kfd:/dev/kfd"
      - "/dev/dri:/dev/dri"
    group_add:
      - video
      #- render
    ports:
      - "11434:11434"



  open-webui:
    image: ghcr.io/open-webui/open-webui:latest
    container_name: open-webui
    volumes:
      - open-webui:/app/backend/data
    depends_on:
      - ollama
    ports:
      - 3000:8080
    environment:
      - WEBUI_URL=https://ki.MEINEDOMAIN.DE
      - ENABLE_OAUTH_PERSISTENT_CONFIG=false
      - OAUTH_PROVIDER_NAME=Authentik
```

```
      - OAUTH_CLIENT_ID=k4I...NF
      - OAUTH_CLIENT_SECRET=JNnJp31ZqcM5Uu...GWl7DHw5qCi0hz0
      - OPENID_PROVIDER_URL=https://auth.MEINEDOMAIN.DE/application/o/<MEIN-SLUG-NAME>/.well-known/openid-configuration
      - OPENID_REDIRECT_URI=https://ki.MEINEDOMAIN.DE/oauth/oidc/callback

      # Allows auto-creation of new users using OAuth. Must be paired with ENABLE_LOGIN_FORM=false.
      - ENABLE_OAUTH_SIGNUP=true

      # Disables user/password login form. Required when ENABLE_OAUTH_SIGNUP=true.
      # Nachtrag von Eike: Required stimmt nicht ganz, es geht auch ENABLE_OAUTH_SIGNUP=true UND ENABLE_LOGIN_FORM=true
      - ENABLE_LOGIN_FORM=true

      - OAUTH_MERGE_ACCOUNTS_BY_EMAIL=true
      - OAUTH_SCOPES=openid email profile
      - OLLAMA_BASE_URL=http://ollama:11434
      - WEBUI_SECRET_KEY=3ba8.....<openssl rand -hex 24>...3a70124
    extra_hosts:
      - host.docker.internal:host-gateway
    restart: unless-stopped

volumes:
  ollama:
  open-webui:
```

Mit Authentik verbinden (OIDC)

https://integrations.goauthentik.io/miscellaneous/open-webui/

https://docs.openwebui.com/features/access-security/auth/sso/

https://docs.openwebui.com/troubleshooting/sso/

Einfach provider in Authentik einstellen mit standard settings (email openid profile) und als redirect-uri strict die gleiche URI wie in der env var OPENID_REDIRECT_URI aus der docker-compose:

```
https://ki.MEINEDOMAIN.DE/oauth/oidc/callback
```

2026-03-22 (mit image von rjmalagon)

> Problem: ROCm im neuesten rjmalagon-image (v0.18.0.2) ist nicht rocm 7.x sondern 6.x enthalten.
>
> Das aktuellste ROCm-7 Image (optm-rocm7-latest) hat zwar rocm 7.x, ist allerdings zu alt, um neuere MoE Modelle zu unterstützen wie qwen3.5 oder nemotron-cascade-2. Lösung siehe oben mit ollama aktuell und HSA_XNACK=1

https://community.frame.work/t/running-ollama-in-docker-on-our-framework-desktop-using-the-gpu/75662/9 < ollama compose part aus diesem thread

https://github.com/rjmalagon/ollama-linux-amd-apu < dieses image

https://github.com/phueper/ollama-linux-amd-apu/tree/apu-optimizer < anderer fork

https://www.reddit.com/r/ollama/comments/1nt5fcr/how_do_i_get_ollama_to_use_the_igpu_on_the_amd_ai/

https://github.com/rjmalagon/ollama-linux-amd-apu/issues/24

https://github.com/ollama/ollama/pull/13000

docker-compose.yml

```
services:
  ollama:
    image:
        ghcr.io/rjmalagon/ollama-linux-amd-apu:optm-rocm7-latest
    restart: unless-stopped
    environment:
```

```
      OLLAMA_FLASH_ATTENTION: true
      OLLAMA_DEBUG: 0
      # setting Context length to 8192 for longer data
      OLLAMA_CONTEXT_LENGTH: 8192
    volumes:
      - ollama_storage:/root/.ollama
    devices:
      - "/dev/kfd:/dev/kfd"
      - "/dev/dri:/dev/dri"
    group_add:
      - video
    ports:
      - "11434:11434"

  open-webui:
    image: ghcr.io/open-webui/open-webui:latest
    container_name: open-webui
    volumes:
      - open-webui:/app/backend/data
    depends_on:
      - ollama
    ports:
      - 3000:8080
    environment:
      - 'OLLAMA_BASE_URL=http://ollama:11434'
      - 'WEBUI_SECRET_KEY='
    extra_hosts:
      - host.docker.internal:host-gateway
    restart: unless-stopped

volumes:
  ollama_storage:
  open-webui:
```

alt (mit ollama image)

```
services:
  ollama:
    # Uncomment below for GPU support
    # deploy:
    #   resources:
    #     reservations:
    #       devices:
    #         - driver: nvidia
    #           count: 1
    #           capabilities:
    #             - gpu

    # AMD GPU:
    devices:
      - /dev/kfd
      - /dev/dri
    volumes:
      - ollama:/root/.ollama
    # Uncomment below to expose Ollama API outside the container stack
    # ports:
    #   - 11434:11434
    container_name: ollama
    pull_policy: always
    tty: true
    restart: unless-stopped
    #image: ollama/ollama
    # AMD GPU:
    image: ollama/ollama:rocm
    #networks:
    #  - ollama-network



  open-webui:
#    build:
#      context: .
#      args:
#       OLLAMA_BASE_URL: '/ollama'
```

```
#      dockerfile: Dockerfile
    image: ghcr.io/open-webui/open-webui:latest
    container_name: open-webui
    volumes:
      - open-webui:/app/backend/data
    depends_on:
      - ollama
    ports:
      - 3000:8080
    environment:
      - 'OLLAMA_BASE_URL=http://ollama:11434'
      - 'WEBUI_SECRET_KEY='
    extra_hosts:
      - host.docker.internal:host-gateway
    restart: unless-stopped
    #networks:
    #  - ollama-network
    #  - proxy
    #labels:
    #  - "traefik.enable=true"
    #  - "traefik.docker.network=proxy"
    #  - "traefik.http.routers.ollama.entrypoints=http"
    #  - "traefik.http.routers.ollama.rule=Host(`ollama.jimsgarage.co.uk`)"
    #  - "traefik.http.middlewares.ollama-https-redirect.redirectscheme.scheme=https"
    #  - "traefik.http.routers.ollama.middlewares=ollama-https-redirect"
    #  - "traefik.http.routers.ollama-secure.entrypoints=https"
    #  - "traefik.http.routers.ollama-secure.rule=Host(`ollama.jimsgarage.co.uk`)"
    #  - "traefik.http.routers.ollama-secure.tls=true"
    #  - "traefik.http.routers.ollama-secure.tls.certresolver=cloudflare"
    #  - "traefik.http.routers.ollama-secure.service=ollama"
    #  - "traefik.http.services.ollama.loadbalancer.server.port=8080"

volumes:
  ollama:
  open-webui:

#networks:
  #ollama-network:
  #proxy:
  #  external: true
```

Revision #12
Created 2025-03-04 21:04:01 UTC
Updated 2026-03-29 21:53:38 UTC