Installation Ollama und OpenWebUI

1. Verzeichnisstruktur vorbereiten

Alle Containerdaten liegen unter `/opt/docker`, damit System und Daten sauber getrennt bleiben:

```
sudo mkdir -p /opt/docker/ollama
sudo mkdir -p /opt/docker/open-webui
```

Optional Zugriffsrechte setzen:

```
sudo chown -R kiadmin:docker /opt/docker
```

---

2. Docker-Compose Datei

Pfad: `/opt/docker/docker-compose.yml`

```
version: '3.8'

services:
  ollama:
    image: ollama/ollama
    container_name: ollama
    ports:
      - "11434:11434"
    restart: always
    runtime: nvidia
    volumes:
      - /opt/docker/ollama:/root/.ollama

  openwebui:
    image: ghcr.io/open-webui/open-webui:ollama
    container_name: open-webui
    restart: always
```

```
    runtime: nvidia
    ports:
      - "3000:8080"
    environment:
      - OLLAMA_API_BASE_URL=http://ollama:11434
    volumes:
      - /opt/docker/open-webui:/app/backend/data
      # Eigene CSS-Anpassungen (optional)
      # - /home/pleibling/docker/ai/custom.css:/app/build/static/custom.css:ro
    depends_on:
      - ollama
```

---

3. Container starten

```
cd /opt/docker
docker compose up -d
```

Status prüfen:

```
docker ps
```

- Ollama-API → `http://<Server-IP>:11434`
- OpenWebUI → `http://<Server-IP>:3000`

---

4. Modelle installieren

Ollama lädt Modelle nach Bedarf.
Installation erfolgt z. B. über:

```
docker exec -it ollama ollama pull gpt-oss:20b
```

Verwendete LLMs in dieser Umgebung:

- **GPT-OSS:20b** → Hauptmodell (groß, sehr leistungsfähig)
- **Llama 3.1:12b**
- **Mistral:7b**
- **Gemma3:12b**
- **DeepSeek-R1:8b**
- **Qwen3:14b**

Optional kannst du auch in OpenWebUI für jedes Modell eigene Presets/Workspaces definieren.

5. Update der Container

```
cd /opt/docker
docker compose pull
docker compose up -d
```

6. Backup-Hinweis

Da alle persistenten Daten in `/opt/docker/ollama` und `/opt/docker/open-webui` liegen, reicht ein Backup dieser Ordner.

Revision #2
Created 2025-09-23 06:37:52 UTC by Peter Leibling
Updated 2025-09-23 06:49:15 UTC by Peter Leibling