Installation Paperless und Paperless-AI

Verzeichnis anlegen (einmalig)

```
sudo mkdir -p /home/pleibling/docker/paperless/{data,media,export,consume,db,redis,ai}
sudo chown -R 1000:1000 /home/pleibling/docker/paperless
```

`docker-compose.yml` (unter `/home/pleibling/docker/paperless/docker-compose.yml`)

```
version: "3.8"

services:
  paperless-ngx:
    image: ghcr.io/paperless-ngx/paperless-ngx:latest
    container_name: paperless-ngx
    restart: always
    environment:
      - PAPERLESS_REDIS=redis://paperless-redis:6379
      - PAPERLESS_DBHOST=paperless-db
      - PAPERLESS_DBUSER=paperless
      - PAPERLESS_DBPASS=PapPW050725
      - PAPERLESS_SECRET_KEY=G2v3eKLZRIkpMeUcGkLor0Lt6vtzHodKLCRVvYHHjtE=
      - PAPERLESS_AI_ENABLED=1
      - PAPERLESS_AI_PROVIDER=ollama
      - PAPERLESS_AI_MODEL=llama3.1:8b
      - PAPERLESS_AI_HOST=http://192.168.33.200:11434
```

```
      - PAPERLESS_ALLOWED_HOSTS=dms.leibling.de,192.168.33.200,localhost
      -
PAPERLESS_CSRF_TRUSTED_ORIGINS=https://dms.leibling.de,http://192.168.33.200:8080,http://local
host:8080
      - USERMAP_UID=1000
      - USERMAP_GID=1000
      - PAPERLESS_TIKA_ENABLED=true
      - PAPERLESS_TIKA_ENDPOINT=http://tika:9998
      - PAPERLESS_CONVERT_OFFICE=true
      - PAPERLESS_TIKA_CONVERT_OFFICE=true
      - PAPERLESS_GOTENBERG_ENABLED=true
      - PAPERLESS_GOTENBERG_ENDPOINT=http://gotenberg:3000
      - RAG_SERVICE_ENABLED=true
      - RAG_SERVICE_URL=http://paperless-ai:8000
    ports:
      - "8080:8000"
    volumes:
      - /opt/docker/paperless/data:/usr/src/paperless/data
      - /opt/docker/paperless/media:/usr/src/paperless/media
      - /opt/docker/paperless/export:/usr/src/paperless/export
      - /opt/docker/paperless/consume:/usr/src/paperless/consume
    depends_on:
      - paperless-db
      - paperless-redis
  gotenberg:
    image: gotenberg/gotenberg:7
    container_name: paperless-gotenberg
    restart: always
    ports:
      - "3002:3000"
    security_opt:
      - no-new-privileges:true
    cap_drop:
      - ALL

  tika:
    image: apache/tika
    container_name: paperless-tika
    restart: always
    ports:
```

```
    - "3003:9998"
  security_opt:
    - no-new-privileges:true
  cap_drop:
    - ALL

paperless-db:
  image: postgres:15
  container_name: paperless-db
  restart: always
  environment:
    - POSTGRES_DB=paperless
    - POSTGRES_USER=paperless
    - POSTGRES_PASSWORD=PapPW050725
  volumes:
    - /opt/docker/paperless/db:/var/lib/postgresql/data

paperless-redis:
  image: redis:7
  container_name: paperless-redis
  restart: always
  volumes:
    - /opt/docker/paperless/redis:/data

paperless-ai:
  image: clusterzx/paperless-ai
  container_name: paperless-ai
  restart: always
  environment:
    - PUID=1000
    - PGID=1000
    - PAPERLESS_AI_PORT=3000
    - RAG_SERVICE_ENABLED=true
    - RAG_SERVICE_URL=http://paperless-ai:3000
  ports:
    - "3010:3000"
  volumes:
    - /opt/docker/paperless/ai:/app/data
  deploy:
    resources:
```

```
        reservations:
          devices:
            - capabilities: ["gpu"]
```

Wichtig:

- Deine **Ollama-Instanz** läuft in einem anderen Stack (Ollama/OpenWebUI). Zugriff erfolgt hier **per IP/Port** (`http://192.168.33.200:11434`) – das ist ideal, da unterschiedliche Compose-Netzwerke sich sonst nicht automatisch sehen.
- Wenn du später **TLS/Reverse Proxy** (z. B. Traefik/Caddy/Nginx) nutzt, passe `PAPERLESS_ALLOWED_HOSTS` und `PAPERLESS_CSRF_TRUSTED_ORIGINS` an.

Start/Update

```
cd /home/pleibling/docker/paperless
docker compose up -d
# Update:
docker compose pull && docker compose up -d
```

Kurzer Funktionstest

- Paperless-NGX: `http://<VM-IP>:8080`
- RAG-API (intern gemappt): `http://<VM-IP>:3010/health` → sollte `ok` liefern
- Tika: `http://<VM-IP>:3003/version`
- Gotenberg: `http://<VM-IP>:3002/health`

Revision #1
Created 2025-09-23 06:49:20 UTC by Peter Leibling
Updated 2025-09-23 06:51:16 UTC by Peter Leibling