Overleaf Setup

- Setup Overleaf on Docker Compose with all Bibtex

Setup Overleaf on Docker Compose with all Bibtex

Na klar! Hier ist die komplette, bereinigte Anleitung als Markdown-Datei. Du kannst den Text unten einfach kopieren und z.B. als `README.md` oder `INSTALL.md` in deinem Server-Ordner speichern.

□□ Setup-Guide: Overleaf Community Edition v5 (Docker Compose)

Diese Anleitung beschreibt die Installation der **Overleaf Community Edition (v5+)** auf einem eigenen Server mittels Docker Compose. Sie beinhaltet Lösungen für die MongoDB 8.0 Requirements, HTTP-Login-Probleme und die persistente Installation der vollen TeX-Live-Umgebung.

Voraussetzungen

- **Docker** & **Docker Compose** sind installiert.
- **Speicherplatz:** Mindestens **10–15 GB** frei (für `scheme-full`).
- **Zeit:** Ca. 1–2 Stunden (wegen des Downloads der TeX-Pakete).

---

1. Ordnerstruktur erstellen

Wir erstellen eine persistente Struktur, damit Datenbanken und die TeX-Installation auch nach Updates erhalten bleiben.

```
# Hauptordner erstellen
mkdir -p overleaf-server
cd overleaf-server

# Daten-Verzeichnisse anlegen
mkdir -p overleaf_data
mkdir -p mongo_data
mkdir -p redis_data
mkdir -p texlive_data
```

---

2. Die `docker-compose.yml`

Erstelle die Datei `docker-compose.yml`.

Wichtig: Ersetze `DEINE_SERVER_IP` bei `OVERLEAF_SITE_URL` mit der echten IP oder Domain deines Servers (z.B. `http://192.168.178.50`).

```
version: '3'

services:
  sharelatex:
    image: sharelatex/sharelatex:latest
    container_name: sharelatex
    restart: always
    depends_on:
      - mongo
      - redis
    ports:
      - "80:80"
    links:
      - mongo
      - redis
    volumes:
      # Overleaf Daten (Projekte, User)
      - ./overleaf_data:/var/lib/overleaf
      # WICHTIG: Persistentes Volume für die TeX-Installation (verhindert Neuinstallation bei
Neustart)
```

```
    - ./texlive_data:/usr/local/texlive
  environment:
    # --- App Konfiguration ---
    OVERLEAF_APP_NAME: Overleaf Community
    # MongoDB Replica Set Connection String (Pflicht für v5)
    OVERLEAF_MONGO_URL: mongodb://mongo:27017/sharelatex?replicaSet=rs0
    OVERLEAF_REDIS_HOST: redis
    REDIS_HOST: redis

    # --- Netzwerk & Login Fixes (für HTTP Nutzung) ---
    # Ersetze dies durch deine echte IP! Ohne das werden Cookies oft blockiert.
    OVERLEAF_SITE_URL: "http://DEINE_SERVER_IP"

    # Sicherheitchecks deaktivieren (da kein HTTPS/SSL aktiv ist)
    OVERLEAF_BEHIND_PROXY: "false"
    OVERLEAF_SECURE_COOKIE: "false"
    # Legacy-Variablen ebenfalls deaktivieren (wichtig für Login!)
    SHARELATEX_BEHIND_PROXY: "false"
    SHARELATEX_SECURE_COOKIE: "false"

    # Optional: Cookies toleranter machen
    OVERLEAF_SAME_SITE_COOKIE: "Lax"

    # Feature: User-Registrierung via E-Mail erlauben
    ENABLED_LINKED_FILE_TYPES: 'project_file,project_output_file'

mongo:
  # Overleaf v5 benötigt zwingend neuere MongoDB Versionen
  image: mongo:8.0
  container_name: mongo
  restart: always
  # Startet MongoDB als Replica Set (Pflicht für Overleaf v5)
  command: "--replSet rs0 --bind_ip_all"
  volumes:
    - ./mongo_data:/data/db

redis:
  image: redis:6.2
  container_name: redis
  restart: always
```

```
    volumes:
      - ./redis_data:/data
```

---

3. Erster Start & Datenbank-Initialisierung

Da MongoDB als Replica Set läuft, muss es **einmalig** aktiviert werden.

1. **Container starten:**

```
docker-compose up -d
```

(Warte ca. 10–20 Sekunden, bis die Container hochgefahren sind) 2. **Replica Set aktivieren:** Führe diesen Befehl aus, um die Datenbank zu initialisieren:

```
docker exec -it mongo mongosh --eval "rs.initiate({_id:'rs0', members:[{_id:0,
host:'mongo:27017'}]})"
```

Erwartete Ausgabe: `{ "ok" : 1 }` 3. **Overleaf neustarten:** Damit Overleaf die nun fertige Datenbank findet:

```
docker-compose restart sharelatex
```

---

4. TeX Live Installation (Persistent)

Das Docker-Image ist standardmäßig leer. Wir installieren TeX Live manuell in das persistente Volume.

A. Installer laden & Basis installieren

Logge dich in den Container ein:

```
docker exec -it sharelatex /bin/bash
```

Führe im Container folgende Befehle aus:

```
# 1. In Temp-Ordner wechseln und Installer laden
cd /tmp
wget https://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz

# 2. Entpacken
zcat < install-tl-unx.tar.gz | tar xf -
cd install-tl-*

# 3. Basis-Installation starten (dauert ca. 5 Min)
perl install-tl --scheme=basic --no-interaction

# 4. Pfade verknüpfen (WICHTIG!)
/usr/local/texlive/*/bin/*/tlmgr path add

# 5. Notwendiges Tool installieren
tlmgr install latexmk
```

B. Das "Full Scheme" installieren (Optional, aber empfohlen)

Um alle Pakete zu haben (ca. 4-5 GB Download), installiere `scheme-full`. Da dies lange dauert, nutze am besten `screen` auf deinem Host-Server, damit du SSH schließen kannst.

Verlasse den Container mit `exit`.

Auf dem Host-Server:

```
# 1. Screen Session starten
screen -S latex-install

# 2. Installation starten
docker exec sharelatex tlmgr install scheme-full
```

Tipp: Du kannst das Fenster nun mit `STRG+A` dann `D` in den Hintergrund schicken ("detachen"). Zurückholen mit `screen -r`.

---

5. Admin-User erstellen

Falls das Web-Formular unter `/launchpad` Probleme macht, erstelle den User über die Konsole:

```
docker exec -it sharelatex /bin/bash -c "cd /overleaf/services/web && node
modules/launchpad/bin/create-admin-user.js --email dein.name@email.com"
```

Kopiere den ausgegebenen Link in deinen Browser, um das Passwort zu setzen.

---

6. Abschluss & Aufräumen

Sobald die Installation von `scheme-full` durchgelaufen ist:

1. **Aufräumen:** Lösche die Installationsdateien im Container, um Platz zu sparen.

```
docker exec sharelatex rm -rf /tmp/install-tl-unx.tar.gz /tmp/install-tl-*
```

2. **Neustart:** Starte den Container einmal neu, um alle Caches zu leeren.

```
docker-compose restart sharelatex
```

Fertig! Dein Overleaf Server ist nun einsatzbereit.

Wartung & Updates

- **Overleaf Update (Web-Oberfläche):**

```
docker-compose pull
docker-compose up -d
```

- **LaTeX Pakete Update (Inhalt):**

```
docker exec -it sharelatex tlmgr update --self --all
```