



Installationsanleitung

IntelliDoc Software

Belbaraka, Ayoub

Bouaasria, Farah

Dablaq, Ilyass

Kilic, Miray-Eren

Moeller, Lennart

Neumann, Luca

Inhaltsverzeichnis

| | |
|---|---|
| Systemvoraussetzungen..... | 3 |
| Minimale Hardwareanforderungen: | 3 |
| Software-Voraussetzungen: | 3 |
| Ports und Netzwerk:..... | 3 |
| Verwalter Voraussetzungen: | 3 |
| Installation..... | 4 |
| Disclaimer | 4 |
| Installation auf Linux (Ubuntu)..... | 4 |
| Programmverwaltung nach Installation..... | 4 |
| Löschen/Deinstallation..... | 5 |
| Verwendung der Software | 6 |
| Nutzerrechte | 6 |
| Adminverwaltung | 7 |
| Dateien/Dateivorlagen | 8 |

Systemvoraussetzungen

Minimale Hardwareanforderungen

| | |
|---------------------------|---|
| CPU | 4 CPU-Kerne, x86_64 Arch. |
| RAM | Mind. 16 GB |
| Verfügbarer Speicherplatz | Mind. 20 GB, Zusätzlicher Speicherplatz für Nutzersateien |
| Netzwerk | Stabile Internetverbindung |

Software-Voraussetzungen

| | |
|----------------|--------------------------------------|
| Betriebssystem | Unabhängig, empfohlen Linux (Ubuntu) |
| Docker | Version 25.x.x |
| Docker Compose | Version 2.24.x |

Ports und Netzwerk

| | |
|---------------|-----------------------|
| Externe Ports | 3000 für http Zugriff |
| Interne Ports | 5432 für PostgreSQL |

Verwalter Voraussetzungen

| |
|---------------------------|
| Nutzer mit 'sudo' Rechten |
|---------------------------|

Installation

Disclaimer

Die Anleitung beschränkt sich auf die Installation auf einem Linux (Ubuntu) Server. Da das Programm durch Docker aber theoretisch auf allen Docker-fähigen Servern laufen kann, sind andere Betriebssysteme und Konfigurationen nicht ausgeschlossen, werden hier aber nicht aufgezählt.

Wir übernehmen keine Garantie für eine erfolgreiche Installation auf einem anderen Betriebssystem als Linux (Ubuntu).

Installation auf Linux (Ubuntu)

Wir empfehlen standard-cleanups durchzuführen, wie z.B. das System vorher zu updaten und vorige Docker Installationen ggf. zu löschen, um Versionskonflikte zu vermeiden.

1. Docker, sowie Docker Compose über offizielle Anleitung installieren

<https://docs.docker.com/engine/install/ubuntu/>

<https://docs.docker.com/compose/install/>

2. Basis-Setup für IntelliDoc

Für einige Befehle wird evtl. ein 'sudo' prefix vorausgesetzt, das hier zur besseren Lesbarkeit weggelassen wird.

```
# Hauptverzeichnis für IntelliDoc erstellen

mkdir intellidoc
cd intellidoc

# Basisdateien erstellen

touch .env
touch docker-compose.prod.yml

# Dateiinhalt bearbeiten (z.B. mit 'nano')
# Inhalte zum einfügen am Ende des Dokuments
nano .env
nano docker-compose.prod.yml
```

3. IntelliDoc via Docker installieren und starten

```
# Das IntelliDoc Docker image ziehen
docker pull ghcr.io/lemmiix/intellidoc:latest

# IntelliDoc starten
docker compose -f docker-compose.prod.yml up
# ggf. docker-compose -f docker-compose.prod.yml up -d
# falls der Prozess nicht im Vordergrund laufen und angezeigt werden soll
```

Programmverwaltung nach Installation

Der Container kann mit den standard Docker Befehlen gesteuert werden. Als Referenz dazu bitte Onlinere Ressourcen verwenden.

Löschen/Deinstallation

IntelliDoc spezifische Container entfernen

```
# 1. Container und deren zugehörige Volumes stoppen und entfernen
docker compose -f docker-compose.prod.yml down -v

# 2. Das spezifische Image entfernen
docker rmi ghcr.io/lemmiix/intellidoc:latest

# 3. Nicht mehr benötigte Volumes manuell entfernen (falls noch vorhanden)
docker volume rm postgres_data pip_cache venv_data

# 4. Aufräumen von ungenutzten Netzwerken (optional)
docker network prune -f

# 5. Lokale Build-Cache bereinigen (optional)
docker builder prune -f
```

Für das Entfernen von Docker/Docker Compose an sich, oder fremden Docker Containern, als Referenz dazu bitte Onlineressourcen verwenden.

Verwendung der Software

Zugriff auf die Webseite geschieht über

http://<public_server_ip>:3000

Bei der Erstellung werden bereits zwei Testnutzer angelegt:

| E-Mail | Passwort | Rechte |
|---------------|----------|-------------|
| admin@idoc.de | aaa | Adminrechte |
| user@idoc.de | aaa | Userrechte |

Nutzerrechte

Jeder Nutzer der sich registriert hat die Standardnutzerrechte, um sich einzuloggen und eigene Dateien und Dokumente zu verwalten.

Die Adminrolle hat zusätzlich Zugriff auf das Admin-Dashboard, um DB-Transaktionen und Statistiken zu überwachen und andere Nutzer zum Admin zu befördern.

Ausschließlich der Serveradministrator/Docker Container-Verwalter kann in der PostgreSQL Shell Nutzern die Adminrolle entziehen (mehr dazu im folgenden Punkt).

Adminverwaltung

Im Terminal während der Containerlaufzeit ausführen:

```
docker exec -it intellidoc-postgres-1 psql -U postgres
```

Dann, in der sich öffnenden Shell:

```
-- Nutzer zum 'Admin' befördern
INSERT INTO main.user_roles_mapping (user_id, role_id)
SELECT [user_id], role_id
FROM main.user_roles
WHERE role_name = 'admin';

-- Nutzer 'Admin'-Rolle entziehen
DELETE FROM main.user_roles_mapping
WHERE user_id = [user_id]
AND role_id = (SELECT role_id FROM main.user_roles WHERE role_name = 'admin');

-- Praktisches Beispiel, mit user_id = 1
-- Admin-Rolle hinzufügen
INSERT INTO main.user_roles_mapping (user_id, role_id)
SELECT 1, role_id FROM main.user_roles WHERE role_name = 'admin';

-- Admin-Rolle entfernen
DELETE FROM main.user_roles_mapping
WHERE user_id = 1
AND role_id = (SELECT role_id FROM main.user_roles WHERE role_name = 'admin');

-- Query-Erfolg überprüfen
-- Alle Rollen eines Nutzers anzeigen
SELECT u.user_name, r.role_name
FROM main.users u
JOIN main.user_roles_mapping m ON u.user_id = m.user_id
JOIN main.user_roles r ON m.role_id = r.role_id
WHERE u.user_id = [user_id];

-- Alle Nutzer mit Admin-Rolle anzeigen
SELECT u.user_name, u.email
FROM main.users u
JOIN main.user_roles_mapping m ON u.user_id = m.user_id
JOIN main.user_roles r ON m.role_id = r.role_id
WHERE r.role_name = 'admin';
```

Dateien/Dateivorlagen

.env:

```
VITE_BACKEND_URL=http://public_ip:3000 # public Server IP hier ersetzen
```

docker-compose.prod.yml:

```
version: '3.8'
services:
  postgres:
    image: pgvector/pgvector:pg16
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: pgres
      POSTGRES_DB: postgres
    volumes:
      - postgres_data:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U postgres"]
      interval: 5s
      timeout: 5s
      retries: 10
    networks:
      - default

  backend:
    image: ghcr.io/lemmiix/intellidoc:latest
    depends_on:
      postgres:
        condition: service_healthy
    environment:
      POSTGRES_SCHEMA: main
      NODE_ENV: production
      BACKEND_URL: ${BACKEND_URL:-http://localhost:3000}
    volumes:
      -
./node_modules/@xenova/transformers/models:/app/node_modules/@xenova/transformers/models
      - pip_cache:/pip_cache
      - venv_data:/app/venv
    ports:
      - "3000:3000"
    networks:
      - default
    deploy:
      resources:
        limits:
          memory: 2G
      restart: unless-stopped

networks:
  default:
    driver: bridge

volumes:
  postgres_data:
  pip_cache:
```


18. Dez. 2024



Belbaraka, Ayoub

Bouaasria, Farah

Dablaq, Ilyass

Kilic, Miray-Eren

Moeller, Lennart

Neumann, Luca