

Installation

n8n wird als Webseite bereitgestellt und auf einem Server gehostet oder aus der Cloud direkt im Browser ausgeführt.

Hier wird die Docker Installation beschrieben. Dabei wird PostgreSQL als Datenbank verwendet. Es kann auch statt einer PostgreSQL die integrierte SQLite verwendet werden, jedoch wird aus Performance Gründen PostgreSQL empfohlen.

Alle benötigten Dateien sind auch im GitHub von n8n verfügbar: [n8n GitHub - Config Files](#)

Folgend die Docker Compose Konfiguration. (`docker-compose.yml`)

Bitte die Pfade bei den beiden Volumes für die Daten anpassen.

```
services:
  n8n-db:
    image: postgres
    restart: always
    hostname: n8n-db
    container_name: n8n-db
    environment:
      - POSTGRES_USER
      - POSTGRES_PASSWORD
      - POSTGRES_DB
      - POSTGRES_NON_ROOT_USER
      - POSTGRES_NON_ROOT_PASSWORD
    volumes:
      - /path/to/n8n/data-db:/var/lib/postgresql/data
      - ./init-data.sh:/docker-entrypoint-initdb.d/init-data.sh
    healthcheck:
      test: ['CMD-SHELL', 'pg_isready -h localhost -U ${POSTGRES_USER} -d ${POSTGRES_DB}']
      interval: 5s
      timeout: 5s
      retries: 10
  n8n:
    image: docker.n8n.io/n8nio/n8n
```

```

restart: always
hostname: n8n
container_name: n8n
ports:
  - 5678:5678
volumes:
  - /path/to/n8n/data-n8n:/home/node/.n8n
environment:
  - DB_TYPE=postgresdb
  - DB_POSTGRESDB_HOST=n8n-db
  - DB_POSTGRESDB_PORT=5432
  - DB_POSTGRESDB_DATABASE=${POSTGRES_DB}
  - DB_POSTGRESDB_USER=${POSTGRES_NON_ROOT_USER}
  - DB_POSTGRESDB_PASSWORD=${POSTGRES_NON_ROOT_PASSWORD}
depends_on:
  n8n-db:
    condition: service_healthy

```

Zusätzlich wird noch eine `.env` Konfiguration für die Benutzer und Passwörter benötigt, welche wie folgt aussieht. Diese Datei bitte in den selben Ordner wie die Compose Konfiguration platzieren.

Bitte die Passwörter durch eigene und sichere Passwörter austauschen.

```

POSTGRES_USER=root
POSTGRES_PASSWORD=Ganz-Super-Dolles-Passwort
POSTGRES_DB=n8n
POSTGRES_NON_ROOT_USER=n8n
POSTGRES_NON_ROOT_PASSWORD=Weniger-Dolles-Passwort

```

Außerdem ist noch ein Shell-Skript hinzuzufügen, welches ebenfalls im selben Ordner liegen muss.

`(init-data.sh)`

```

#!/bin/bash
set -e;

if [ -n "${POSTGRES_NON_ROOT_USER:-}" ] && [ -n "${POSTGRES_NON_ROOT_PASSWORD:-}" ]; then
  psql -v ON_ERROR_STOP=1 --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" <<-EOSQL
  CREATE USER ${POSTGRES_NON_ROOT_USER} WITH PASSWORD '${POSTGRES_NON_ROOT_PASSWORD}';
  GRANT ALL PRIVILEGES ON DATABASE ${POSTGRES_DB} TO ${POSTGRES_NON_ROOT_USER};

```

```
GRANT CREATE ON SCHEMA public TO ${POSTGRES_NON_ROOT_USER};
EOSQL
else
echo "SETUP INFO: No Environment variables given!"
fi
```

Damit wären alle notwendigen Konfigurationen abgeschlossen und das Setup kann gestartet werden.

Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach gewählter Installation `sudo docker-compose up -d` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegebenen Optionen. Sollten die Container bereits mit dieser Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

Nach einigen Sekunden sollte n8n einsatzbereit sein. Die Seite von n8n ist über <http://localhost:5678> erreichbar.

Version #1

Erstellt: 3 Juli 2024 19:12:52 von Marcel

Zuletzt aktualisiert: 3 Juli 2024 19:24:28 von Marcel