



FF OPERATION

Anleitung

bis v1.0.x



Open Source Software
entwickelt durch JK Effects
von Julian Krauser
15. Februar 2025

Inhaltsverzeichnis

Inhaltsverzeichnis	II
1 Einleitung	1
2 Installation	2
2.1 Docker	2
2.2 Docker-Compose	2
2.3 Git	4
2.4 Konfiguration	5
2.5 Update der Version	6
2.6 Wechsel des Datenbanksystems	6
2.7 WebApp	6

1 Einleitung

FF Operation - Flexibles Einsatzmanagement für Feuerwehren und Vereine

FF Operation ist eine leistungsfähige Einsatzverwaltungssoftware und ein zentraler Bestandteil des FF-Ökosystems. Neben Feuerwehreinsätzen können auch Arbeitseinsätze von Vereinen erfasst werden. Durch den modularen Aufbau und frei definierbare Stammdaten kann die Software flexibel an unterschiedliche Organisationen angepasst werden.

FF Operation ermöglicht darüber hinaus die kollaborative Erstellung von Einsatzberichten. Daten können offline ohne Login erfasst und später nahtlos in ein Online-Formular übertragen werden, was eine flexible und effiziente Dokumentation gewährleistet.

Wird FF Admin eingesetzt, kann es als zentrale Datenquelle für Basisinformationen dienen und ein zentrales Login-Management übernehmen.

2 Installation

FF Operation kann über mehrere Wege betrieben werden. Zum einen werden Docker-Images versioniert zur Verfügung gestellt. Weiterhin kann auch das Release Projekt heruntergeladen und verwendet werden.

2.1 Docker

Disclaimer: Die Anleitung zum Betrieb von FF Operation mit Docker setzt Kenntnisse mit Docker und Docker-Compose voraus.

Die Docker-Images können gemeinsam über eine Compose-File konfiguriert und gestartet werden. Auch können die Images einzeln gestartet werden.

Die Docker-Images sind versioniert. Der `<tag>` des Images kann entweder `latest` für die neueste Version oder `vX.Y.Z` für eine bestimmte Version sein. Die Versionen können auch in den Releases der Repositories der Anwendungen nachgeschlagen werden. Dort lassen sich auch Informationen zu neuen Funktionen, Änderungen oder Fehlerbehebungen der jeweiligen Funktion finden.

2.2 Docker-Compose

Datenbank

Als Datenbank können MySQL, Postgres und SQLite verwendet werden. Postgres wird für den Produktiven Einsatz empfohlen.

Konfiguration von MySQL:

Anleitung zu FF Operation bis v1.0.x – Installation

```
1 ff-db:
2   image: mariadb:<version (bsp 11.2)>
3   container_name: ff_db
4   restart: unless-stopped
5   ports:
6     - "3306:3306"
7   environment:
8     - MYSQL_DATABASE=<database name>
9     - MYSQL_USER=<username>
10    - MYSQL_PASSWORD=<user password>
11    - MYSQL_ROOT_PASSWORD=<root password>
12   volumes:
13     - <volume|local path>:/var/lib/mysql
```

YAML

MYSQL_USER und MYSQL_PASSWORD sind optional. Werden diese nicht gesetzt, kann der Server entweder mit dem Nutzer `root` und dem gesetzten `MYSQL_ROOT_PASSWORD` Zugang zur Datenbank erhalten, oder es wird im nachhinein ein Nutzerzugang erstellt, der Zugriff auf die erstellte Datenbank hat. `MYSQL_DATABASE` erstellt direkt eine Datenbank, die durch einen angelegten `MYSQL_USER` verfügbar ist.

Konfiguration von Postgres:

```
1 ff-db:
2   image: postgres:<version (bsp 16)>
3   container_name: ff_db
4   restart: unless-stopped
5   ports:
6     - "5432:5432"
7   environment:
8     - POSTGRES_DB=<database name>
9     - POSTGRES_USER=<username>
10    - POSTGRES_PASSWORD=<user password>
11   volumes:
12     - <volume|local path>:/var/lib/postgresql/data
```

YAML

`POSTGRES_DB` erstellt direkt eine Datenbank, die durch einen angelegten `POSTGRES_USER` verfügbar ist.

Hinweis Wenn eine Docker-Compose Datei verwendet wird, kann zusätzliche ein Netzwerk angelegt werden. Dadurch ist das Veröffentlichen der Datenbank-Port-Exposes nicht mehr notwendig. Das Entfernen der port-Exposes verhindert den direkten Zugriff auf die Ports von außerhalb. Ergänzt muss hierfür das `network` und die Teilhabe des Backend-Containers am Netzwerk:

1. Ergänzung zu Server und Datenbank:

```
1 networks:
2   - ff_internal
```

Anleitung zu FF Operation bis v1.0.x – Installation

2. Ergänzung zur finalen Compose:

```
1 networks:
2   ff_internal:
```

3. Optionale Ergänzung zum Server:

```
1 depends_on:
2   - ff-db
```

Hierdurch kann der Server nicht starten, wenn die verwendete Datenbank nicht läuft.

2.3 Git

Eine Alternative zu Docker ist die direkte Ausführung der Anwendungen auf dem Server oder Desktop Gerät.

Hierzu müssen die App und der Server als Quellcode auf das System geladen und dort direkt verwendet werden.

Die Veröffentlichung der App und des Servers, damit diese aus dem Internet erreichbar sind, muss gesondert eingerichtet werden.

Das System muss NodeJs und die bevorzugte Datenbank installiert haben.

Für das Hosting von statischen Inhalten kann Apache oder Nginx verwendet werden. Eine Konfiguration für Nginx ist im Repo der App enthalten.

Die NodeJs Prozesse können auch durch Tools wie pm2 verwaltet werden.

Um die Konfiguration mittels ENV-Variablen an die Anwendungen weitergeben zu können, müssen .env Dateien erstellt werden. Hierzu kann die .env.example Datei kopiert und die definierten Werte ausgefüllt werden. Nicht benötigte Einträge sollten entfernt werden.

Die env-Datei im Frontend muss vor dem build-Prozess erstellt sein, da dort die Werte fest in den Code übernommen werden. Weiterhin muss die Datei im Frontend .env.production heißen. Die bestehende Datei kann modifiziert werden. Bei einer Änderung muss die App neu gebaut werden. Die env-Datei im Backend muss vor der Ausführung von `npm run start` angelegt sein. Bei einer Änderung der Einträge muss der Server lediglich neu gestartet werden.

App

Anleitung zu FF Operation bis v1.0.x – Installation

```
1 git clone https://forgejo.jk-effects.cloud/Ehrenamt/ff-operation.git
2 cd ff-operation
3 npm install
4 npm run build
```

Shell

Der durch `npm run build` erstellte `dist` Ordner kann mit Apache oder Nginx zur Verfügung gestellt werden.

Server

```
1 git clone https://forgejo.jk-effects.cloud/Ehrenamt/ff-operation-server.git
2 cd ff-operation-server
3 npm install
4 npm run build
5 npm run start
```

Shell

2.4 Konfiguration

Folgende Werte können zu einem Container konfiguriert werden:

Variable	Zweck	Fallback	optional
Database-Variablen			
MYSQL_DATABASE	Name der Datenbank, die bei Erstellung direkt angelegt wird.		✖
MYSQL_USER	Benutzername des Users, der bei Erstellung direkt angelegt wird.		✓
MYSQL_PASSWORD	Passwort zum User, das bei Erstellung gesetzt wird.		✓
MYSQL_ROOT_PASSWORD	Passwort für den User root, das bei Erstellung gesetzt wird.		✖
POSTGRES_DB	Name der Datenbank, die bei Erstellung direkt angelegt wird.		✖
POSTGRES_USER	Benutzername des Users, der bei Erstellung direkt angelegt wird.		✖
POSTGRES_PASSWORD	Passwort zum User, das bei Erstellung gesetzt wird.		✖

✖: Ein Fehlen dieser Variable verhindert das Starten der Anwendung! ⚠: Bei Verwendung von SQLite sind diese Variablen nicht notwendig!

Hinweis: Eine fehlerhafte Konfiguration der optionalen oder geforderten Variable verhindert das Starten der Anwendung.

2.5 Update der Version

Um eine Version auf eine Neuere zu aktualisieren, muss meist nur der Docker-Tag oder das Repo ersetzt werden.

Wer Docker mit `latest` nutzt, kann das neue Image direkt mit `docker pull` neu beziehen und dann den Container neustarten.

Informationen zu neuen Versionen können innerhalb der App unter `Benutzer > Version` oder in den Release-Pages gefunden werden.

Die Releases beinhalten Informationen zu einem Update und was zu beachten ist. So enthalten die Release-Informationen beispielsweise Vorbereitungen vor einem Update.

Bei Verwendung mittels Git, müssen die Repos neu bezogen werden. Anschließend müssen die Dependencies neu installiert und die Anwendungen neu gebaut werden.

2.6 Wechsel des Datenbanksystems

`AUTO RESTORE` ist standardmäßig aktiviert und ermöglicht die automatische Wiederherstellung des letzten Backups - sofern vorhanden - wenn die Datenbank beim Start des Servers leer ist.

Dies vereinfacht auch den Wechsel zwischen Datenbanken, da nur eine neue Verbindung zu einer anderen Datenbank aufgebaut werden muss und die Daten nach dem Neustart des Servers automatisch in die neue leere Datenbank übertragen werden.

2.7 WebApp

FF Operation ist als WebApp verfügbar. Dadurch lässt sich die Anwendung auf einem Smartphone oder Desktop über den Browser installieren.