

Betriebsdokumentation
Antrags- und Beschlussverwaltungstool

Inhaltsverzeichnis

1. Systemvoraussetzungen	1
1.1. Betriebssystem	1
1.2. Virtualisierung	1
1.3. Datenbank	1
1.4. Webserver	1
1.5. Software	1
2. Systemeinrichtung	2
3. Systembetreuung	3
3.1. Updates von Bibliotheken	3
3.2. Ausfallsicherheit	3

Kapitel 1. Systemvoraussetzungen

Die Mindestanforderungen der Hardware sind die gleichen wie die vom WSGI-kompatiblen Webserver und die vom Betriebssystem.

1.1. Betriebssystem

Hier ist es wichtig, dass ein beliebiges **UNIX**-System verwendet wird. Wir empfehlen **NixOS**, da wir damit die beste Erfahrung gemacht haben. Allerdings ist es auch möglich, das Tool auf einem anderen **UNIX**-System zu installieren.

1.2. Virtualisierung

Für das Deployment verwenden wir **Docker** sowie **docker-compose**. Diese müssen auf dem Zielsystem installiert sein.

1.3. Datenbank

Wir verwenden eine **PostgreSQL** Datenbank. Diese wird mittels der **docker-compose.yml** automatisch installiert und über **Django** konfiguriert.

1.4. Webserver

Es wird ein **WSGI-kompatibler Webserver** benötigt. Wir nutzen hier **Nginx**, da wir damit die beste Erfahrung gemacht haben. Allerdings ist es auch möglich, das Tool auf einem anderen Webserver wie **Apache** zu installieren.

1.5. Software

Damit alles ordnungsgemäß funktioniert, müssen folgende Softwarepakete installiert sein: * Python == 3.11.1 * Django == 4.2.2

Ebenfalls benötigen wir für die Generierung der Tagesordnung eine Etherpad-Instanz, welche vom Webserver aus erreichbar ist. Diese muss in der **settings.py** konfiguriert werden.

Kapitel 2. Systemeinrichtung

Die Befehle müssen über ein beliebiges UNIX-Terminal eingegeben werden.

1. Kclone das Git-Repository auf den Zielsever, auf welchem Docker & docker-compose installiert sind.
 - `git clone git@github.com:ABV-Tool/ABV_Tool.git`
2. Wechsle in den Ordner `src`:
 - `cd src/`
3. Starte die einzelnen Docker-Container (Datenbank, Django-Anwendung):
 - `docker-compose up -d`
4. Befülle die Datenbank mit den in `abv/Antragstool/fixtures` festgelegten Standard-Daten
 - `docker exec -i abv-tool bash < dbinit.sh`
5. Erstelle einen Antragsverwalter (superuser)
 - `docker exec -it abv-tool python3 abv/manage.py createsuperuser`
6. Die Anwendung ist nun einsatzbereit.

Kapitel 3. Systembetreuung

3.1. Updates von Bibliotheken

Bei der Installation können automatische Updates aktiviert werden. Außerdem sollte in regelmäßigen Abständen Updates von Python Paketen installiert werden. Dies ist mithilfe des Befehls `docker-compose build` möglich: dabei werden jeweils die neusten Pakete, welche in der `requirements.txt` hinterlegt sind, installiert.

Allerdings muss beachtet werden, dass hierbei keine Kompatibilitätsprobleme auftreten; andernfalls muss der Anwendungscode bearbeitet oder lediglich eine ältere Version der Pakete installiert werden.

3.2. Ausfallsicherheit

Die Uptime des Tools ist mit der Uptime des StuRa-Servers verbunden. Die Sicherheit der Daten musste nicht näher betrachtet werden, da wir eine Datenbank in PostgreSQL im ACID Format nutzen und der StuRa-Server tägliche Backups durchführt.