

I7 Teamverwaltung - Betriebsdokumentation

Übersicht

1. Quickstart Guide	1
2. Anforderungen	2
2.1. Hardware	2
2.2. Betriebssystem	2
2.3. Bibliotheken	2
2.4. SSL Zertifikat	3
3. Quellcode	3
4. Docker	4
4.1. Aufbau	4
4.2. Ausführen	5
5. Zugangsdaten	5
5.1. iseproject Server	5
5.2. MySQL	5
5.3. Django	6
6. Deinstallation	6

Hinweis: In diesem Dokument wird die Inbetriebnahme anhand des iseproject01 Servers beschrieben.

1. Quickstart Guide

Notwendige Pakete installieren:

```
sudo apt update
sudo apt install curl -y
sudo apt install docker -y
sudo curl -L "https://github.com/docker/compose/releases/download/v2.6.0/docker-
compose-linux-x86_64" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

Teamverwaltung Repository clonen und Leserechte freigeben:

```
cd ~
git clone git@github.com:davidkirchner/I7_Teamverwaltung.git
```

```
chmod 774 I7_Teamverwaltung
```

Docker Container starten:

```
cd I7_Teamverwaltung  
docker-compose up --build -d
```

Die Website sollte nun unter <https://iseproject01.informatik.htw-dresden.de> erreichbar sein.

2. Anforderungen

Das System ist ausschließlich auf den Betrieb im HTW-internen Netzwerk ausgelegt.

2.1. Hardware

Folgende Hardware Spezifikationen sind empfohlen:

- RAM: 4 GB oder mehr
- freier Massenspeicher: 20 GB oder mehr
- CPU: Intel Xeon X3430, vergleichbar oder besser
- Netzwerk: 50 MBit/s symm. oder mehr

2.2. Betriebssystem

- Ubuntu 20.04 LTS

Hinweis: andere Versionen werden mit sehr hoher Wahrscheinlichkeit unterstützt, sind aber ungetestet.

2.3. Bibliotheken

Folgende Pakete und Bibliotheken müssen installiert werden, um die Teamverwaltung zu starten:

- Docker v.20.10.12 oder neuer
erlaubt die Containerisierung der Anwendung

```
sudo apt install docker -y
```

- Curl v.7.68.0 oder neuer
zum Herunterladen von docker-compose benötigt

```
sudo apt install curl -y
```

- docker-compose v.2.6.0
ermöglicht die Konfiguration von komplexen Docker Containern und Netzwerken

```
sudo curl -L "https://github.com/docker/compose/releases/download/v2.6.0/docker-  
compose-linux-x86_64" -o /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose
```

2.4. SSL Zertifikat

Um die Login Funktionalität und verschlüsselte Kommunikation zu garantieren, ist ein gültiges SSL Zertifikat der HTW notwendig. Ohne ein solches von der HTW registrierte Zertifikat werden den Benutzern der Website zum einen Sicherheitswarnungen durch den Browser angezeigt und andererseits ist der Studenten-Login mittels s-Nummer nicht möglich. Dies kommt daher, dass Django die Certificate Authorities vom LDAP Server und seinem eigenen SSL Zertifikat auf Gleichheit überprüft, um sicherzugehen, dass das Zertifikat des LDAP Servers valide ist.

Das Zertifikat muss wie dargestellt abgelegt werden

```
I7_Teamverwaltung/  
├── docker/  
│   ├── nginx/  
│   │   ├── ssl/  
│   │   │   ├── server.crt  
│   │   │   └── server.key
```

Ein entsprechendes Zertifikat kann beim Rechenzentrum der HTW beantragt werden.

Für die Domain iseproject01.informatik.htw-dresden.de sind die notwendigen Dateien bereits im entsprechenden Directory vorhanden.

3. Quellcode

Der Quellcode für die Teamverwaltung ist im GitHub Repository unter folgendem Link verfügbar: https://github.com/davidkirchner/I7_Teamverwaltung

Diese Dateien und Verzeichnisse sind zum Starten der Anwendung notwendig:

```
I7_Teamverwaltung/  
├── src/  
├── docker/  
├── docker-compose.yml  
└── requirements.txt
```

4. Docker

4.1. Aufbau

Das Deployment wird mittels Docker und docker-compose umgesetzt. Dazu kommen 3 Container Images zum Einsatz:

- nginx - dient als reverse proxy
- db - stellt den MySQL Datenbank Server bereit
- django - stellt Django Backend bereit

Das Docker Netzwerk hat den Aufbau wie in Figure 1 dargestellt.

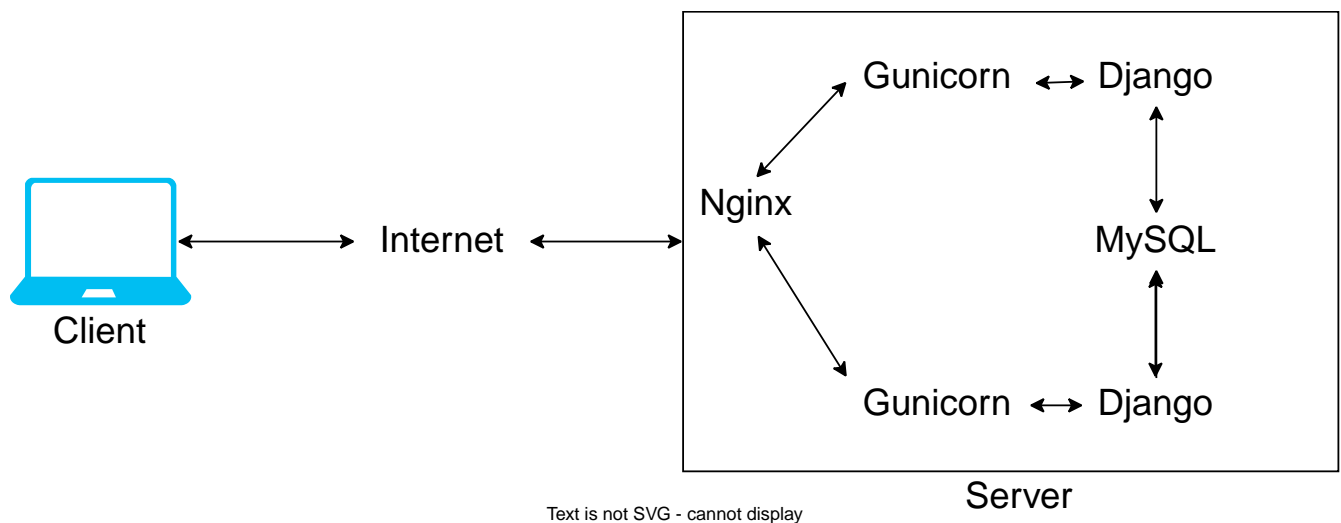


Figure 1. Docker Aufbau des Backends der Teamverwaltung

Nginx fungiert dabei als Reverse Proxy und stellt den Eintrittspunkt in das Docker-Netzwerk dar. Die Abkapselung durch ein virtuelles Docker Netzwerk mit Nginx als Eintrittspunkt für die Kommunikation mit dem Client erhöht die Sicherheit. Bei Anfragen vom Client zu Nginx über Port 80 (HTTP), antwortet Nginx automatisch mit dem Statuscode 301 und einer neuen URL, angepasst auf Port 443 (HTTPS). Der Client sendet anschließend seine Anfrage an die neue URL und die Kommunikation läuft dementsprechend verschlüsselt über HTTPS auf Port 443. Zusammengefasst bedeutet das, dass alle unverschlüsselten Anfragen über Port 80 direkt auf Port 443 umgeleitet werden, um verschlüsselte Kommunikation zwischen Client und Server zu garantieren.

Zur besseren Lastverteilung gibt es zwei django Container. Diese kommunizieren über das Web Server Gateway Gunicorn mit Nginx, welcher die Requests an die Django Server verteilt. Auf den Django Servern wird letztlich der Request ausgewertet. Die Django Server können dabei mit der MySQL Datenbank im db Container kommunizieren.

Beim Starten der Container wird der MySQL Server das SQL Script **I7_Teamverwaltung/docker/db/django.sql** ausführen. In diesem sind die nötigen Tabellendefinitionen gespeichert. Falls gewünscht, können dort auch weitere Daten hinzugefügt werden, die beim initialen Start geladen werden sollen.

4.2. Ausführen

Vor dem starten der Container muss für die Nutzergruppe "others" das Leserecht freigegeben werden, damit der Webserver die HTML Files korrekt an den Client ausliefern kann.

```
chmod 774 I7_Teamverwaltung
```

Mit dem Kommando

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

können die Docker Container im detached Modus hochgefahren werden. Das heißt sie werden abgekoppelt vom Terminal gestartet, sodass sie auch nach dem Schließen des Terminals weiterhin online bleiben. Bei Bedarf werden alle nötigen Images automatisch heruntergeladen.

Zum Stoppen der Container muss dieses Kommando ausgeführt werden:

```
docker-compose down
```

5. Zugangsdaten

5.1. iseproject Server

```
URL: iseproject01.informatik.htw-dresden.de  
Benutzer: seproject  
Passwort: 2N4PTE-86+8gbP
```

Der Standard SSH Port 22 wurde aus sicherheitstechnischen Gründen geschlossen. Der Server ist auf Port 14730 über SSH erreichbar. Mit folgendem Befehl kann die Verbindung aufgebaut werden:

```
ssh -p 14730 seproject@iseproject01.informatik.htw-dresden.de
```

5.2. MySQL

Die hier aufgelisteten Zugangsdaten sind für die MySQL Datenbank im Container "db".

```
Benutzer: root  
Passwort: 05912065344aae  
Rechte: alle Rechte auf allen Datenbanken
```

```
Benutzer: dev
Passwort: Er3dyWTM#a8_HE#
Rechte: alle Rechte auf der Datenbank "django"
```

Diese Passwörter werden aus dem **docker-compose.yml** File beim Starten der Container gelesen und können bei Bedarf angepasst werden. Bei Änderung ist zu beachten, dass im Django **settings.py** File die Zugangsdaten ebenfalls aktualisiert werden müssen.

5.3. Django

Die folgenden Zugangsdaten erlauben den Login auf der Website als Dozent.

```
Benutzer: dozent
Passwort: RzWJSQMbjwfv29Q
```

Das Passwort kann im Django-Admin Bereich unter "Users" mit einem Klick auf "dozent" im Abschnitt "password" geändert werden.

6. Deinstallation

Um die Teamverwaltung inklusiv zuvor geladener Pakete wieder zu deinstallieren, müssen die unten aufgelisteten Befehle ausgeführt werden.

```
sudo rm -r ~/I7_Teamverwaltung /usr/local/bin/docker-compose
sudo apt purge docker
sudo apt autoremove
```