

Installation via Docker

Beschreibung

Kimai via Docker mit https via Letsencrypt.

Installation

Vorbereitung

Sollte Apparmor enabled sein, wie das bei hetzner der Fall ist dann noch eben die Apparmor Pakete installieren

```
apt-get install apparmor apparmor-utils curl docker.io docker-compose
```

Als erstes erstellen wir unsere Verzeichnisse

```
mkdir -p /root/kimai/mariadb  
mkdir -p /root/kimai/data  
mkdir -p /root/kimai/plugins
```

Nun erstellen wir im Verzeichnis /root/kimai eine neue Datei namens .env

```
nano /root/kimai/.env
```

Inhalt:

```
# MySQL Credentials  
MYSQL_ROOT_PASSWORD=12345678  
MYSQL_USER=kimai  
MYSQL_PASSWORD=1234567890  
MYSQL_DATABASE=kimai  
  
# Volume directories
```

```
MARIADB_VOLUME_DIR=/root/kimai/mariadb
```

```
DATA_VOLUME_DIR=/root/kimai/data
```

Docker-compose Datei erstellen

```
nano /root/kimai/docker-compose.yml
```

Inhalt

```
version: '3.3'

services:
  mariadb:
    image: mariadb:latest
    restart: unless-stopped
    environment:
      MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
      MYSQL_USER: ${MYSQL_USER}
      MYSQL_PASSWORD: ${MYSQL_PASSWORD}
      MYSQL_DATABASE: ${MYSQL_DATABASE}
    volumes:
      - ${MARIADB_VOLUME_DIR}:/var/lib/mysql

  kimai:
    image: kimai/kimai2:apache
    restart: unless-stopped
    environment:
      - DATABASE_URL=mysql://${MYSQL_USER}:${MYSQL_PASSWORD}@mariadb:${MYSQL_DATABASE}
    volumes:
      - ${DATA_VOLUME_DIR}:/opt/kimai/var
      - ./plugins:/opt/kimai/var/plugins

  caddy:
    image: caddy:latest
    container_name: caddy-reverse-proxy
    ports:
      - "80:80"
      - "443:443"
```

```
volumes:
- ./Caddyfile:/etc/caddy/Caddyfile
- ./caddy_data:/data
- ./caddy_config:/config
restart: unless-stopped
```

Nun Caddyfile erstellen

```
nano /root/kimai/Caddyfile
```

Inhalt

```
zeit.example.tld {
  reverse_proxy http://kimai:8001
  tls info@example.tld {
    #ca https://acme-staging-v02.api.letsencrypt.org/directory
  }
}
```

Nun sicher stellen das die ports 80 und 443 auch über die Domain auf den Server erreichbar sind.

hier ein kleines Script das auf den port 80 lauscht um dies zu testen

```
nano /root/kimai/testeweb.sh
```

Inhalt

```
#!/bin/bash

# Funktion, die aufgerufen wird, wenn das Skript ein SIGINT-Signal (Ctrl+C) empfängt
cleanup() {
  echo "Server wird beendet..."
  exit 0
}

# Trap für das SIGINT-Signal einrichten
trap cleanup SIGINT
```

```
# Endlosschleife für den Server
while true; do
  # Listen on port 80
  { echo -ne "HTTP/1.1 200 OK\r\nContent-Length: $(echo -n "Hello, World!" | wc -c)\r\n\r\nHello, World!"; } | nc
-l -p 80 -q 1
done
```

Danach ausführbar machen

```
chmod +x /root/kimai/testeweb.sh
```

nun das Script starten, kann mit strg+c wieder abgebrochen werden.

Nun kann getestet werden ob Firewall/Portforwarding richtig funktioniert, indem in einem Webbrowser die public ip aufgerufen wird.

Es muss eine hello World Seite zurück gegeben werden.

Container Starten

Ins kimai Verzeichnis gehen

```
cd /root/kimai
```

Nun den Conatiner starten

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

Benutzer anlegen

Die Rollen

Role name	Description
User	Normal user can track their working times, see basic reports and change their own preferences. Technical name: <code>ROLE_USER</code>
Teamlead	Manages <u>teams</u> with permissions for invoices and access to all team timesheets. Technical name: <code>ROLE_TEAMLEAD</code>

Role name	Description
Administrator	Can manage all content and timesheet related data, but lack user administration and system privileges. Technical name: <code>ROLE_ADMIN</code>
System-Admin	Has permissions to manage everything in Kimai, from content to timesheets to users, plugins and system configurations. Technical name: <code>ROLE_SUPER_ADMIN</code>

die container id mit `docker ps` herausfinden

```
docker exec -ti <containerid> \  
/opt/kimai/bin/console kimai:user:create <username> <emailadresse> ROLE_SUPER_ADMIN
```

Beispiel:

```
docker exec -ti 32da986c57c2 \  
/opt/kimai/bin/console kimai:user:create admin admin@example.com ROLE_SUPER_ADMIN
```

Dann kann ein Kennwort vergeben werden. Muss mindestens 8 Zeichen haben

Benutzer Passwort ändern

die container id mit `docker ps` herausfinden

```
docker exec -ti <containerid> \  
/opt/kimai/bin/console kimai:user:password <username>
```

Beispiel:

```
docker exec -ti 32da986c57c2 \  
/opt/kimai/bin/console kimai:user:password admin
```

Dann kann ein neues Kennwort vergeben werden. Muss mindestens 8 Zeichen haben

Version #20

Erstellt: 3 Januar 2024 06:29:17 von Admin

Zuletzt aktualisiert: 5 Juni 2025 09:34:39 von Admin