CPSC 449 WEB BACK-END ENGINEERING PROJECT 7- FALL 2020

Project by:

- Ankita Udaykumar Kulkarni – CWID 887871861, Email - ankitak@csu.fullerton.edu
- Aditi Pratap Patil - CWID 887465649, Email - aditipatil138@csu.fullerton.edu

Project Description:
This is a web back-end project in which three microservices are created:

1. Users
2. Timeline
3. Worker

These microservices are created for a microblogging service. This project has used Python Flask, SQLite3 Database and Redis for implementing Messaging Queue. The WorkerService is responsible for implementing a messaging queue when a user accesses the /postTweet endpoint. Along with that, it also includes a hashtag analysis method. When a user accesses the /trending endpoint, he/she will get the top trending hashtags. The User microservice includes creating user, authenticating user, add followers, remove followers. Timeline microservice posts a post on timeline, displays post of the user, displays post of all the follower's user follows, Displays post of the all the users.

Steps to run the project:

To run this project on Tuffix OS-

1. Install python 3
   ```
   $ sudo apt-get install python3.8
   ```

2. Install pip
   ```
   $ sudo apt install python3-pip
   ```

3. Install Flak API and pugsql
   ```
   $ sudo apt update
   $ python3 -m pip install Flask-API pugsql
   ```

4. Install Foreman and HTTPie
   ```
   $ sudo apt install --yes ruby-foreman httpie
   ```

5. Install Redis
   ```
   $ sudo apt update
   $ sudo apt install --yes redis
   ```

   Then verify that Redis is up and running:
   ```
   $ redis-cli ping
   PONG
   ```

6. Install python libraries for Redis

```
$ sudo apt install --yes python3-hiredis
$ pip install redis
```

7. Install RQ

   ```
   $ sudo apt install --yes python3-rq
   ```

   Then verify that RQ is available:

   ```
   $ rq info
   0 queues, 0 jobs total

   0 workers, 0 queues

   Updated: 2020-11-23 14:24:53.725242
   ```

8. Clone the GitHub repository

   https://github.com/ankitakulkarnigit/messaging_queue_hashtag_analysis.git

9. cd to messaging_queue_hashtag_analysis dir and run following commands:

   ```
   $ export FLASK_APP=app.py
   $ export FLASK_ENV=development
   $ flask init
   $ foreman start
   ```

   After this we can see the three applications running on different ports.

10. On a second terminal run the following command

    ```
    $ rq worker --with-scheduler
    ```

    APIs can now be tested using the curl commands mentioned below or in WorkerService.py.

11. Run the following command to post a new Tweet

    ```
    $ curl -i -X POST -H 'Content-Type:application/json' -d '{"usernameAPI":"ankita", "tweetAPI":"#COVID-19 vaccine: #UK regulators warn people with history of 'significant' allergic reactions not to have #Pfizer"}' http://localhost:6200/postTweet;
    ```

12. To see top 25 trends

    ```
    $ curl -i -X GET http://localhost:6200/trending;
    ```

13. To see which is the last posted tweet

    ```
    $ curl -i -X GET http://localhost:6200/query;
    ```

14. Testing on Apache Benchmark Server for 5000 requests with 25 requests running concurrently

Synchronous Post Testing (Results found in sync_test.txt)

```
$ ab -n 5000 -c 25 -p post.json -T application/json -rk http://127.0.0.1:6100/postTweet > sync_test.txt
```

Asynchronous Post Testing (Results found in async_test.txt)

```
$ ab -n 5000 -c 25 -p post.json -T application/json -rk http://127.0.0.1:6200/postTweet > async_test.txt
```