

Major Project Report

Comprehensive Movie Reviews Analysis system using Microservice Architecture

Submitted by:

Kanishk Barhanpurkar
Nikita Mandlik
Pavan Gangareddy
Prashanth Devineni

Under Supervision of: Prof. Hui Lu
Course: Introduction to Cloud Computing (CS552)



State University of New York

May 2022 (Spring '22 Semester)

List of Contents

S.No.	Contents	Page Number
1.	Motivation	2
2.	Goals	2
3.	System Configuration	2
4.	Major issues	3
5.	Scoring Mechanism for movie verdict	3
6.	Snapshots of working system	4
7.	Conclusion	5
8.	Future Scope	5
9.	References	6

List of Figures

S.No.	Figure	Page Number
1.	Block diagram for entire system configuration.	2
2.	Data-flow diagram for the designed system.	3
3.	Web-page to input movie name.	4
4.	Web-page displaying movie information and verdict.	4
5.	Docker containers running independently on portainer.	5
6.	Docker containers running independently on Google CLI.	5

1. Motivation

The major inspiration of the big project is to gain the comprehensive knowledge of text-analytical systems deployed on cloud. Additionally, we learned the concept of several layer microservices architecture for NLP Application.

2. Goals

The main aim of the project is to develop a movie reviews based analysis system based on the services provided by Google Cloud. The Natural Language Processing application is one of the important buzz-word and the working of microservices can play an important role in the data-scraping, data-storage and NLP-based applications.

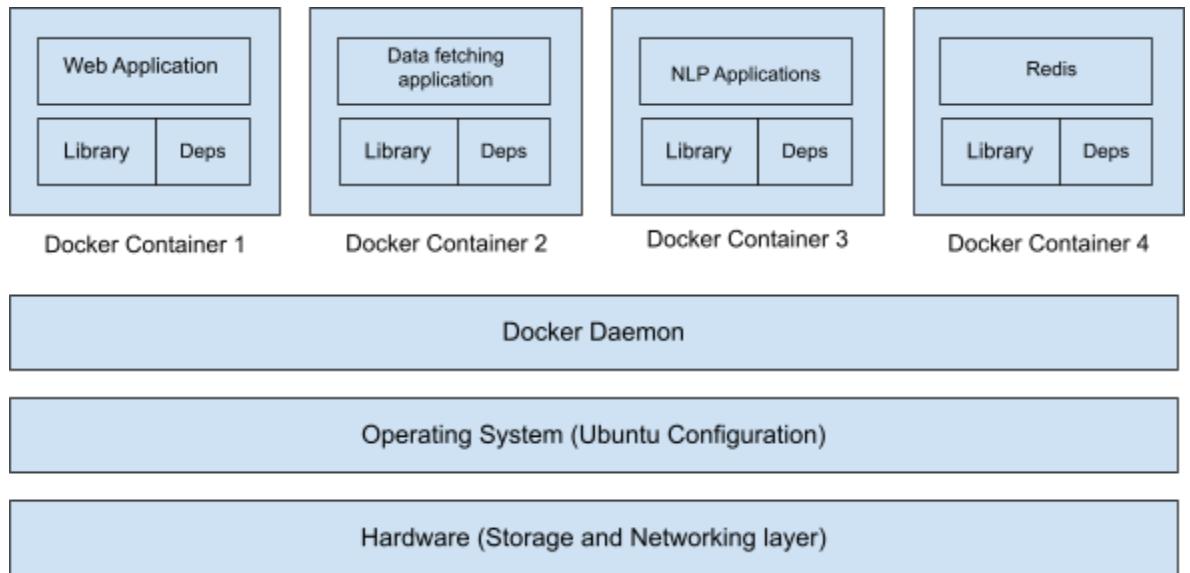


Figure 1: Block diagram for entire system configuration.

3. System Configuration

The entire system is based on the docker containers which are working simultaneously. In the first docker, we deployed the web applications (front-end) on the system. The second docker container contains the data fetching application where we are calling necessary IMDB-API services to fetch the data from the IMDB website which contains a lot of movie reviews. Additionally, in the third container and fourth container we are deploying the NLP sentiment analysis based services and Redis-based cache (Figure 1).

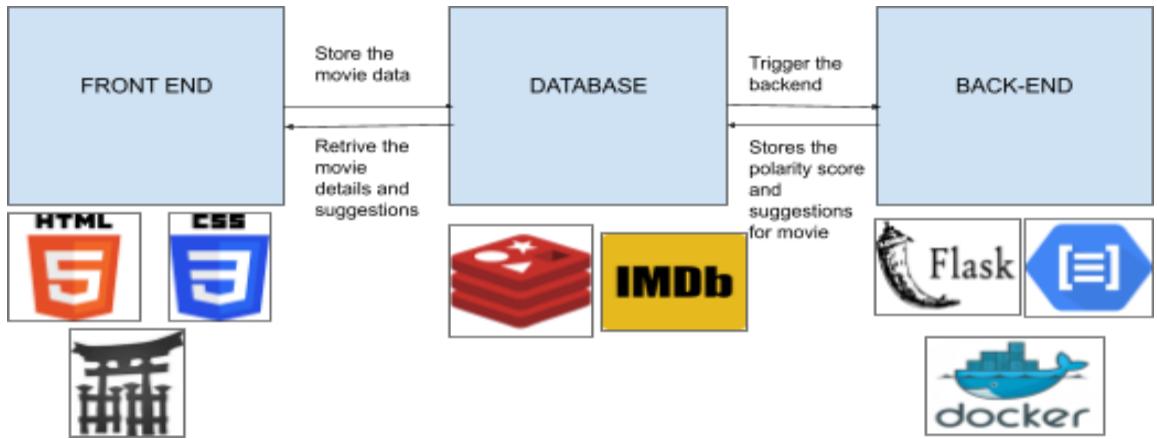


Figure 2: Data-flow diagram for the designed system.

The front-end consists of HTML, CSS and Jinja whereas we used Redis, a flexible data structure used for data-storage. Entire movie-based data is stored in it collected from the IMDB website. At the back-end we are using flask and google-cloud based services for calculating polarity score and magnitude. Additionally, the docker is used for creating containers and every container is isolated from each other (Figure 2).

4. Major issues

The above mentioned system is based on several containers working independently due to which we use triggers and flags to setting up the containers. Secondly, the integration of Google NLP services and IMDB API is important. We have try to implement it in one container and hence its not working properly due to which we have to use separate containers for both.

5. Scoring Mechanism for movie verdict

We have calculated the movie score based on the polarity of every review. Polarity determines the sentimental aspect of an opinion. The score ranges from -1 for extreme negative, +1 for the extreme positive and 0 for neutral. Magnitude is how much it is differing from the absolute neutral.

n:= number of reviews in one API calls.

p:= polarity score for every review.

M:=average score calculated for movie

$$M = \frac{\sum p}{n} \text{ ----- Equation 1}$$

6. Snapshots of working system



Figure 3: Web-page to input movie name.



Figure 4: Web-page displaying movie information and verdict.

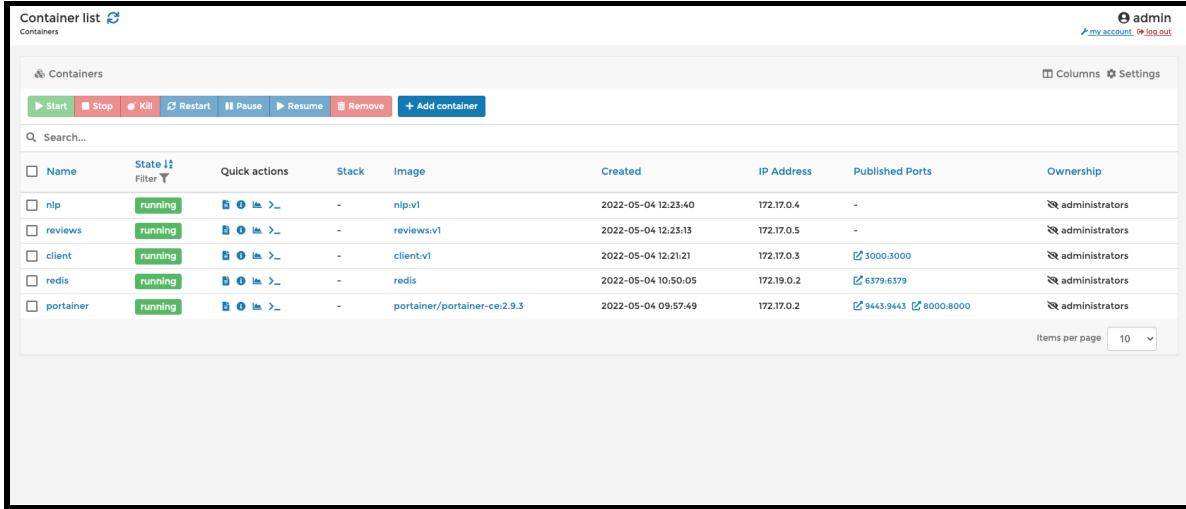


Figure 5: Docker containers running independently on portainer.

```
pavanreddy@final-project:~/CC-Project2-Final$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
1cfba996a230      nlp:v1            "sh run.sh"           41 minutes ago    Up 30 seconds
nlp
92de77b926af      reviews:v1        "python3 getMovieRev..."   42 minutes ago    Up 23 seconds
reviews
cbf487225d87      client:v1         "python3 getMovieNam..."   44 minutes ago    Up 30 seconds
client
9360eeeefdc0e     redis             "docker-entrypoint.s..."  2 hours ago       Up 2 hours
redis
5c4ef329508d      portainer/portainer-ce:2.9.3  "/portainer"          3 hours ago       Up 3 hours
portainer
pavanreddy@final-project:~/CC-Project2-Final$
```

Figure 6: Docker containers running independently on Google CLI.

7. Conclusion

The current system is developed to suggest movie recommendations based on the user comments on IMDB platform. We are using APIs to collect the movie reviews and movie trivia based on the IMDB public APIs. We used Redis, an in-memory data structure store for storing comments and movie information.

8. Future Scope

- To create a recommendation system based on the users previous search history.
- To include and apply other NLP-based APIs for different purposes like segmentation speech recognition.
- To increase the number of reviews for movie review analysis system
- To add more statistical parameters to improve the rating mechanism.
- We can use orchestration as it creates replicas for the containers and load balancing.

9. References

- [1] Tamrakar, S., Madhavi, B. K., & Mohan, V. (2022). Democratizing Sentiment Analysis of Twitter Data Using Google Cloud Platform and BigQuery. *Handbook of Intelligent Computing and Optimization for Sustainable Development*, 287-304.
- [2] Ghorbani, M., Ba Haghigat, M., Xin, Q., & Özen, F. (2020). ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing. *Journal of Cloud Computing*, 9(1), 1-12.
- [3] Brar, G. S., & Sharma, A. (2018). Sentiment analysis of movie review using supervised machine learning techniques. *International Journal of Applied Engineering Research*, 13(16), 12788-12791.
- [4] Redis Documentation. <https://redis.io/docs/>
- [5] Docker Documentation. <https://docs.docker.com/get-started/overview/>
