**CSE 586 DISTRIBUTED SYSTEMS PROJECT 2**

**Authors: -** Tanush Tripathi (50411177) and Rishi Joshi (50365611) (Team No 27)

**Date of Submission: -** 11th December 2021

**Introduction**

A publish-subscribe system is a system where publishers publish structured events to an event service and subscribers’ express interest in particular events through subscriptions which can be arbitrary patterns over the structured events.

Project 2 deals with implementing a publish subscribe system with Apacahe Kafka. Kafka is used to create multiple brokers that act as an intermediary between publishers and subscribers.

**Architectural Diagram**

The architectural diagram of our system is as follows:-

**Diagram

Description automatically generated**

**Technologies Used**

We have used the following technologies for this phase: -

* Python
* Apache Kafka
* Docker
* IMDb API

**Requirements and Dependencies**

The requirements and dependencies are as follows: -

* Kafka version: - 2.12-2.8.1
* Docker version: - 4.3.0
* Python version: - 3.7
* KafkaPython
* Kafka image: - <https://hub.docker.com/r/confluentinc/cp-kafka>
* Zookeeper: - <https://hub.docker.com/r/confluentinc/cp-zookeeper>

**Implementation**

We have taken the following steps as our implementation approach: -

1. Use the IMDb API for accessing the IMDb Movies Database.
2. Write a python implementation for the publisher that uses the IMDb API to get the data.
3. Use the Kafka Producer API to implement the publisher. Publish the data in real time every 2 seconds.
4. Implement 3 brokers using Apache Kafka. Configure the topics, partitions and replication factor as needed.
5. Write a python implementation for the subscriber that sends the subscription request to Kafka broker.
6. Use the Kafka Consumer API to implement the subscriber. Start getting real time updates.
7. Write the docker-compose.yaml and docker files in order to dockerize the system

**How to deploy**

Deploy the application with the following steps:-

* Navigate to 27\_50411177\_p2\Source\_Code\kafka\_pubsub
* Run the command docker-compose build
* Run the command docker-compose up
* After brokers are up and running go to localhost:8080 and subscribe to topics
* Once subscribed, the data is visible in real time

**Contribution**

Tanush Tripathi: - 50%

Rishi Joshi: - 50%