

# SDN PROJECT REPORT

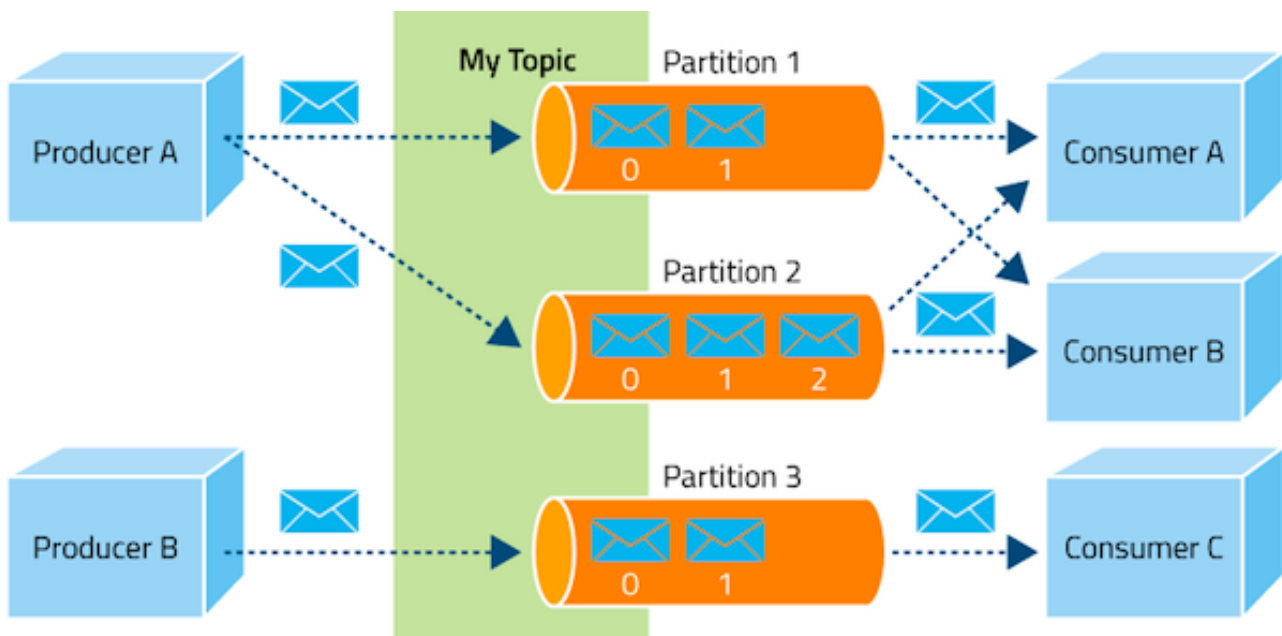
Team: Divyanshu Khandelwal (IMT2016065)  
Vaibhav Kumar (IMT2016086)

## AIM

Developing Kafka deployment and publisher/consumer for ReST based services.

## INTRODUCTION OF KAFKA

- **Kafka** is a distributed publish-subscribe messaging system that maintains feeds of messages in partitioned and replicated topics. In the simplest way there are three players in the Kafka ecosystem: producers, topics (run by brokers) and consumers.
- **Producers** produce messages to a topic of their choice. It is possible to attach a key to each message, in which case the producer guarantees that all messages with the same key will arrive to the same partition.



- **Consumers** read the messages of a set of partitions of a topic of their choice at their own pace. If the consumer is part of a consumer group, i.e. a group of consumers subscribed to the same topic, they can commit their

offset. This can be important if you want to consume a topic in parallel with different consumers.

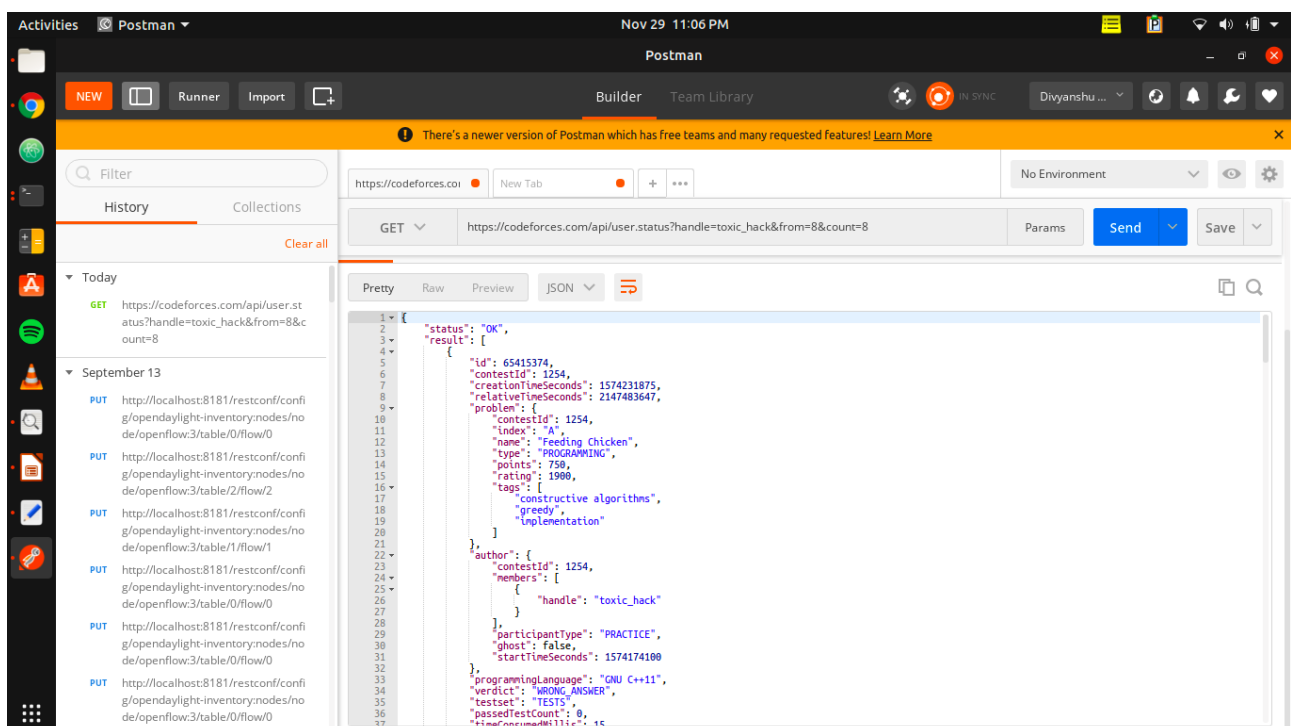
- **Topics** are logs that receive data from the producers and store them across their partitions. Producers always write new messages at the end of the log.
- **Zookeeper** is a service that helps Kafka to maintain the information about the current state of the consumers and helps in the management of coordination of Kafka system.

## DEPLOYMENT STEPS

1. Using application of Kafka messaging system, where data is from Codeforces, which is efficiently distributed across 3 different consumers at the same time.
2. Installing python's kafka library using **`pip install kafka-python`**.
3. Zookeeper bundle was downloaded and the server was started.
4. Kafka bundle was downloaded from [here](#) and then we used the command **`bin/kafka-server-start.sh config/server.properties`** to begin the services.

## DATA EXTRACTION

- Our project takes in user's real time submission data sends it to the consumers in our Kafka.
- The Following Dataset was chosen because exposure of its data is through ReST API and also the factor that it provides a huge steam.

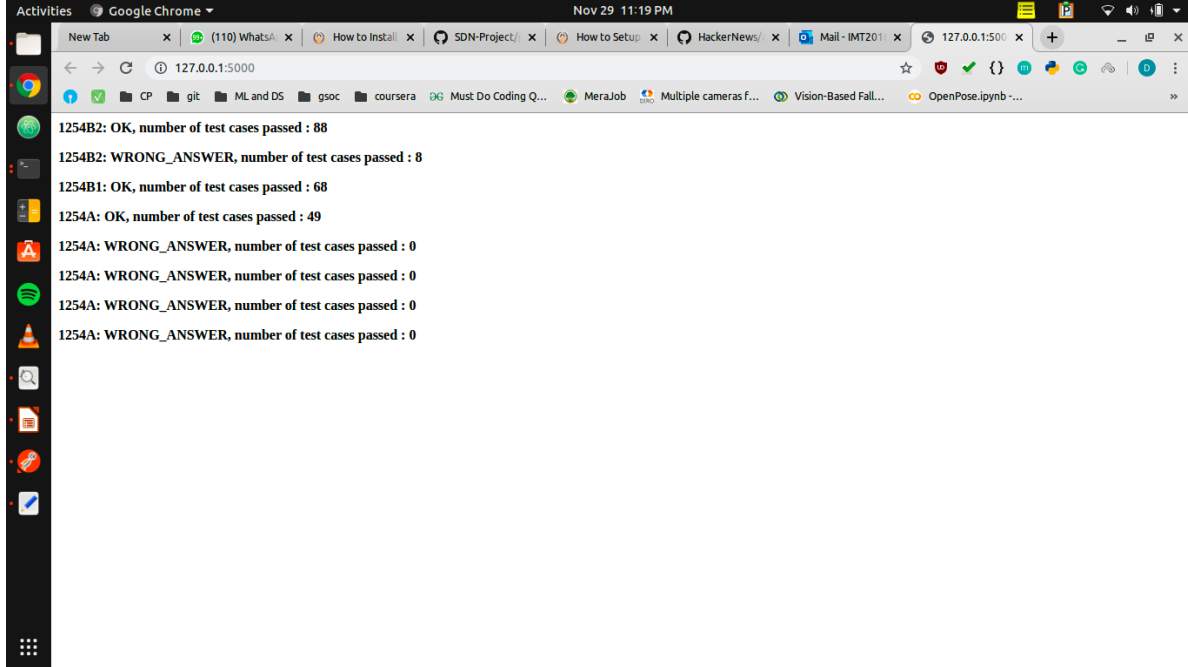


```
Activities Terminal Nov 29 10:01 PM
divyansh@devilizer: /media/divyansh/New Volume/Semester_7/SDN/SDN_kafka-Project

divyansh@devilizer: /media/divyansh/New Volume/Semester_7/SDN/SDN_kafka-Project$ python producer.py
{"status": "OK", "result": [{"id": 65419572, "contestId": 1254, "creationTimeSeconds": 1574238289, "relativeTimeSeconds": 2147483647, "problem": {"contestId": 1254, "index": "B2", "name": "Send Boxes to Alice (Hard Version)", "type": "PROGRAMMING", "points": 750.0, "rating": 2100, "tags": ["constructive algorithms", "greedy", "math", "number theory", "ternary search", "two pointers"]}, "author": {"contestId": 1254, "members": [{"handle": "toxic_hack"}]}, "participantType": "PRACTICE", "ghost": false, "startTimeSeconds": 1574174100, "programmingLanguage": "GNU C++11", "verdict": "OK", "testset": "TESTS", "passedTestCount": 88, "timeConsumedMillis": 873, "memoryConsumedBytes": 16076800}]}
{"status": "OK", "result": [{"id": 65418805, "contestId": 1254, "creationTimeSeconds": 1574237243, "relativeTimeSeconds": 2147483647, "problem": {"contestId": 1254, "index": "B2", "name": "Send Boxes to Alice (Hard Version)", "type": "PROGRAMMING", "points": 750.0, "rating": 2100, "tags": ["constructive algorithms", "greedy", "math", "number theory", "ternary search", "two pointers"]}, "author": {"contestId": 1254, "members": [{"handle": "toxic_hack"}]}, "participantType": "PRACTICE", "ghost": false, "startTimeSeconds": 1574174100, "programmingLanguage": "GNU C++11", "verdict": "WRONG ANSWER", "testset": "TESTS", "passedTestCount": 8, "timeConsumedMillis": 30, "memoryConsumedBytes": 16076800}]}
{"status": "OK", "result": [{"id": 65418759, "contestId": 1254, "creationTimeSeconds": 1574237178, "relativeTimeSeconds": 2147483647, "problem": {"contestId": 1254, "index": "B1", "name": "Send Boxes to Alice (Easy Version)", "type": "PROGRAMMING", "points": 500.0, "rating": 2000, "tags": ["constructive algorithms", "greedy", "math", "number theory", "ternary search", "two pointers"]}, "author": {"contestId": 1254, "members": [{"handle": "toxic_hack"}]}, "participantType": "PRACTICE", "ghost": false, "startTimeSeconds": 1574174100, "programmingLanguage": "GNU C++11", "verdict": "OK", "testset": "TESTS", "passedTestCount": 68, "timeConsumedMillis": 78, "memoryConsumedBytes": 16076800}]}]
```

```
Activities Terminal Nov 29 10:01 PM
divyansh@devilizer: /media/divyansh/New Volume/Semester_7/SDN/SDN_kafka-Project

divyansh@devilizer: /media/divyansh/New Volume/Semester_7/SDN/SDN_kafka-Project$ python consumer1.py
The recieved user submission data is {"status": "OK", "result": [{"id": 65415880, "contestId": 1254, "creationTimeSeconds": 1574232664, "relativeTimeSeconds": 2147483647, "problem": {"contestId": 1254, "index": "A", "name": "Feeding Chicken", "type": "PROGRAMMING", "points": 750.0, "rating": 1900, "tags": ["constructive algorithms", "greedy", "implementation"]}, "author": {"contestId": 1254, "members": [{"handle": "toxic_hack"}]}, "participantType": "PRACTICE", "ghost": false, "startTimeSeconds": 1574174100, "programmingLanguage": "GNU C++11", "verdict": "OK", "testset": "TESTS", "passedTestCount": 49, "timeConsumedMillis": 62, "memoryConsumedBytes": 307200}]}
The recieved user submission data is {"status": "OK", "result": [{"id": 65419572, "contestId": 1254, "creationTimeSeconds": 1574238289, "relativeTimeSeconds": 2147483647, "problem": {"contestId": 1254, "index": "B2", "name": "Send Boxes to Alice (Hard Version)", "type": "PROGRAMMING", "points": 750.0, "rating": 2100, "tags": ["constructive algorithms", "greedy", "math", "number theory", "ternary search", "two pointers"]}, "author": {"contestId": 1254, "members": [{"handle": "toxic_hack"}]}, "participantType": "PRACTICE", "ghost": false, "startTimeSeconds": 1574174100, "programmingLanguage": "GNU C++11", "verdict": "OK", "testset": "TESTS", "passedTestCount": 88, "timeConsumedMillis": 873, "memoryConsumedBytes": 16076800}]}
The recieved user submission data is {"status": "OK", "result": [{"id": 65418805, "contestId": 1254, "creationTimeSeconds": 1574237243, "relativeTimeSeconds": 2147483647, "problem": {"contestId": 1254, "index": "B2", "name": "Send Boxes to Alice (Hard Version)", "type": "PROGRAMMING", "points": 750.0, "rating": 2100, "tags": ["constructive algorithms", "greedy", "math", "number theory", "ternary search", "two pointers"]}, "author": {"contestId": 1254, "members": [{"handle": "toxic_hack"}]}, "participantType": "PRACTICE", "ghost": false, "startTimeSeconds": 1574174100, "programmingLanguage": "GNU C++11", "verdict": "WRONG ANSWER", "testset": "TESTS", "passedTestCount": 8, "timeConsumedMillis": 30, "memoryConsumedBytes": 16076800}]}
The recieved user submission data is {"status": "OK", "result": [{"id": 65418759, "contestId": 1254, "creationTimeSeconds": 1574237178, "relativeTimeSeconds": 2147483647, "problem": {"contestId": 1254, "index": "B1", "name": "Send Boxes to Alice (Easy Version)", "type": "PROGRAMMING", "points": 500.0, "rating": 2000, "tags": ["constructive algorithms", "greedy", "math", "number theory", "ternary search", "two pointers"]}, "author": {"contestId": 1254, "members": [{"handle": "toxic_hack"}]}, "participantType": "PRACTICE", "ghost": false, "startTimeSeconds": 1574174100, "programmingLanguage": "GNU C++11", "verdict": "OK", "testset": "TESTS", "passedTestCount": 68, "timeConsumedMillis": 78, "memoryConsumedBytes": 16076800}]}]
```



## CONSUMERS

- **Consumer 1**

Consumer simply prints the data that is extracted from the producer.

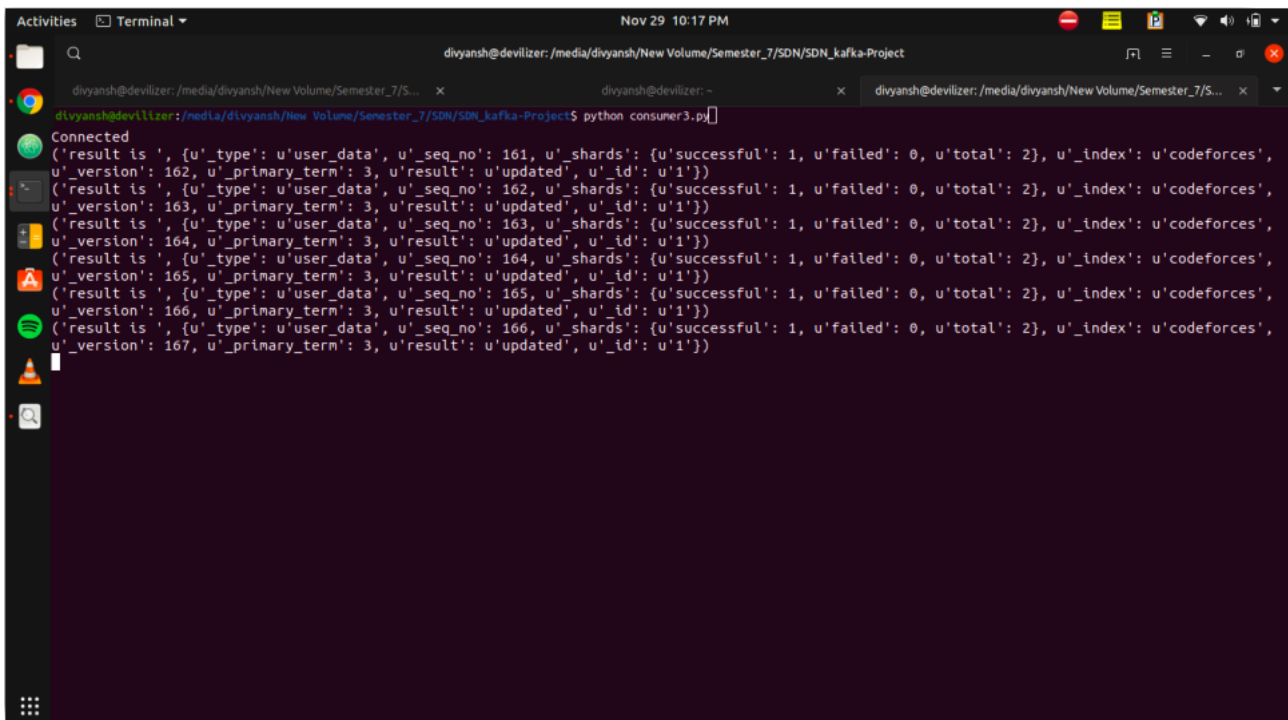
- **Consumer 2**

consumer uses Python's Flask server as a remote application. The server hosts a website which displays the verdict of the user's submission on Codeforces.

- **Consumer 3**

Consumer will add the collected data from producer to a **ElasticSearch Database**.

- Elasticsearch is a full-text, distributed NoSQL database i.e. uses documents instead of tables
- open source tool that allows for real-time searching and analyzing of your data and provides with dynamic content.
- ElasticDB can be downloaded from the website, and server is started using [./bin/elasticsearch](#)

A terminal window titled 'Terminal' with a timestamp of 'Nov 29 10:17 PM'. The terminal shows a user running a Python script 'python consumer3.py' in a directory path. The output displays a series of Kafka messages being processed, each containing a result status, message type, sequence number, shard information, index name, and version. The messages are for 'u' type data, with sequence numbers ranging from 161 to 167, and are indexed under 'u:codeforces'.

```
divyansh@devilizer: /media/divyansh/New Volume/Semester_7/SDN/SDN_kafka-Project$ python consumer3.py
Connected
('result is ', {'u_type': 'u:user_data', 'u_seq_no': 161, 'u_shards': {'u_successful': 1, 'u_failed': 0, 'u_total': 2}, 'u_index': 'u:codeforces',
u_version': 162, 'u_primary_term': 3, 'u_result': 'u:updated', 'u_id': 'u:1'})
('result is ', {'u_type': 'u:user_data', 'u_seq_no': 162, 'u_shards': {'u_successful': 1, 'u_failed': 0, 'u_total': 2}, 'u_index': 'u:codeforces',
u_version': 163, 'u_primary_term': 3, 'u_result': 'u:updated', 'u_id': 'u:1'})
('result is ', {'u_type': 'u:user_data', 'u_seq_no': 163, 'u_shards': {'u_successful': 1, 'u_failed': 0, 'u_total': 2}, 'u_index': 'u:codeforces',
u_version': 164, 'u_primary_term': 3, 'u_result': 'u:updated', 'u_id': 'u:1'})
('result is ', {'u_type': 'u:user_data', 'u_seq_no': 164, 'u_shards': {'u_successful': 1, 'u_failed': 0, 'u_total': 2}, 'u_index': 'u:codeforces',
u_version': 165, 'u_primary_term': 3, 'u_result': 'u:updated', 'u_id': 'u:1'})
('result is ', {'u_type': 'u:user_data', 'u_seq_no': 165, 'u_shards': {'u_successful': 1, 'u_failed': 0, 'u_total': 2}, 'u_index': 'u:codeforces',
u_version': 166, 'u_primary_term': 3, 'u_result': 'u:updated', 'u_id': 'u:1'})
('result is ', {'u_type': 'u:user_data', 'u_seq_no': 166, 'u_shards': {'u_successful': 1, 'u_failed': 0, 'u_total': 2}, 'u_index': 'u:codeforces',
u_version': 167, 'u_primary_term': 3, 'u_result': 'u:updated', 'u_id': 'u:1'})
```

## CONCLUSION

After doing this project we found that Kafka is powerful tool for handling loads of data efficiently

[Here](#) is the link to the Github repository.