A REPORT ON

KAFKA Deployment And Creating Publisher/Consumer
For ReST Based Servies

International Institute Of Information
Technology, Bangalore
Computer Science and Engineering

Guidance by

Prof. Samar Shailendra

Submitted by

Shubham Gupta (IMT2016118)

Suparn S Lele (MT2018122)

INDEX

1.0 Kafka Introduction:

Apache Kafka is an open-source stream-processing software platform developed by LinkedIn and donated to the Apache Software Foundation, written in Scala and Java. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds.

Kafka uses a binary TCP design that is optimized for efficiency and relies on a "message set" abstraction that naturally groups messages together to reduce the overhead of the network roundtrip. This "leads to larger network packets, larger sequential disk operations, contiguous memory blocks [...] which allows Kafka to turn a bursty stream of random message writes into linear writes."

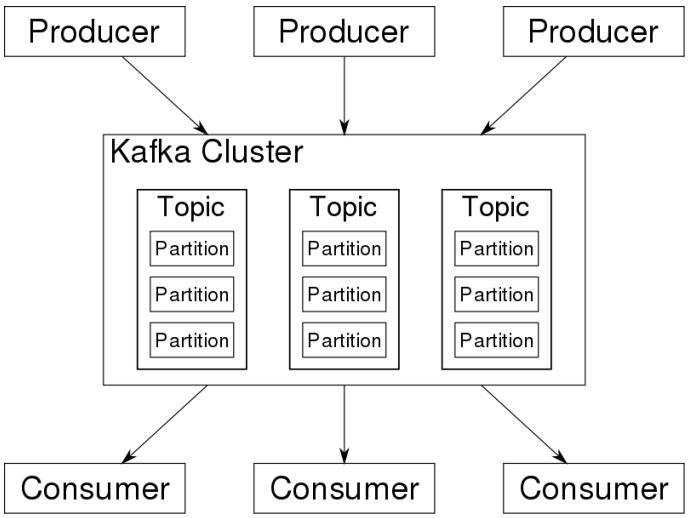


Fig 1. Kafka Architecture

2.0 Components :

2.1 Zookeeper

Apache Zookeeper is a distributed, open-source configuration,synchronization service along with naming registry for distributed applications. ZooKeeper stores a lot of shared information about **Kafka Consumers** and Kafka **Brokers.**

2.2 Broker

A Kafka cluster is made up of multiple Kafka Brokers. Each Kafka Broker has a unique ID (number). Kafka Brokers contain topic log partitions. Connecting to one broker bootstraps a client to the entire Kafka cluster. For failover, you want to start with at least three to five brokers. A Kafka cluster can have, 10, 100, or 1,000 brokers in a cluster if needed.

2.3 Producer

Producers push data to brokers. When the new broker is started, all the producers search it and automatically sends a message to that new broker.

2.4 Consumer

Consumers read data from brokers. Consumers subscribes to one or more topics and consume published messages by pulling data from the brokers.

2.5 Topic

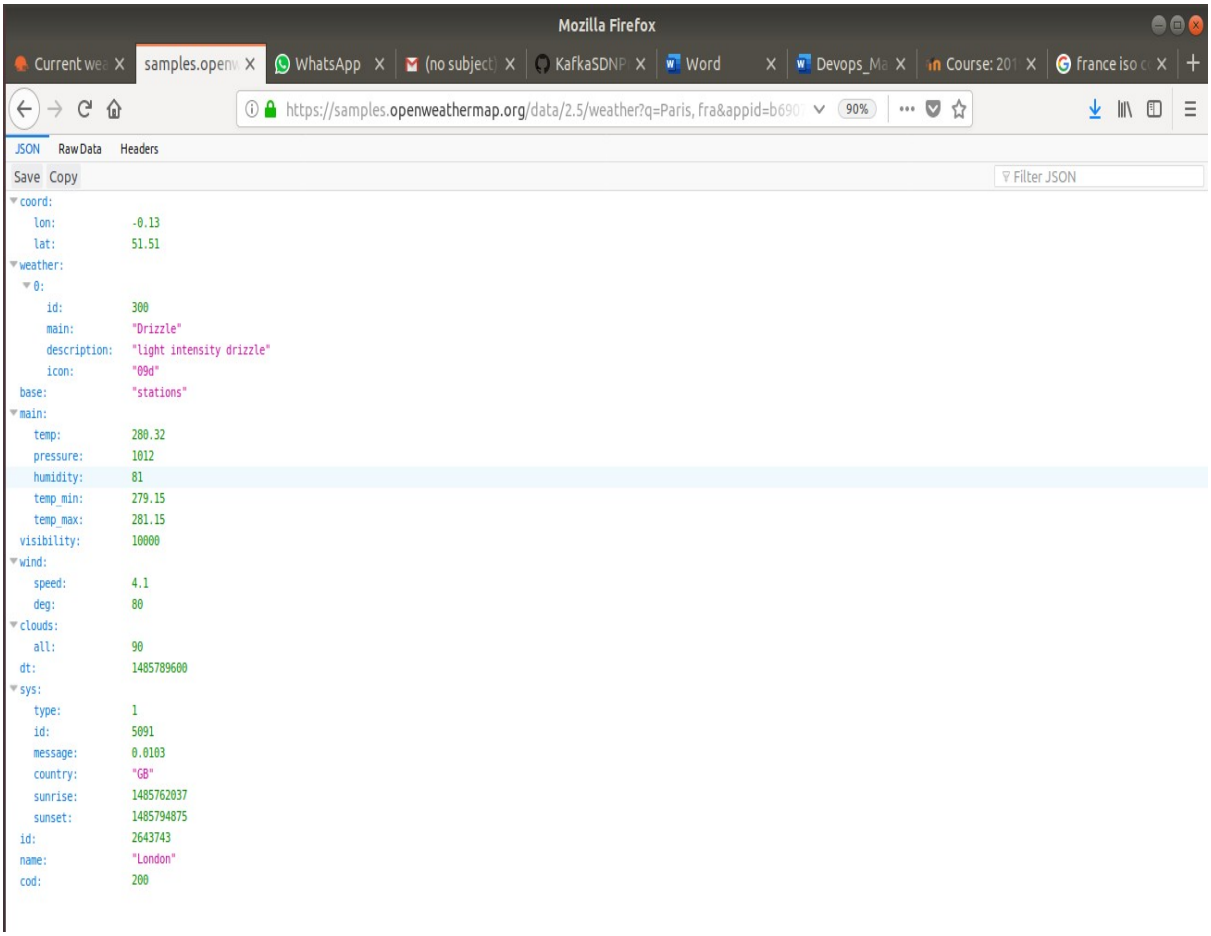
Kafka topic is a named stream of records. Kafka stores topics in logs. A topic log is broken up into partitions. Kafka spreads log’s partitions across multiple servers or disks.

3.0 Data :

We are collecting weather data from https://openweathermap.org/current#name. Sample output from this website when we put London as a city is provided below.

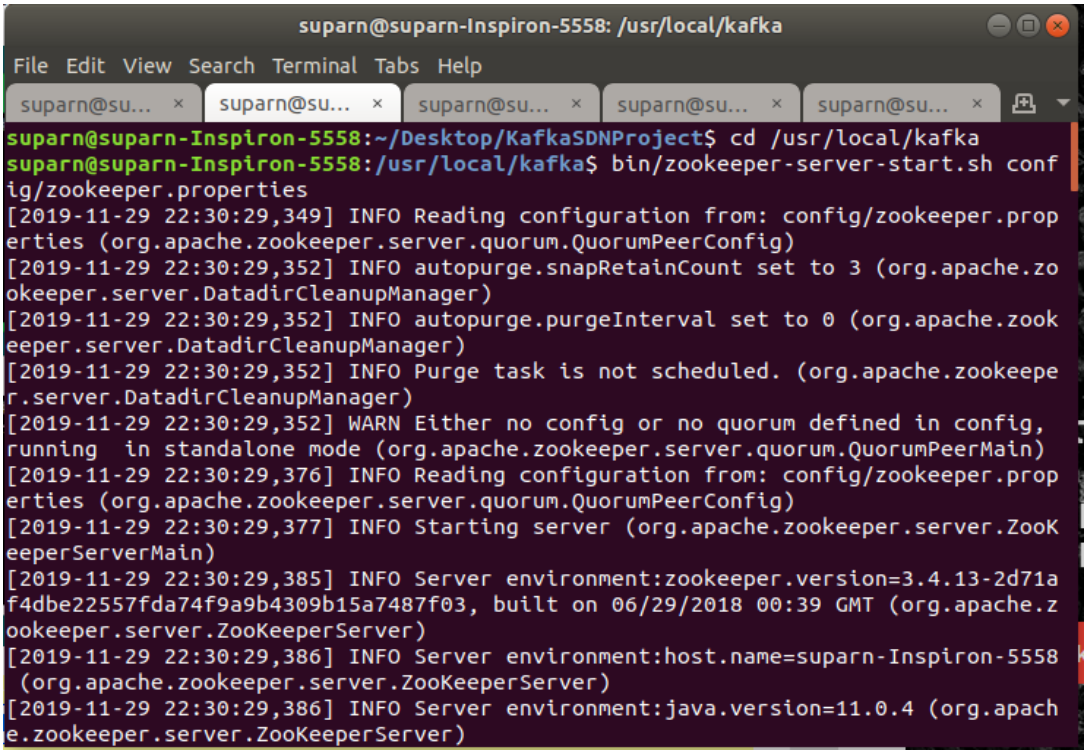
Link :-

https://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid=b6907d289e10d714a6e88b30761fae22

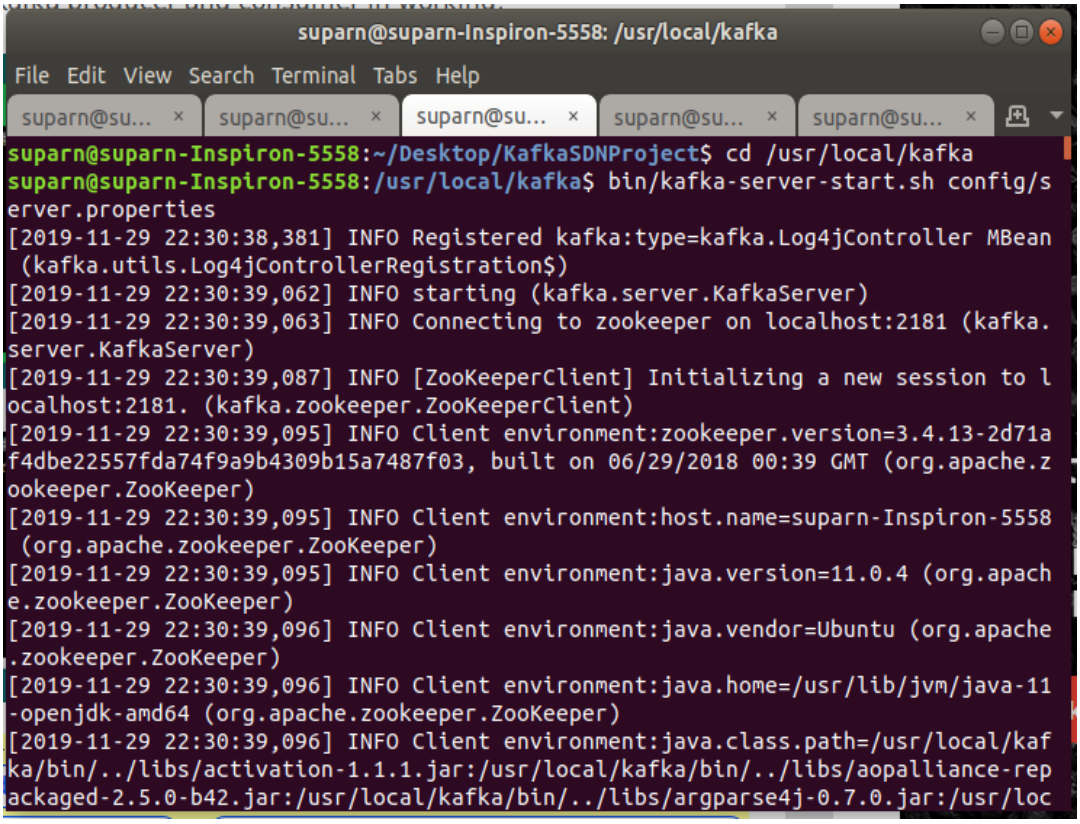


4.0 Screenshots of Steps :

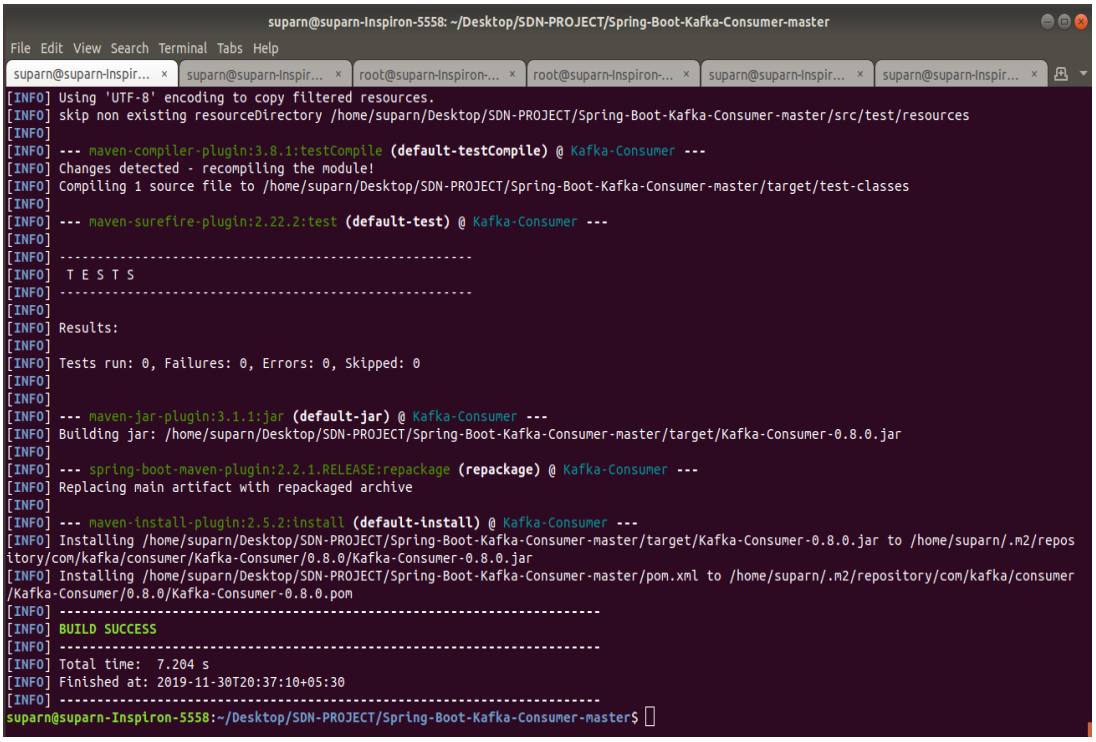
Zoo-Keeper Server



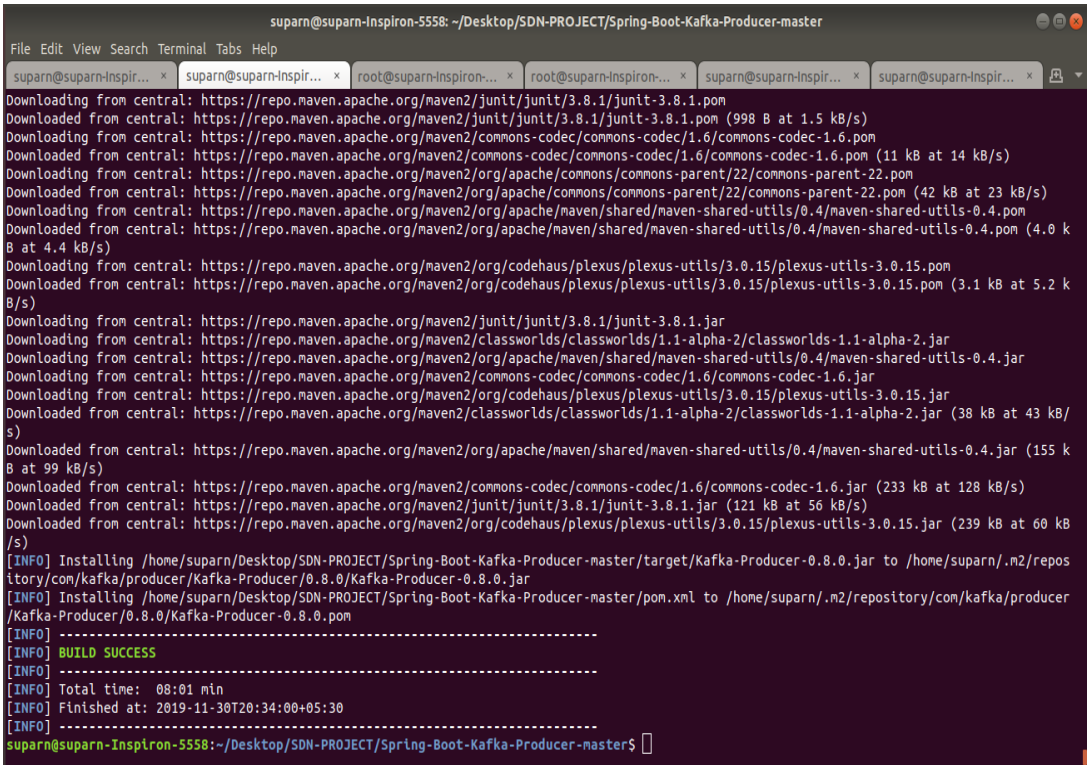
Kafka Server



Consumer Build

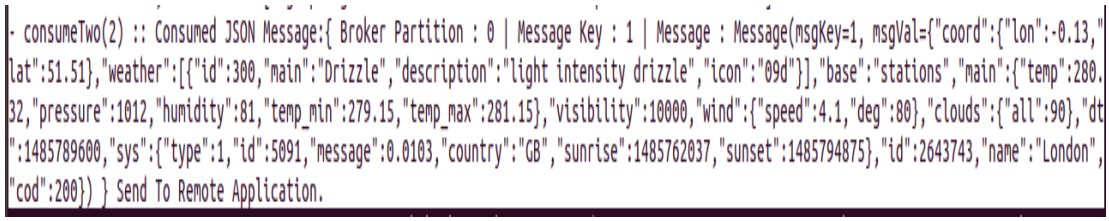


Producer Build



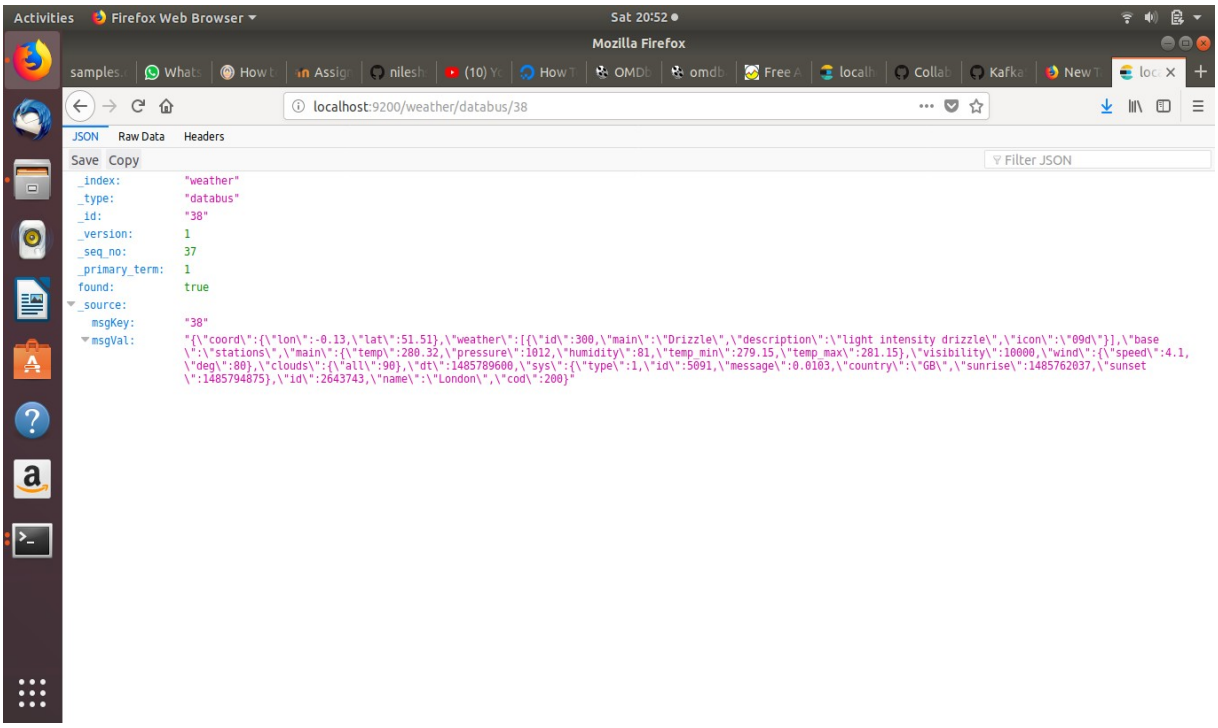
Consumer 1

Consumer 2

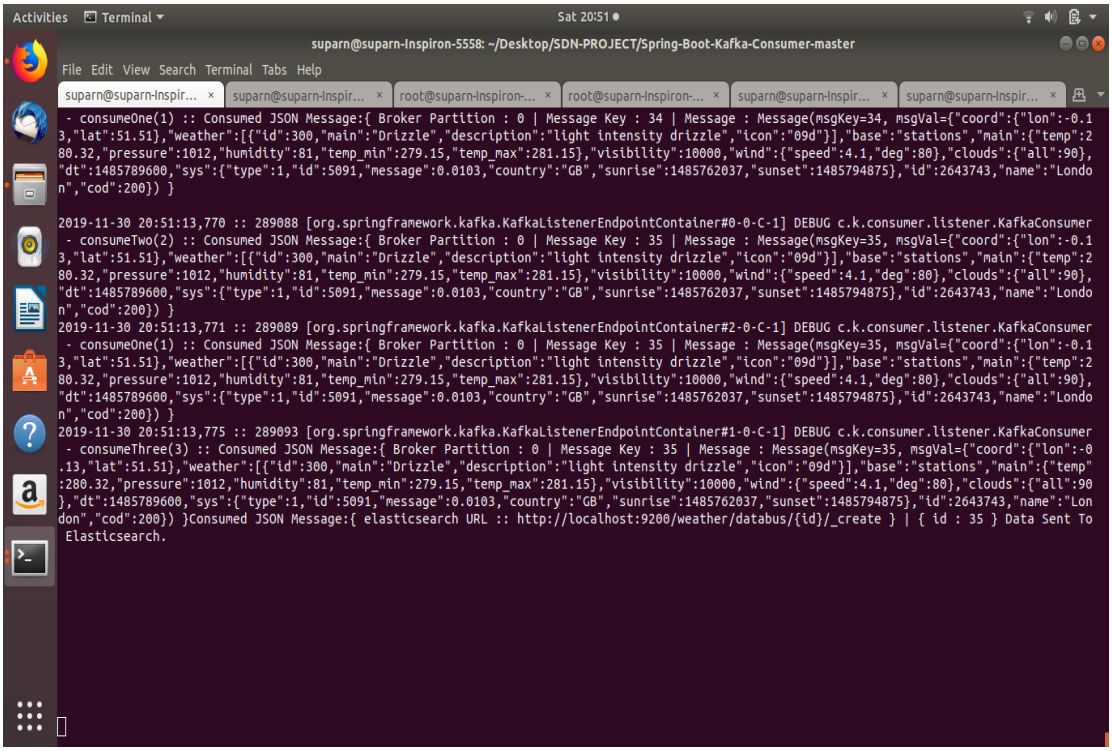


Consumer 3

Elastic Search Sent Data at id 38



Combined Screenshot of all consumers



5.0 Conclusion

From this project we learn about the kafka and its multitasking ability with High efficiency. Joining kafka with SDN reduces the load from the control plane by distributing the load to the multiple consumers.

6.0 Git Hub Link Of the Repository

https://github.com/suparnlele/SDN-Kafka-Project

7.0 References:

1. https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-on-an-ubuntu-vps – Elastic search installation
2. https://tecadmin.net/install-apache-kafka-ubuntu/ - Kafka Installation
3. https://samples.openweathermap.org – Weather Data