Purpose:

**To push the data from a set of files , using kafka .**

Here , we will use **Kafka** , a high throughput , distributed , publish-subsribe messaging system.

We will build 2 kafka topics .

1st to push the data from a set of 10 csv files into 1 kafka topic .

2nd when Spark streaming code takes data from Kafka topic in a window of 60 seconds, process it so that we have the total count of each unique order status in that 60 seconds window. After processing the total count of each unique order status gets pushed to new Kafka topic

Following is the data flow of the data pipeline :

e-commerce Portal 🡪

**(Stage 1)**Push (datatime , ordered ,order\_status ) to a Kafka topic 🡪

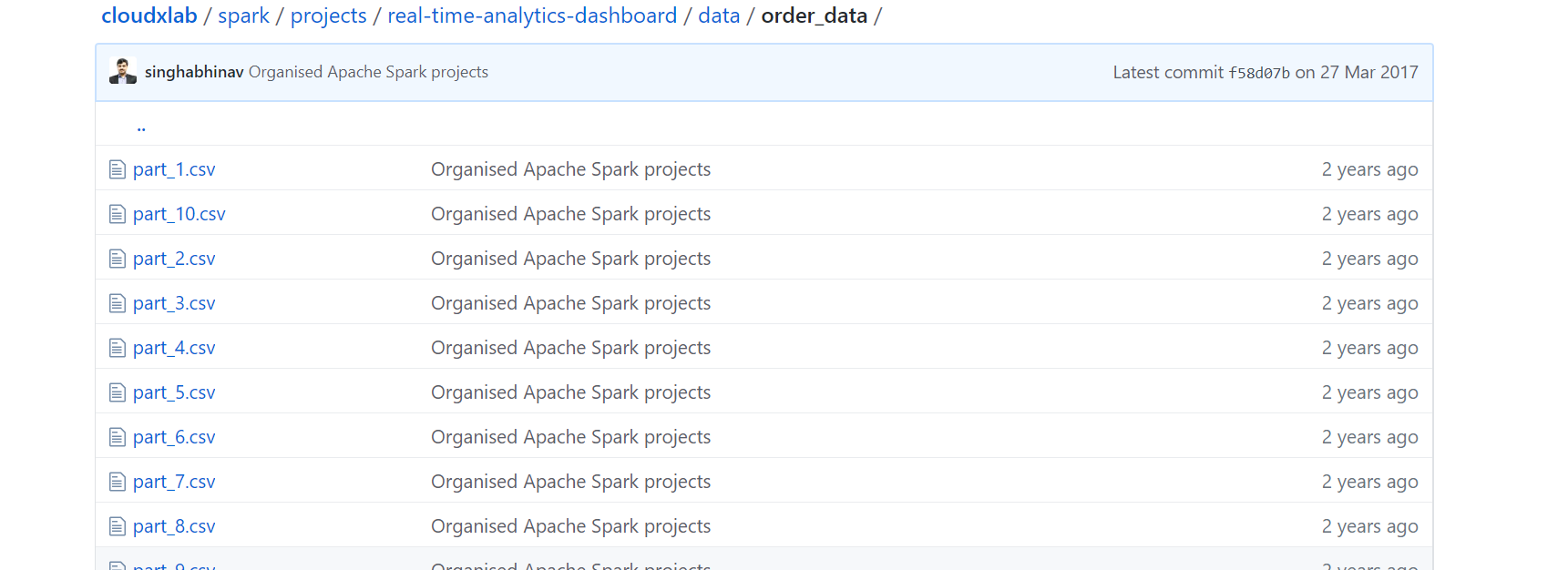
**(Stage 2)**Check the Kafka consumer for the data pushed by the producer 🡪

Now coming down to explicit discussion for each stage mentioned above :

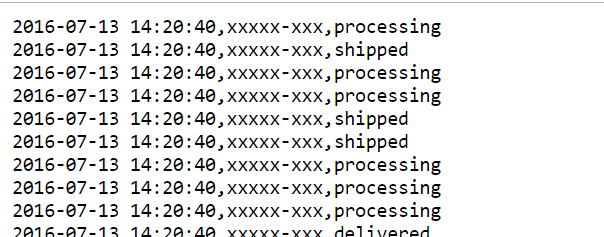
**Stage 1**:

When a customer buys an item or an order status changes in the order management system, the corresponding order id along with the order status and time get pushed to the Kafka topic.

Data files are present along the following path:



Each of the data files ,looks like the below :

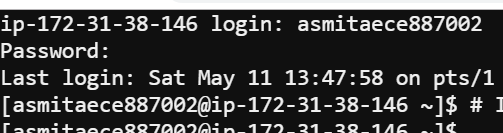


Our dataset contains three columns ‘DateTime’, ‘OrderId’ & ‘Status’. Each row in the dataset represents the order status in particular date time. We are only interested in the number of orders getting shipped every minute, so we don’t need actual ordered , hence those have been masked .

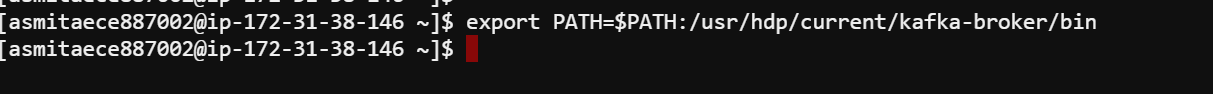
**Push Dataset to Kafka**

Create a topic in kafka , push messages to Kafka .

First login to webconsole



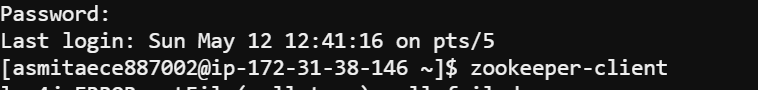
Next , export the path for adding Kafka binary files



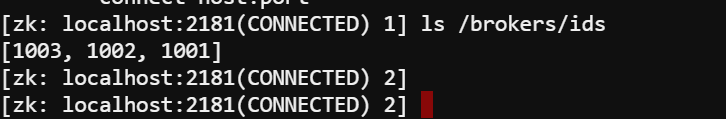
Now we need to locate kafka brokers , before calling kafka producer

The kafka brokers inform zookeeper about their IPs addresses. Most of the eco-system considers the zookeeper as a central registry.

First launch zookeeper client in another Command line interface prompt ,as shown below :

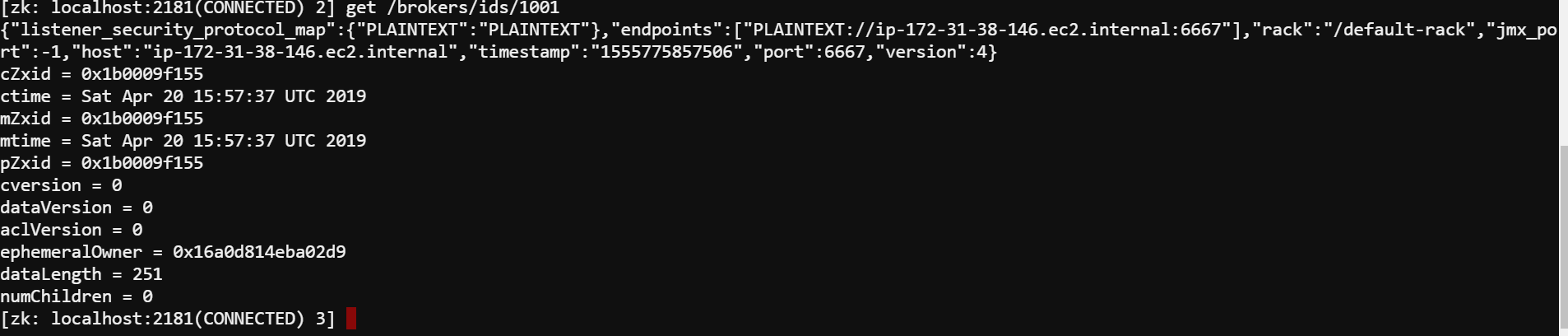


On the zookeeper-client prompt, list all the brokers that registered.



Next , Find the ip address of any broker ( say the 1st broker)from zookeeper-client using command:

**get /brokers/ids/1001**



So the localhost for zookeeper is ip-172-31-38-146.ec2.internal

Now , 1st ,Create Kafka topic with your username.

Now my cloudxlab username is asmitaece887002

Next ,Create topic with name asmitaece887002-order-data and push messages to Kafka.

**kafka-topics.sh --create --zookeeper ip-172-31-38-146.ec2.internal:2181 --replication-factor 1 --partitions 1 --topic asmitaece887002-order-data**



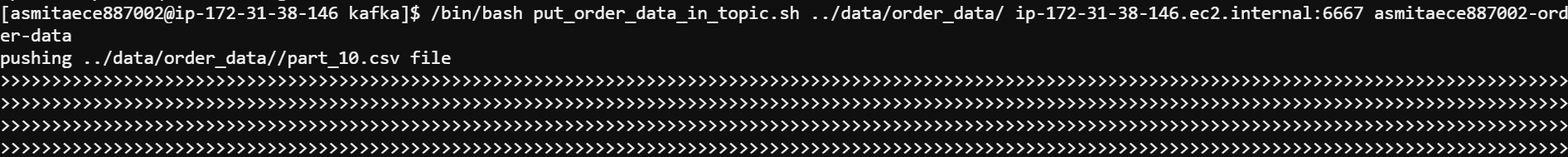
Topic created .

Next go to **spark/projects/real-time-analytics-dashboard/kafka** directory ,and view the shell script **put\_order\_data\_in\_topic.sh**



Next , run the script to push data to Kafka topic.Script pushes CSV files one by one to Kafka topic in one-minute interval

/bin/bash put\_order\_data\_in\_topic.sh ../data/order\_data/ ip-172-31-38-146.ec2.internal:6667 asmitaece887002-order-data



The >>> are the messages pushed from data files one by one

For example , the data from the part\_10.csv got started to being pushed , subsequently data from all the .csv files are pushed .

As denoted , data from part\_5.csv also gets pushed





Finally after sometime , the data from all the .csv files are pushed to kafka topic

Stage 2:

**To view some of the messages pushed to kafka by producer , we need to view the Kafka consumer in new CLI terminal**

**kafka-console-consumer.sh --zookeeper ip-172-31-38-146.ec2.internal:2181 --topic asmitaece887002-order-data --from-beginning**

For this , we will view as below :

