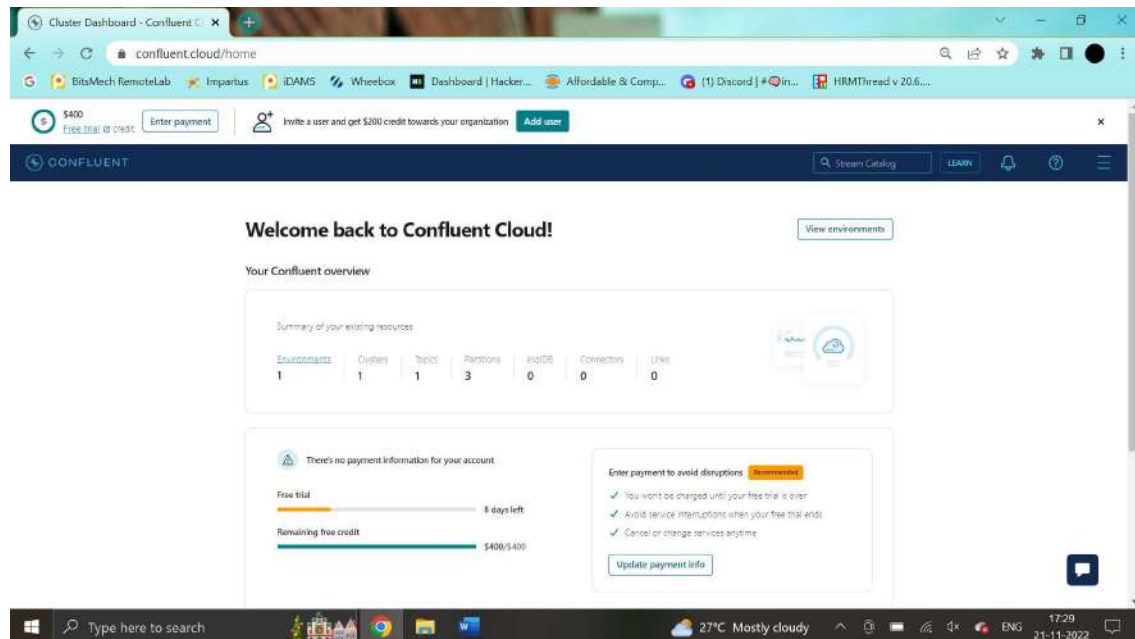# KAFKA - ASSIGNMENT

download restaurant data from below mentioned link

Download Data Link -> https://github.com/shashank-mishra219/Confluent-Kafka-Setup/blob/main/restaurant_orders.csv

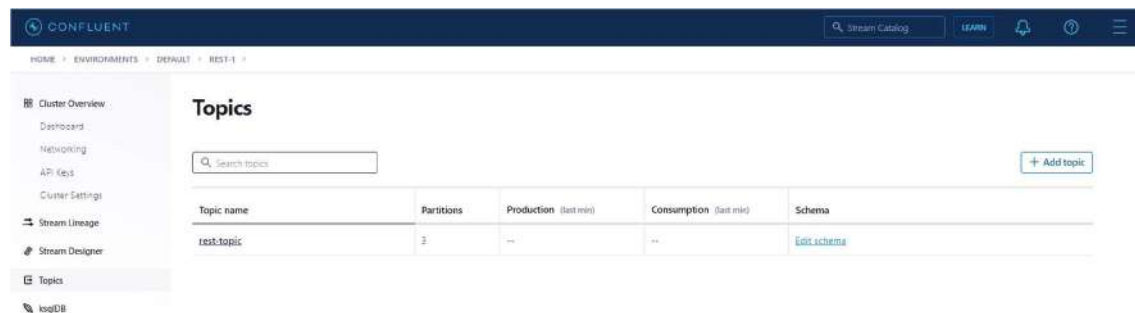**Complete the given below task to finish this assignment.**

1. **Setup Confluent Kafka Account**
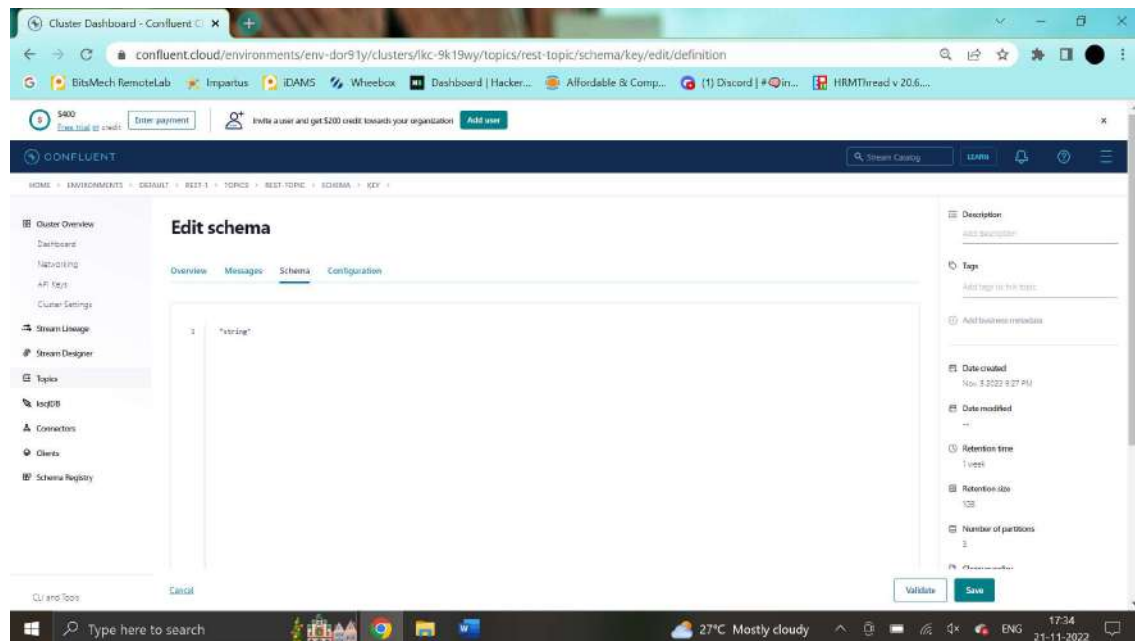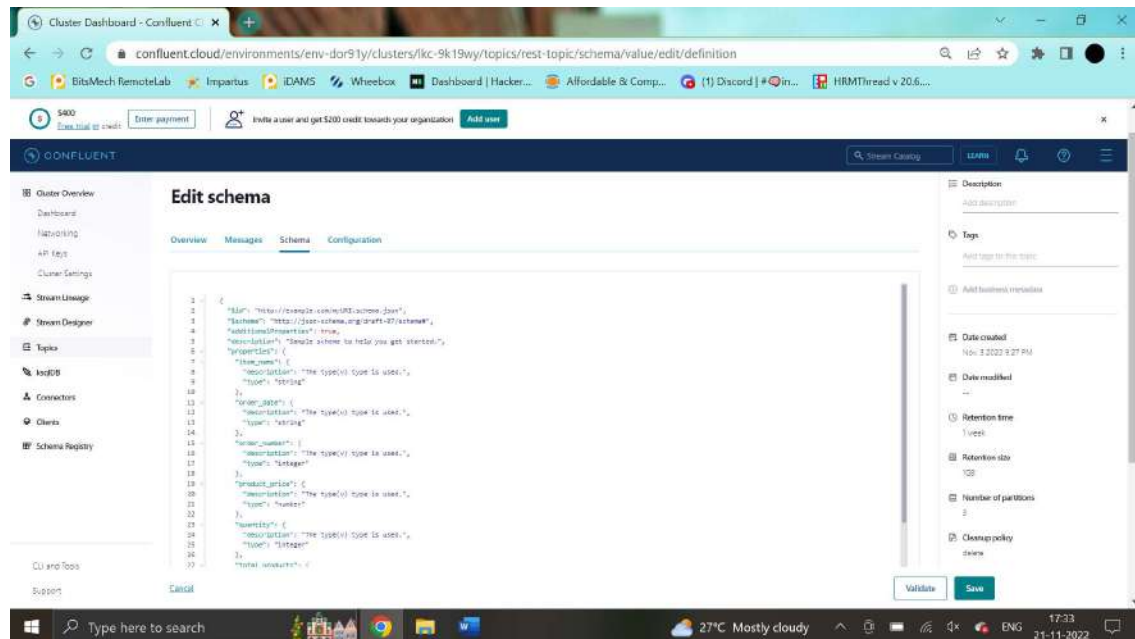   Confluent cloud account home page

2. **Create one kafka topic named as "restaurent-take-away-data" with 3 partitions**

   Created topic named with "rest-topic" with 3 partitions

3. Setup key (string) & value (json) schema in the confluent schema registry

Schema setup done using key as string and values as json with in confluent schema registry

4. **Write a kafka producer program (python or any other language) to read data records from restaurent data csv file,**

  make sure schema is not hardcoded in the producer code, read the latest version of schema and schema_str from schema registry and use it for data serialization.

Reading CSV data

```
80
81
82    def get_order_instance(file_path):
83        df=pd.read_csv(file_path)
84        orders:List[Order]=[]
85        for data in df.values:
86            order=Order(dict(zip(columns,data)))
87            orders.append(order)
88            yield order
```

Reading schema from Schema registry

```
119
120   def main(topic):
121
122
123       schema_registry_conf = schema_config()
124       schema_registry_client = SchemaRegistryClient(schema_registry_conf)
125       schema_str = schema_registry_client.get_latest_version('rest-topic-value').schema.schema_str
126       string_serializer = StringSerializer('utf_8')
127       json_serializer = JSONSerializer(schema_str, schema_registry_client, order_to_dict)
128
129       producer = Producer(sasl_conf())
130
131       print("Producing user records to topic {}. ^C to exit.".format(topic))
132       #while True:
133           # Serve on_delivery callbacks from previous calls to produce()
134       producer.poll(0.0)
135       try:
136           for order in get_order_instance(file_path=FILE_PATH):
```

Click link for Complete producer code - Link

5. From producer code, publish data in Kafka Topic one by one and use dynamic key while publishing the records into the Kafka Topic

Producing messages to topic

6. Write kafka consumer code and create two copies of same consumer code and save it with different names (kafka_consumer_1.py & kafka_consumer_2.py), again make sure lates schema version and schema_str is not hardcoded in the consumer code, read it automatically from the schema registry to desrialize the data.

  Now test two scenarios with your consumer code:

  a.) Use "group.id" property in consumer config for both consumers and mention different group_ids in kafka_consumer_1.py & kafka_consumer_2.py, apply "earliest" offset property in both consumers and run these two consumers from two different terminals. Calculate how many records each consumer consumed and printed on the terminal
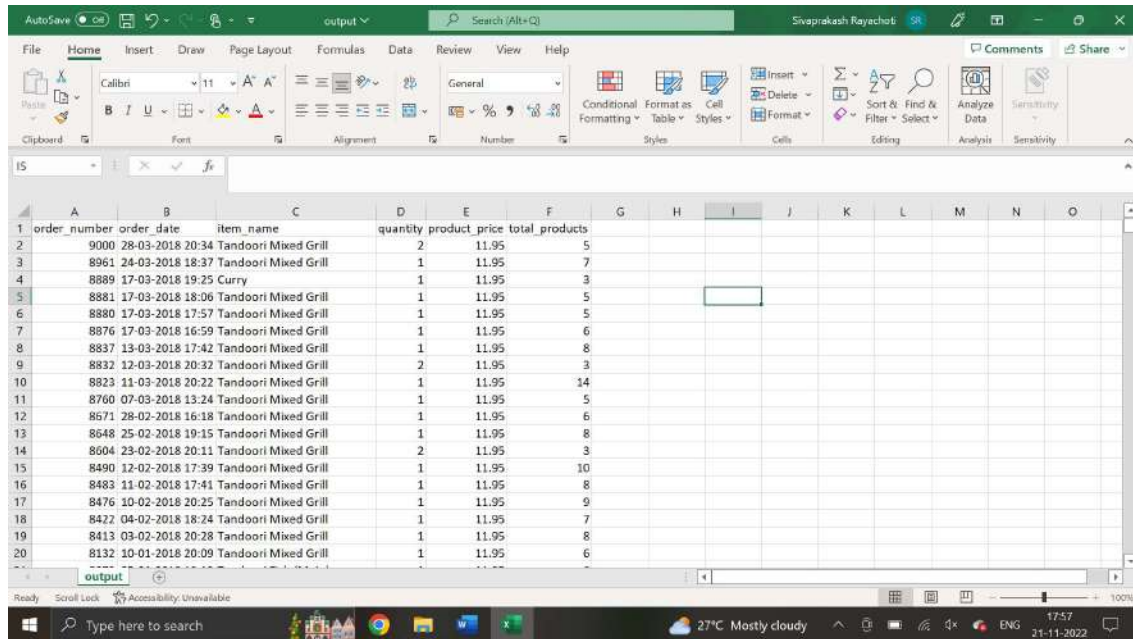
   Created 2 consumer codes with different group ids as "groupid1" and "groupid2".

Messages consumed by consumer – 1 is 74818 Nos

Messages consumed by consumer – 2 is 74818 Nos

Click on link for consumer 1 code - Link

Click on link for consumer 2 code - Link

b.) Use "group.id" property in consumer config for both consumers and mention same group_ids in kafka_consumer_1.py & kafka_consumer_2.py, apply "earliest" offset property in both consumers and run these two consumers from two different terminals. Calculate how many records each consumer consumed and printed on the terminal

Now, used same group id for both consumers and consumed messages

Messages consumed by consumer – 1 is 24935 Nos

Messages consumed by consumer – 2 is 49883 Nos

It is observed that messages being shared by both consumers

7. Once above questions are done, write another kafka consumer to read data from kafka topic and from the consumer code create one csv file "output.csv" and append consumed records output.csv file.

Consumer code updated with required code and after deploying following output obtained.

Click on link for consumer code - Link