**Kafka Topic, Kafka Producer, Kafka Consumer**

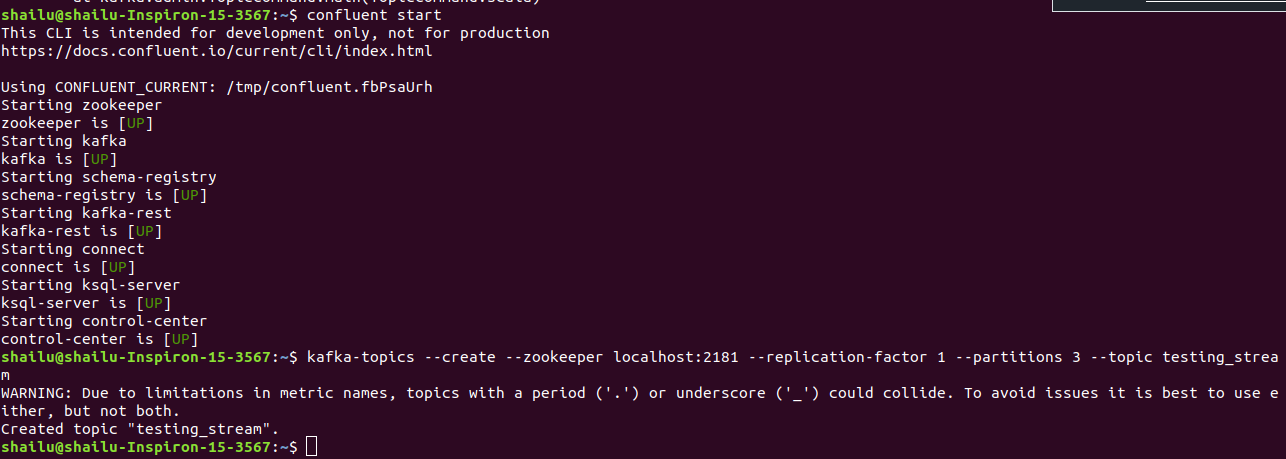
I have installed confluent platform 5.1.2 on my local system for this assignment work. I am using json format for this assignment. We can also load data in Avro format and can utilize schema registry feature to maintain schema versions for the data loaded to kafka topic.

**1. Create Kafka topic named testing\_stream with 3 partitions:**

Solution: As per latest versions of confluent we can create topics using Control Centre UI as well.

Here I am giving CLI command to create the topic:

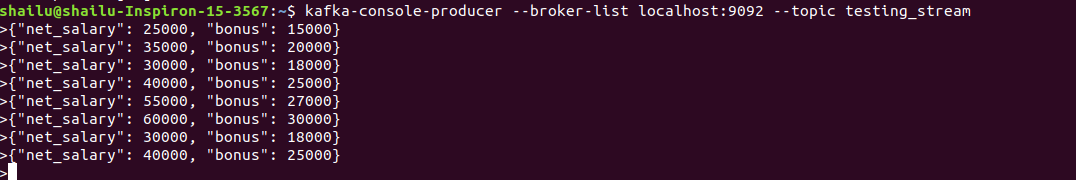
**$** *kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic testing\_stream*



**4. Create producer of the topic (pass 2 int to the consumer):**

CLI command to produce data to the topic:

**$** *kafka-console-producer --topic testing\_stream --broker-list localhost:9092*



**2. Create consumer group named stream\_Group with 3 consumers:**

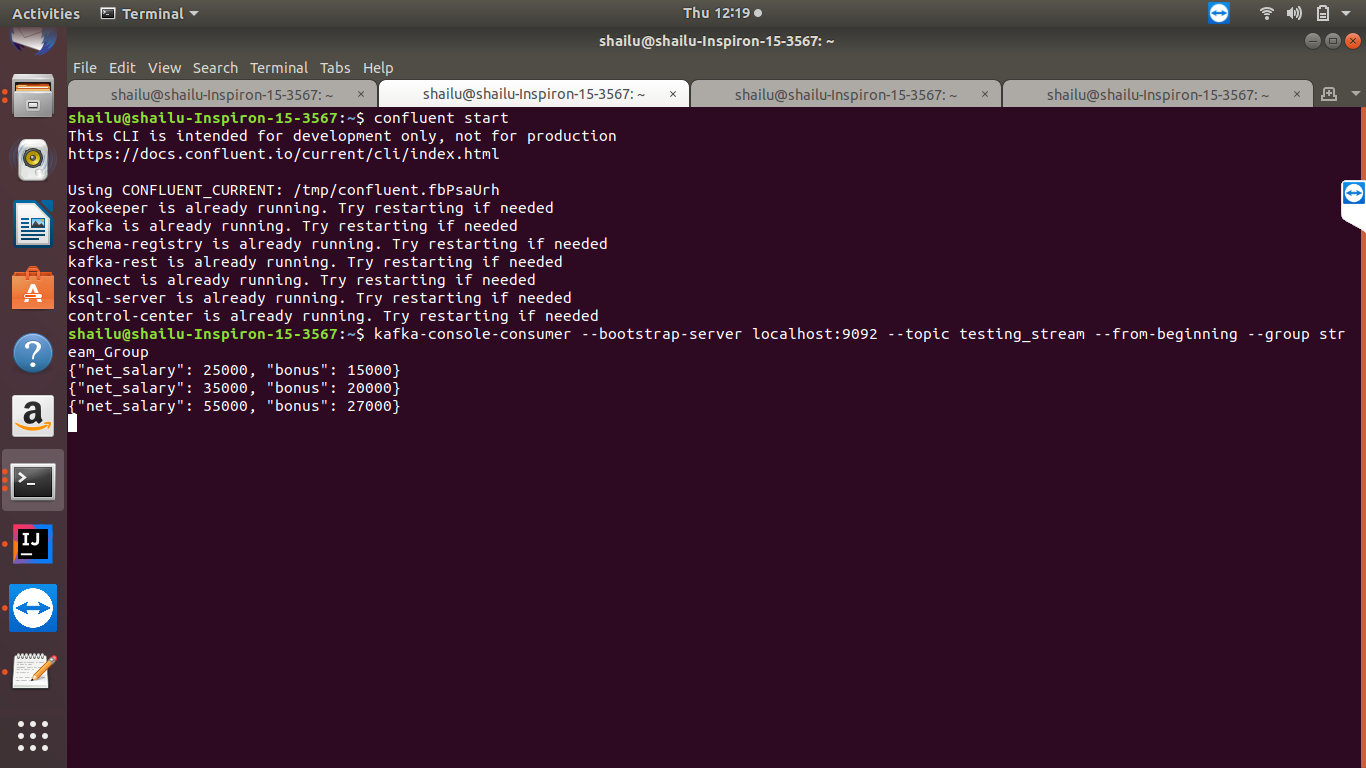
Solution:

CLI command to consume topic data with consumer group as stream\_Group:

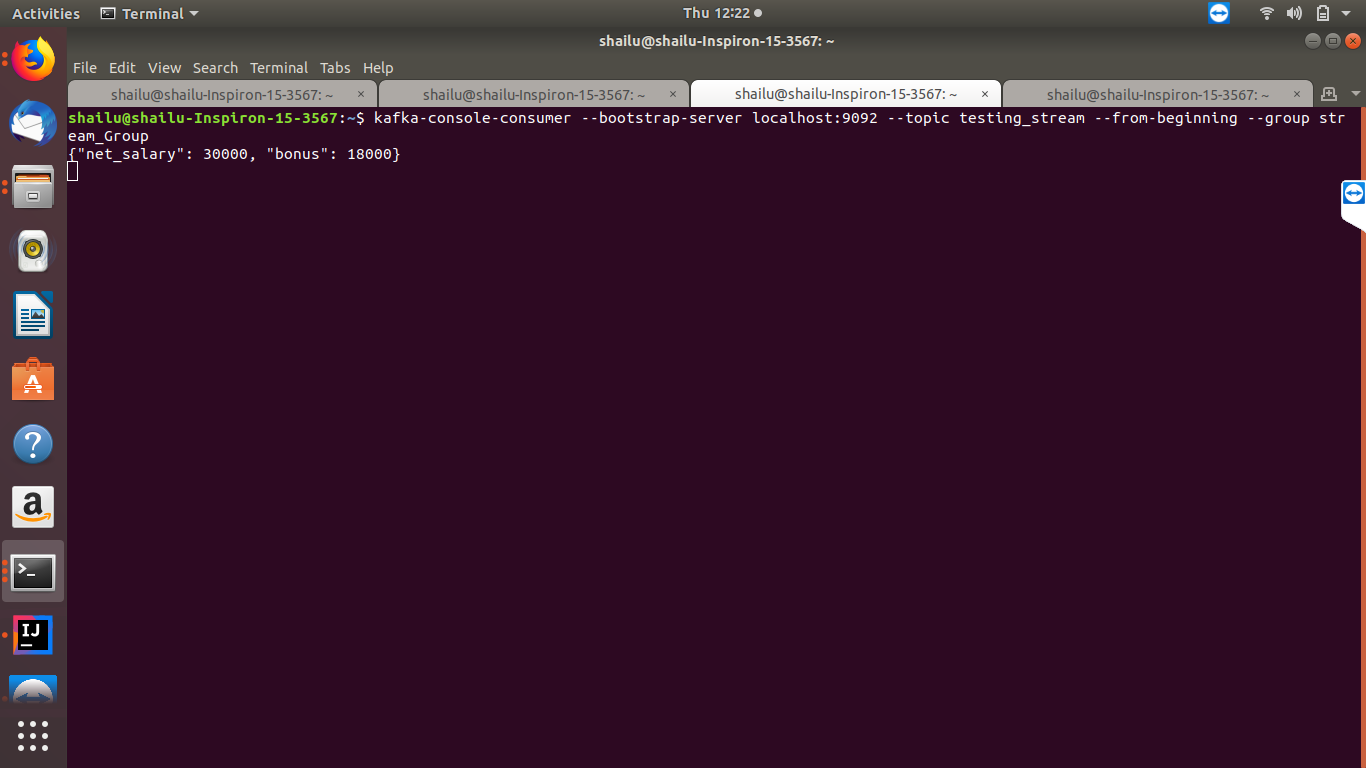
**$** *kafka-console-consumer --bootstrap-server localhost:9092 --topic testing\_stream --from-beginning --group stream\_Group*

Run the above command in three different terminals, there will be three consumers created for the same consumer group.

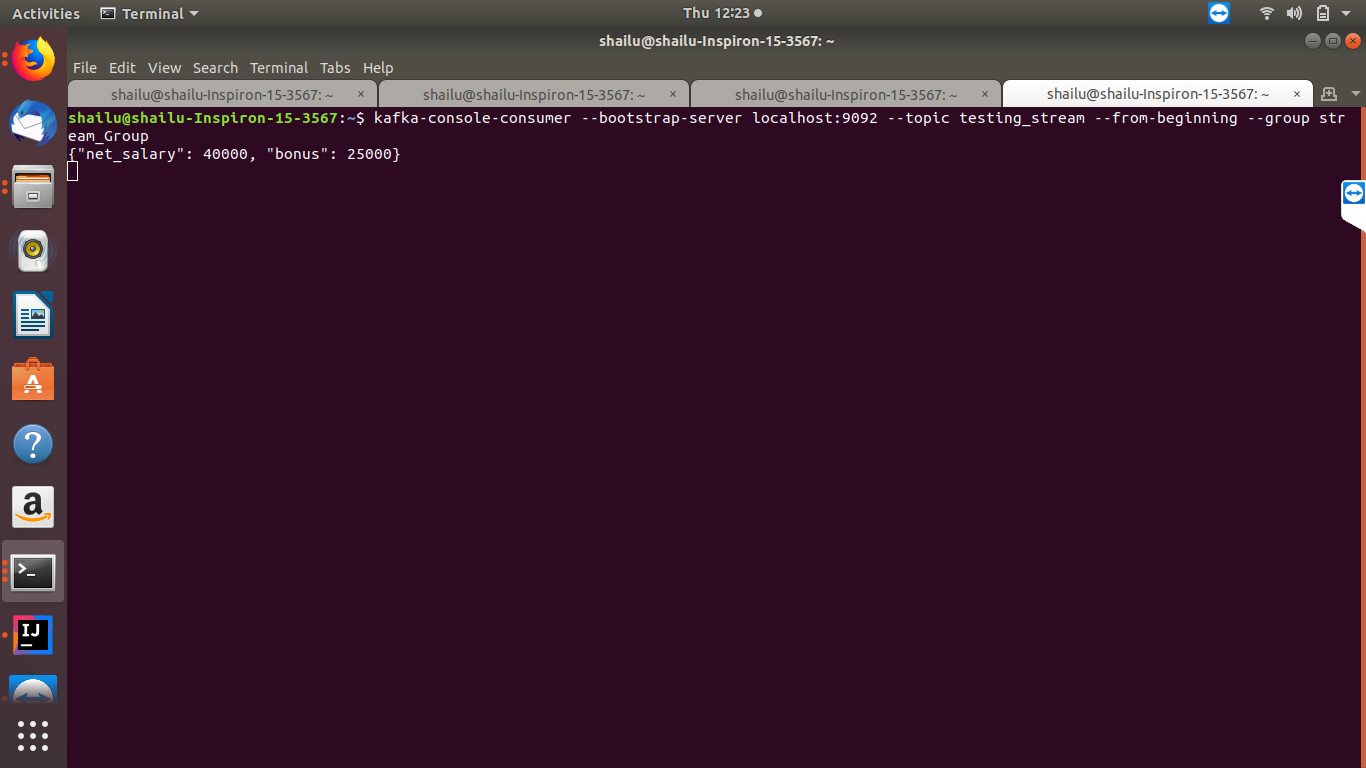
Consumer1:



Consumer2:



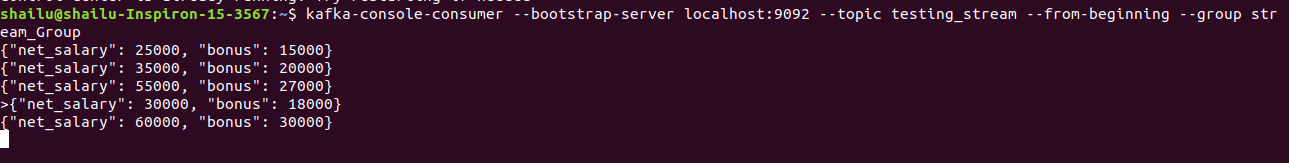
Consumer3:



**3. Subscribe the topic testing\_stream:**

CLI command to subscribe to the topic:

**$***kafka-console-consumer --bootstrap-server localhost:9092 --topic testing\_stream --from-beginning --group stream\_Group*



**5. Read stream from the topic and perform transformation:**

a. Pass this stream through the function which adds the 2 integers and return sum

b. Show the sum on console as soon as producer produces the message

Solution: I have used structured streaming to read the stream of data from topic and created an udf to sum the two columns. For validation purpose I have kept both the input columns along with sum in the output.

*import org.apache.spark.sql.functions.from\_json*

*import org.apache.spark.sql.types.{IntegerType, StructType}*

*import org.apache.spark.sql.functions.udf*

*val schema = (new StructType()).add("net\_salary",IntegerType).add("bonus",IntegerType)*

*val data = spark.readStream.format("kafka").option("kafka.bootstrap.servers","localhost:9092").option("subscribe","testing\_stream").option("startingOffsets","earliest").option("groupid","stream\_Group").load*

*def sumUdf = udf {(a: Int,b: Int) => a+b}*

*val jsonData = data.selectExpr("cast(value as string) as json").select(from\_json($"json", schema).as("data")).select("data.\*").select(col("net\_salary"),col("bonus"),sumUdf(col("net\_salary"),col("bonus")).as("sum")).writeStream.outputMode("append").format("console").option("checkpointLocation","/user/shailu/firstProject").start().awaitTermination()*

