# **Hands-on Lab: Message Keys and Offset**

Similar to first 4 Exercises.

5. Create a topic and producer for processing bank ATM transactions

Start a new terminal and go to the extracted Kafka folder:

cd kafka\_2.12-2.8.0

Create a new topic using the --topic argument with the name bankbranch. In order to simplify the topic configuration and better explain how message key and consumer offset work, here we specify --partitions 2 argument to create two partitions for this topic. You may try other partitions settings for this topic if you are interested in comparing the difference.

bin/kafka-topics.sh --bootstrap-server localhost:9092 --create --topic bankbranch -- partitions 2

Now let’s list all the topics to see if bankbranch has been created successfully.

bin/kafka-topics.sh --bootstrap-server localhost:9092 –list

We can also use the --describe command to check the details of the topic bankbranch

bin/kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic bankbranch

and you can see bankbranch has two partitions Partition 0 and Partition 1. If no message keys are specified, messages will be published to these two partitions in an alternating sequence, like this:

Partition 0 -> Partition 1 -> Partition 0 -> Partition 1 …

Next, we can create a producer to publish some ATM transaction messages.

Stay in the same terminal window with the topic details, then create a producer for topic bankbranch

bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --topic bankbranch

To produce the messages, look for the the > icon, and copy and paste the following ATM messages after it:

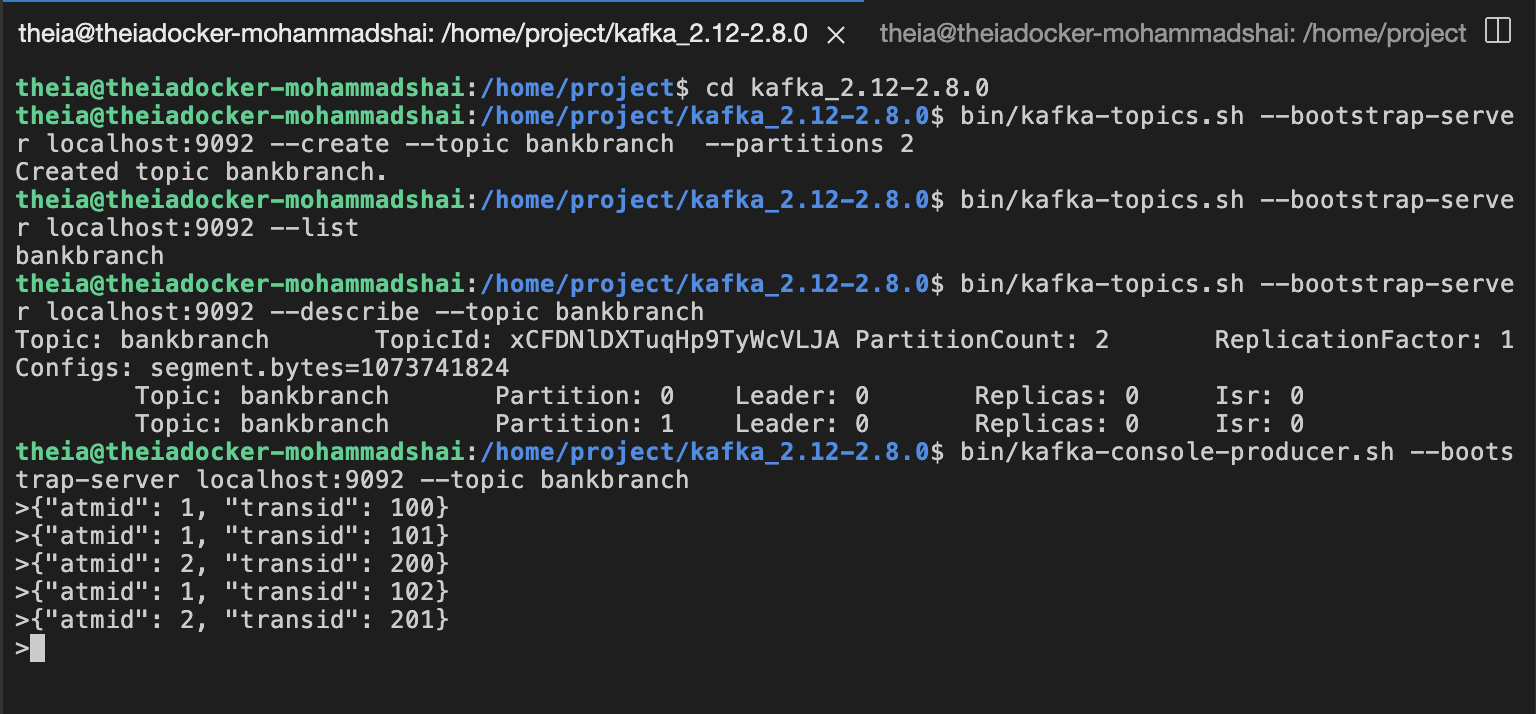
{"atmid": 1, "transid": 100}

{"atmid": 1, "transid": 101}

{"atmid": 2, "transid": 200}

{"atmid": 1, "transid": 102}

{"atmid": 2, "transid": 201}



Then, let’s create a consumer in a new terminal window to consume these 5 new messages.

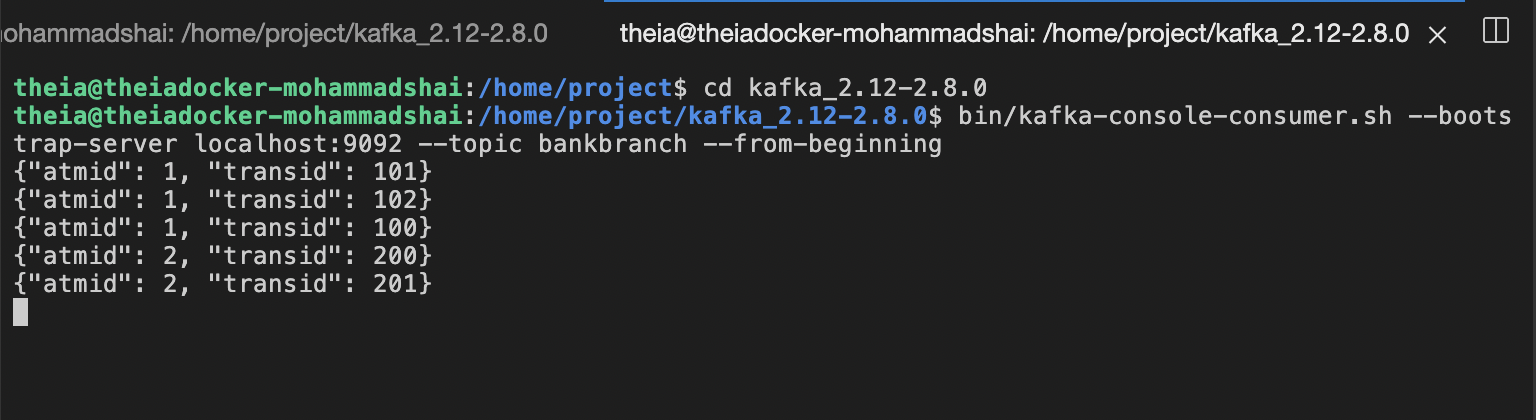
Start a new terminal and go to the extracted Kafka folder:

cd kafka\_2.12-2.8.0

Then start a new consumer to subscribe to the bankbranch topic:

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic bankbranch --from-beginning

Then, you should see the 5 new messages we just published, but very likely, they are not consumed in the same order as they were published. Normally, you need to keep the consumed messages sorted in their original published order, especially for critical use cases such as financial transactions.

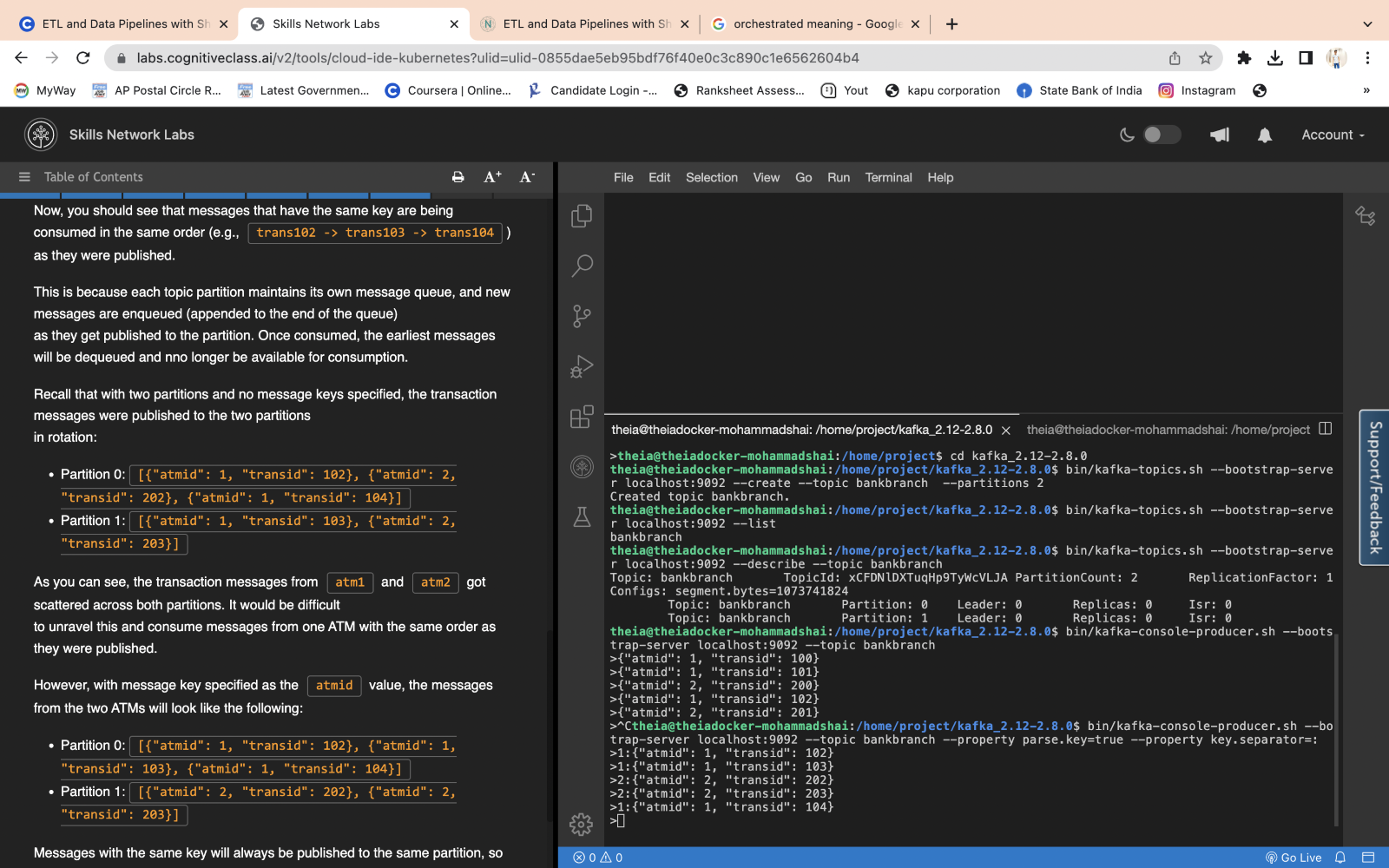


Produce and consume with message keys

Stop both producer and consumer terminals by using cntrl+c

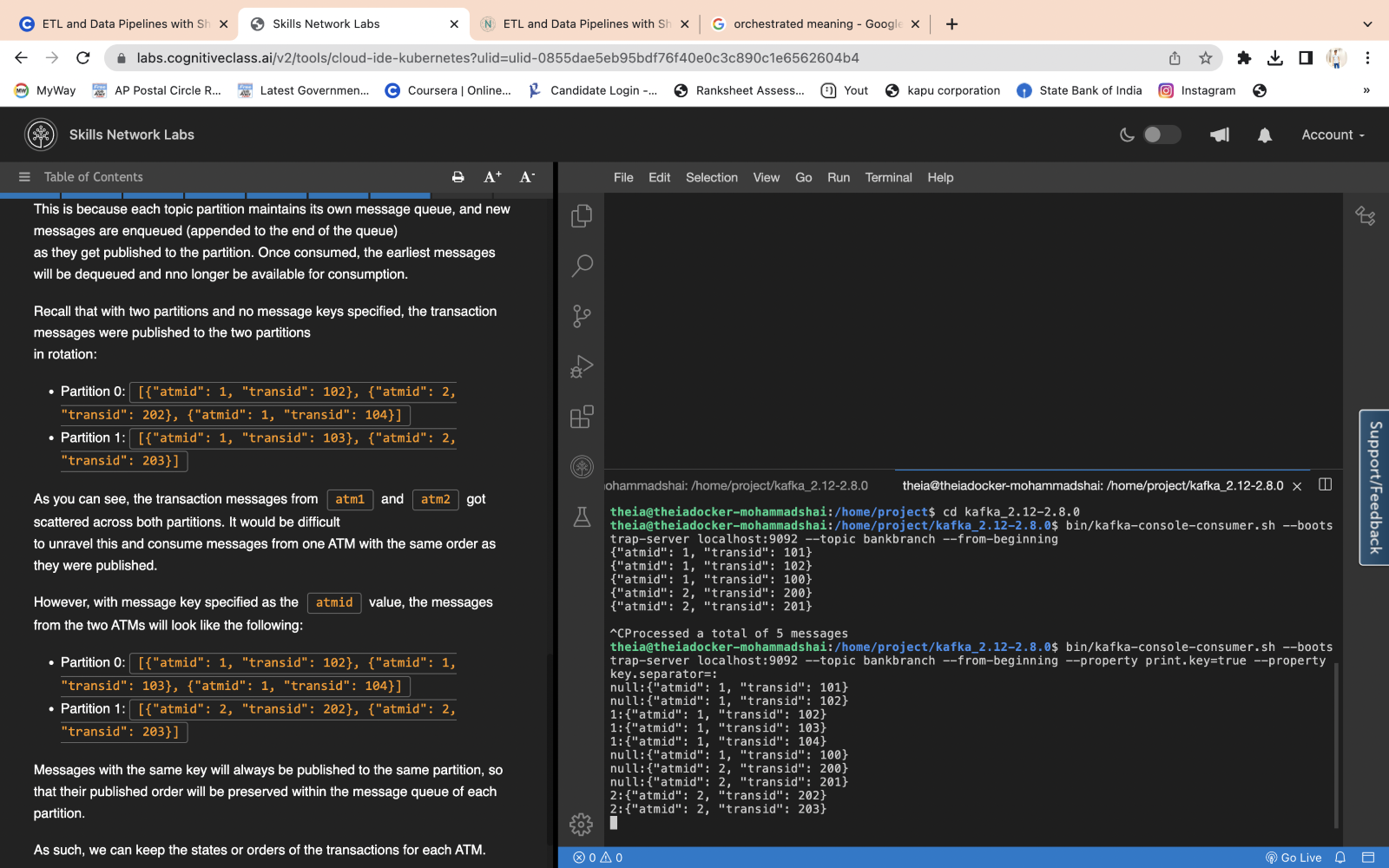
Start a new producer with message key enabled:

bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --topic bankbranch --property parse.key=true --property key.separator=:

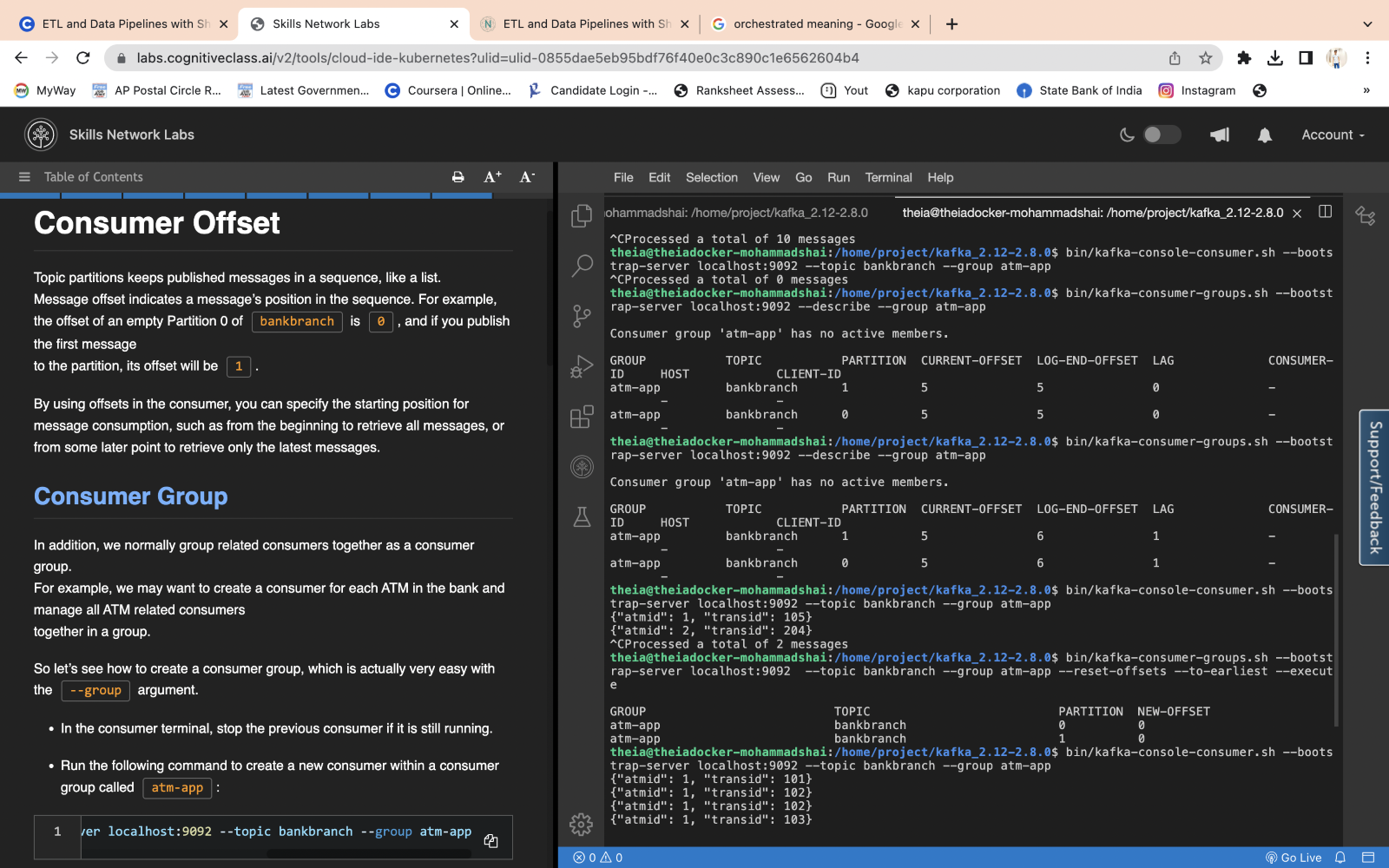


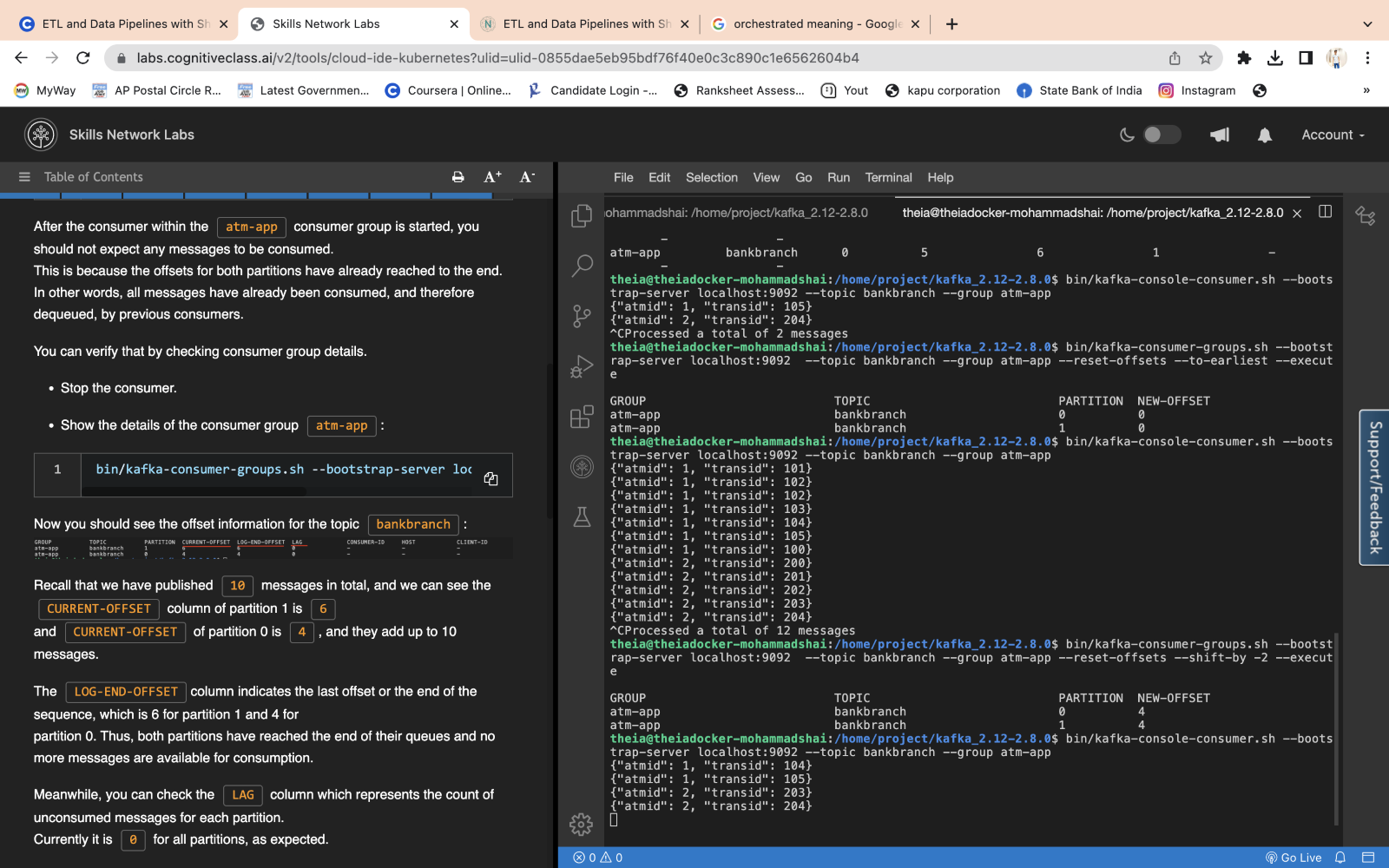
start a new consumer

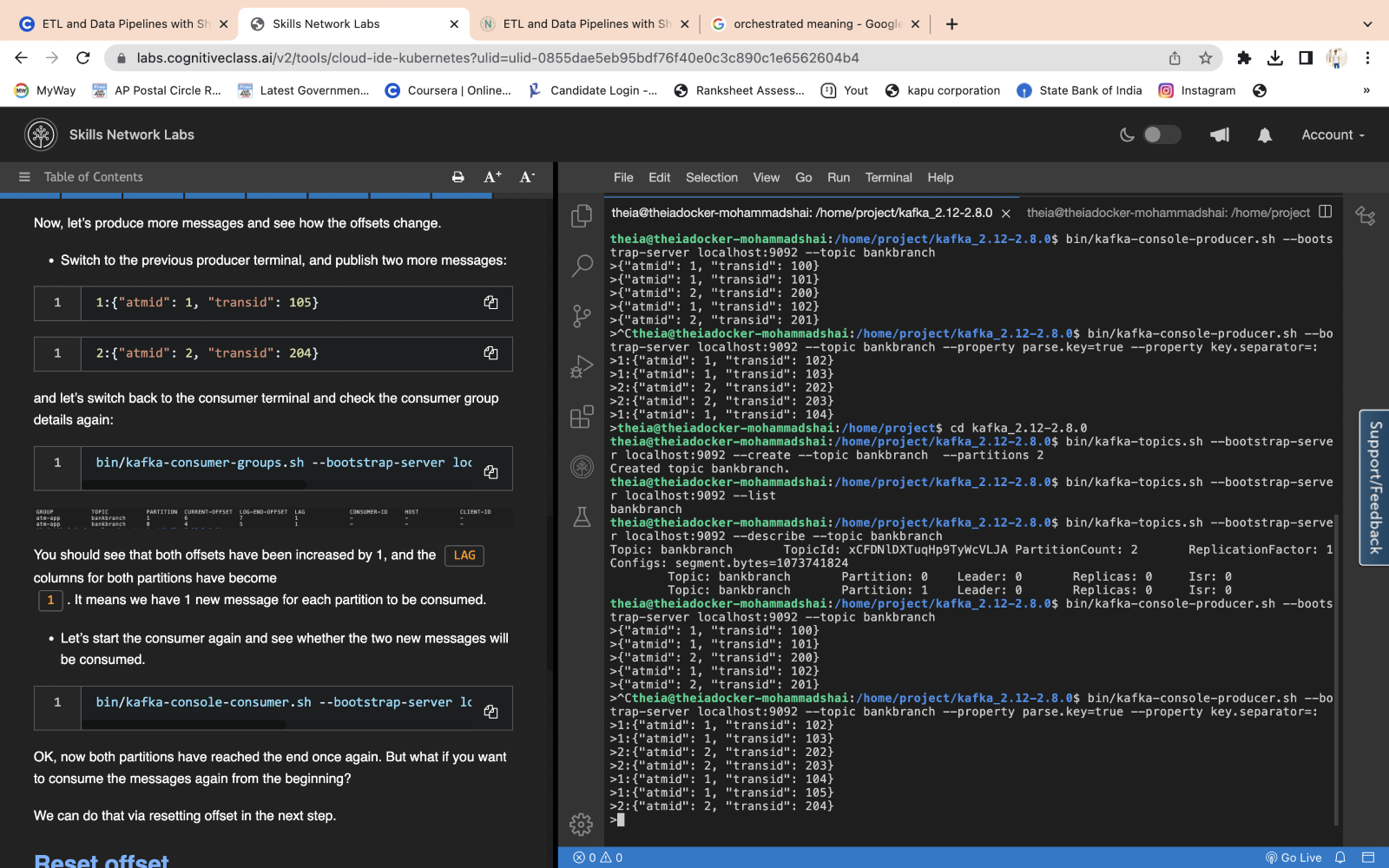
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic bankbranch --from-beginning --property print.key=true --property key.separator=:



Consumer Offset







In this lab, you have learned how to include message keys in publication to keep their message states/order.  
You have also learned how to reset the offset to control the message consumption starting point.