Report: Real-Time E-commerce Order Processing System Using Kafka

Srinibas Masanta (202318054)

May 7, 2024

1 Setting Up Kafka Environment

- Installed and configured Apache Kafka on my local machine.
- Started Zookeeper server using `zookeeper-server-start.bat`.

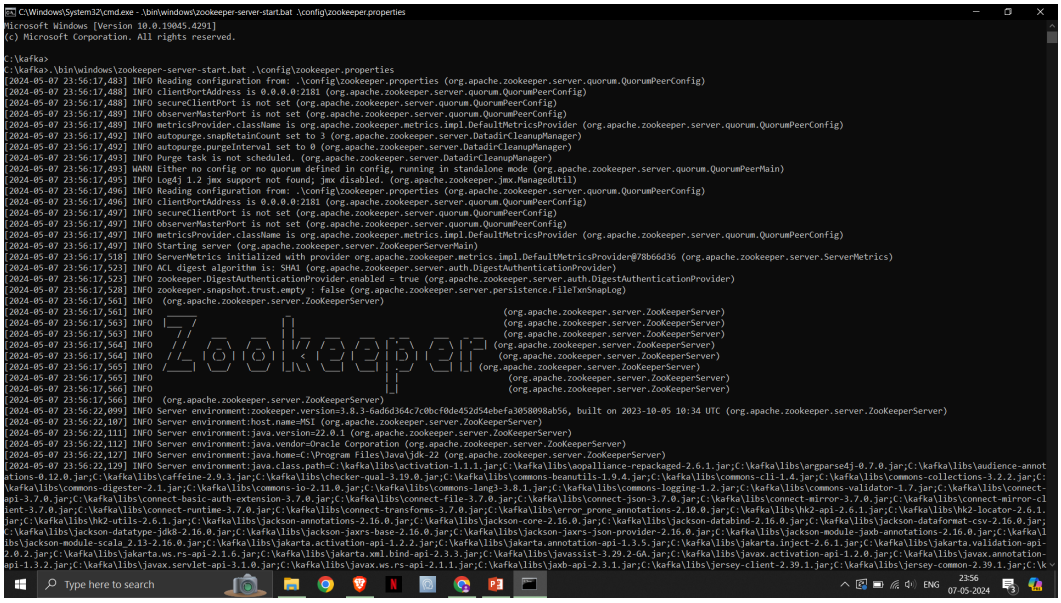

Figure 1: Zookeeper server startup using `zookeeper-server-start.bat` **(1)**

- Started Kafka server using `kafka-server-start.bat`.

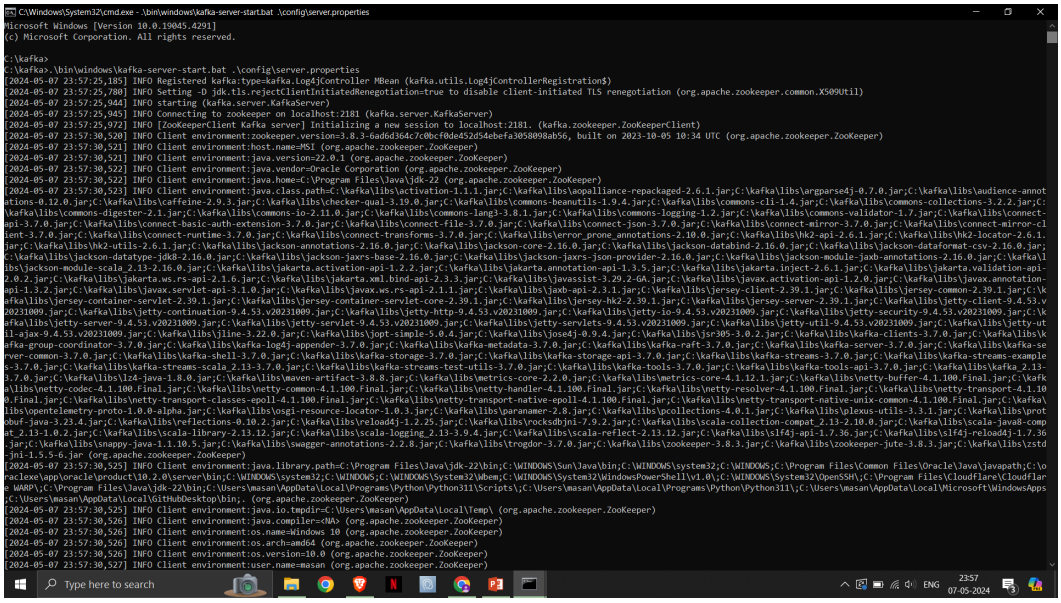

Figure 2: Launching Kafka server with `server.properties` configuration **(2)**

2 Creating Kafka Topic

- Created a Kafka topic named "test" using the `kafka-topics.bat` command.

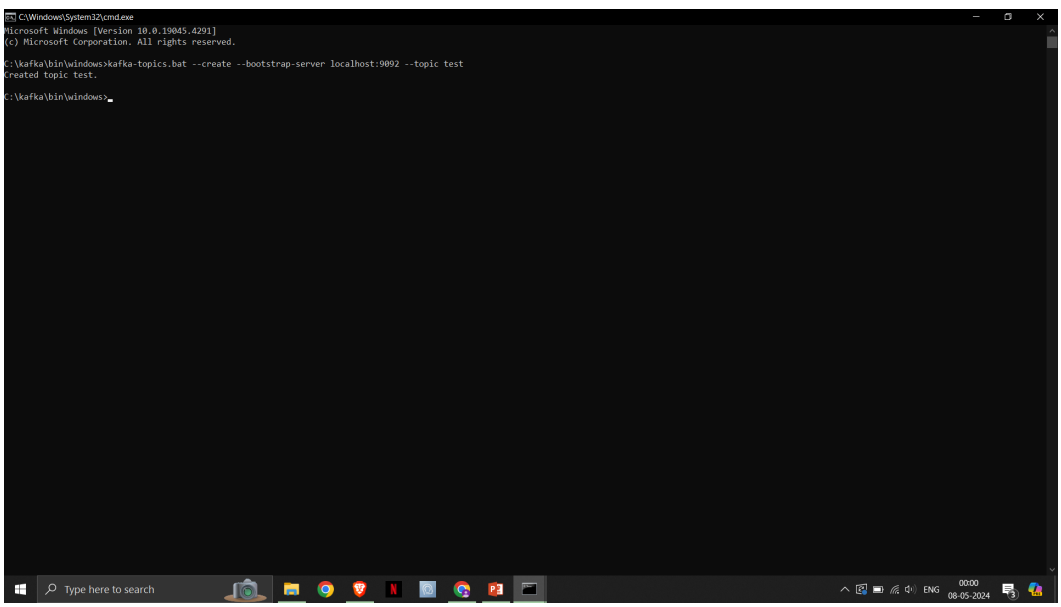

Figure 3: "Topic 'test' created successfully." **(3)**

3 Producing E-commerce Orders

- Developed Python scripts to act as Kafka producers.
- Producer 1: Inventory Orders Producer
  - Subscribed to the Kafka topic "test" to consume inventory messages.
  - Printed received inventory messages to the console.
- Producer 2: Delivery Orders Producer
  - Subscribed to the Kafka topic "test" to consume delivery messages.
  - Printed received delivery messages to the console.

4 Consuming E-commerce Orders

- Developed Python scripts to act as Kafka consumers.
- Consumer 1: Inventory Data Consumer
  - Subscribed to the Kafka topic "test" to consume inventory messages.
  - Printed received inventory messages to the console.

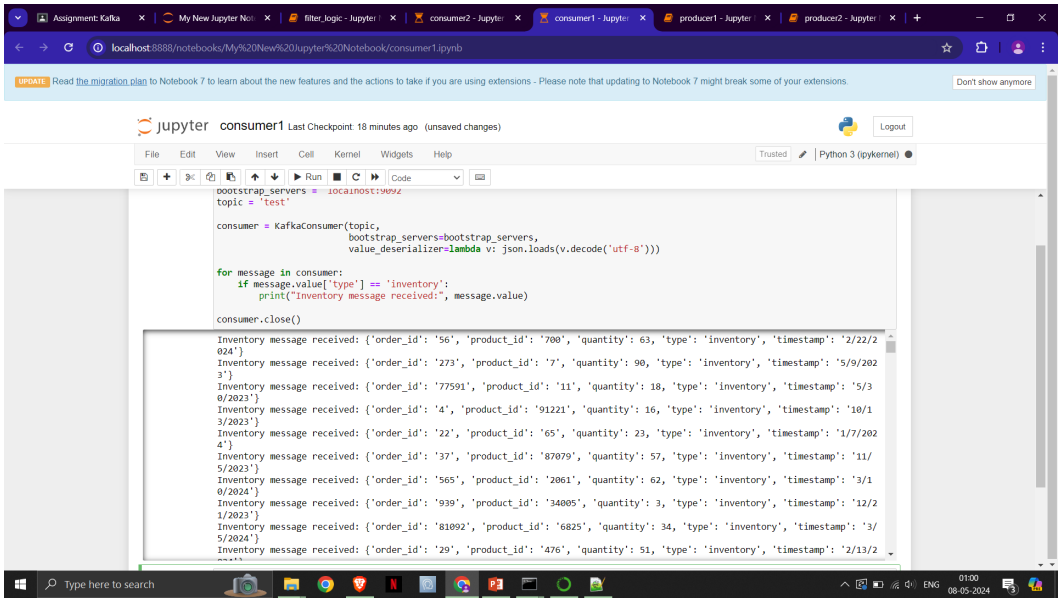

Figure 4: Consumer1 Result **(4)**

- Consumer 2: Delivery Data Consumer
  - Subscribed to the Kafka topic "test" to consume delivery messages.
  - Printed received delivery messages to the console.

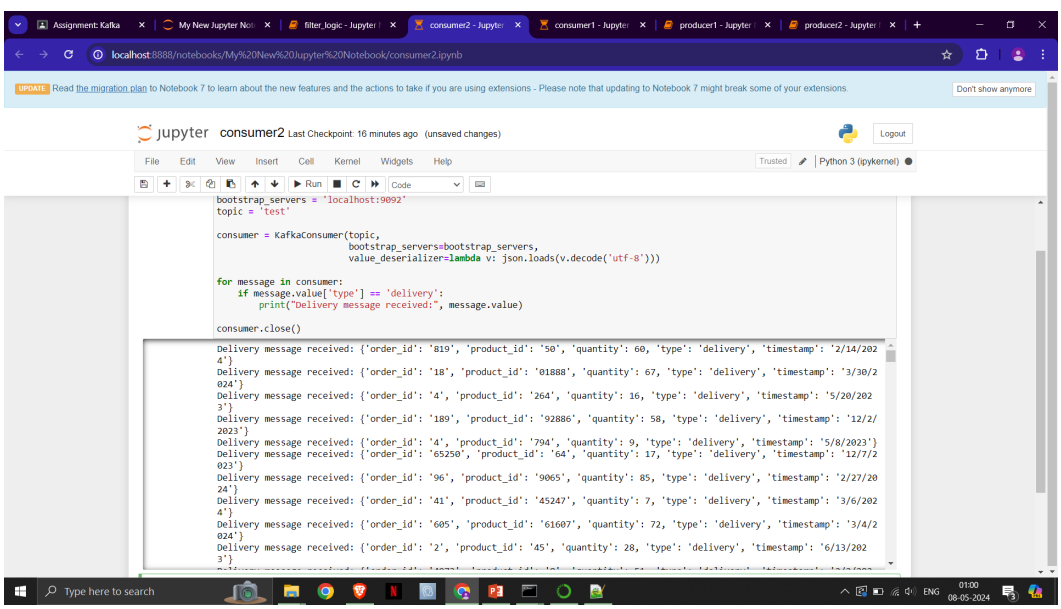

Figure 5: Consumer2 Result **(5)**

5 Verifying Message Consumption

- Executed the consumer scripts to verify that inventory and delivery messages were being consumed from the Kafka topic "test".
- Confirmed successful consumption of inventory and delivery messages by observing the output in the console.

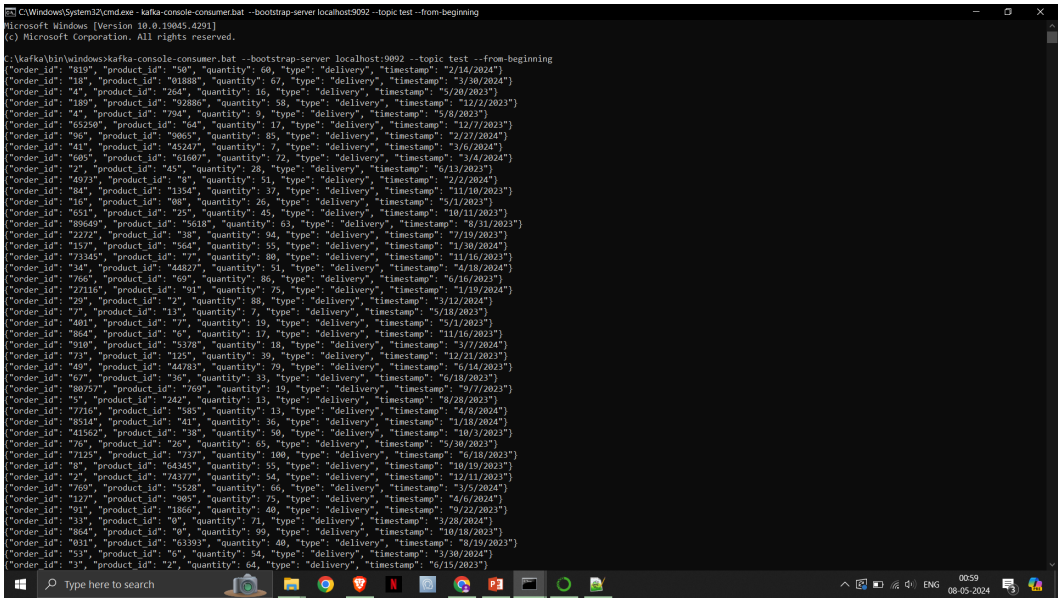

Figure 6: Output of Kafka Console Consumer displaying delivery orders from the 'test' topic **(6)**

6 Implementing Message Filtering Logic

- Developed a standalone Python script to filter inventory and delivery messages from the provided JSON data.
- Implemented separate functions for filtering inventory and delivery messages based on their "type" field.
- Printed the filtered inventory and delivery messages to the console.

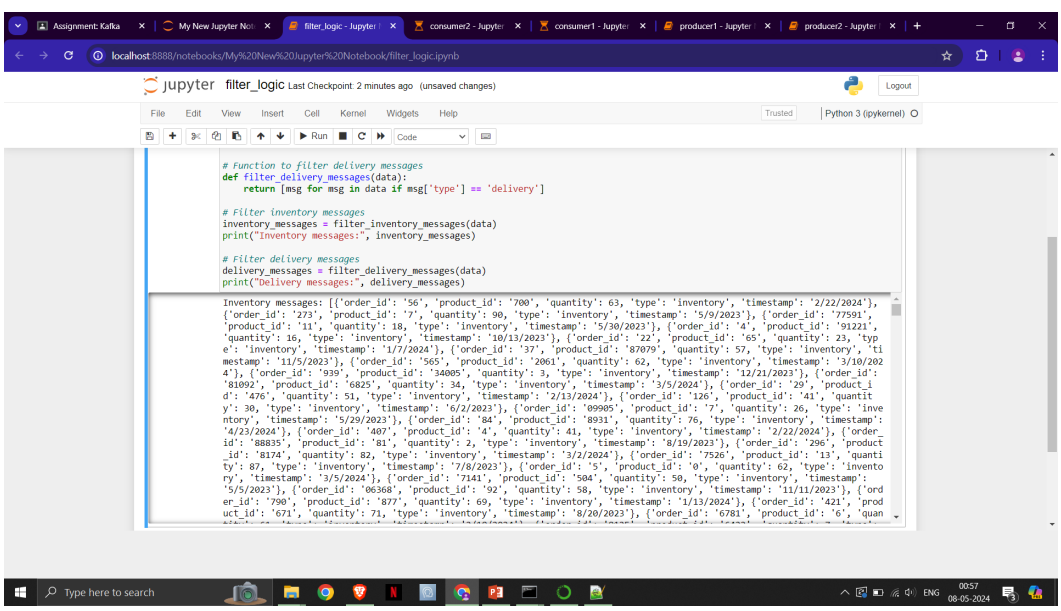

Figure 7: Filtered inventory messages displayed as part of the filter logic output **(7)**

7 Report Summary

- Successfully set up Apache Kafka environment on the local machine.
- Created a Kafka topic named "test" for handling e-commerce orders.
- Developed Kafka producer and consumer scripts to handle inventory and delivery messages in real-time.
- Tested the system by producing and consuming e-commerce orders, ensuring correct processing and output.