

# Assignment Report on Real-Time E-commerce Order Processing System Using Kafka

Mitul Dudhat (202318024)

May 7, 2024

## 1 Introduction

The objective of this assignment is to develop a Kafka-based system for managing e-commerce orders in real-time. This system will handle two critical aspects of e-commerce operations: inventory management and delivery processing.

## 2 Steps Taken

### 2.1 Step 1: Set Up Kafka

1. Installed Kafka and ensured it's running on the local system.
2. Created Kafka topics named 'inventory-topic' and 'delivery-topic' for producers to send messages to.

### 2.2 Step 2: Implement Kafka Producers

#### 2.2.1 Producer 1: Inventory Orders Producer

- Developed a Kafka producer (`inventory_producer`) that filters messages where the type field is 'inventory'.
- Read inventory-related events from a JSON data source and sent messages with type set to 'inventory' to the 'inventory-topic'.

#### 2.2.2 Producer 2: Delivery Orders Producer

- Implemented a Kafka producer (`delivery_producer`) that filters messages where the type field is 'delivery'.
- Read delivery-related events from the same JSON data source and sent messages with type set to 'delivery' to the 'delivery-topic'.

## 2.3 Step 3: Implement Kafka Consumers

### 2.3.1 Consumer 1: Inventory Data Consumer

- Configured a Kafka consumer (`inventory_consumer`) to subscribe to the 'inventory-topic'.
- Processed inventory messages received by updating inventory databases or systems accordingly.

### 2.3.2 Consumer 2: Delivery Data Consumer

- Set up a Kafka consumer (`delivery_consumer`) for the 'delivery-topic'.
- Handled delivery-related messages such as scheduling deliveries, updating delivery status, and notifying customers.

## 2.4 Step 4: Develop Message Filtering Logic

- Implemented logic within each producer to filter messages based on the type field from the JSON data source.
- Ensured that only messages matching the desired type (inventory or delivery) were sent to Kafka topics.

## 2.5 Additional Considerations

- Included error handling mechanisms within producers and consumers to manage exceptions or failed operations gracefully.
- Designed the system considering Kafka partitioning, consumer groups, and scaling strategies for scalability.
- Utilized Kafka monitoring tools and logging frameworks for effective system performance monitoring and issue troubleshooting.

## 3 Conclusion

By following these steps and best practices, a robust Kafka-based e-commerce order management system capable of real-time inventory management and delivery processing has been successfully developed.

## 4 Images

