Submitted by: Simran Dalvi (202318042)

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

To create a Kafka-based system for handling e-commerce orders in real-time, the following steps can be taken:

Phase 1: Setting Up Kafka
1. Install Kafka: Ensure the installation and activation of Kafka either on your local machine or a designated server.
2. Establish Kafka Topics: Create two Kafka topics, namely 'inventory_orders' and 'delivery_orders,' to facilitate message transmission from producers.

Phase 2: Producer Implementation
1. Inventory Orders Producer:
   - This producer is responsible for filtering messages with the 'inventory' type.
   - Develop a Kafka producer that retrieves inventory-related events from a data source and forwards messages marked as 'inventory' to the 'inventory_orders' topic.
2. Delivery Orders Producer:
   - This producer filters messages with the 'delivery' type.
   - Construct a Kafka producer that fetches delivery-related events and dispatches messages tagged as 'delivery' to the 'delivery_orders' topic.

Phase 3: Consumer Implementation
1. Inventory Data Consumer:
   - Configure a Kafka consumer subscribed to the 'inventory_orders' topic.
   - Develop processing logic to handle incoming inventory messages by updating relevant databases or systems.
2. Delivery Data Consumer:
   - Set up a Kafka consumer for the 'delivery_orders' topic.
   - Develop processing logic to manage delivery-related messages, including scheduling deliveries, updating delivery status, and informing customers.

Phase 4: Message Filtering Logic Development
1. Producer-side Message Filtering:
   - Integrate filtering logic within each producer to screen messages based on their type field from the incoming data source.
   - Only dispatch messages to Kafka if they correspond to the desired type (i.e., 'inventory' or 'delivery').

Additional Considerations

- Error Handling: Implement error handling mechanisms within producers and consumers to gracefully manage exceptions or unsuccessful operations.
- Scalability: Design the system to accommodate increasing workloads by considering Kafka partitioning, consumer groups, and scaling techniques.
- Monitoring and Logging: Employ Kafka monitoring tools and logging frameworks to effectively monitor system performance and troubleshoot any encountered issues.

By adhering to these steps and adopting best practices, a robust Kafka-based e-commerce order management system capable of real-time inventory management and delivery processing can be developed.