

# FedEx Logistics Stream Data Analysis

- **Objective:**

The goal of this assignment is to set up a data streaming pipeline using Kafka and MongoDB, and to create a dashboard for visualizing logistics data. You will:

1. Set up a Kafka Cluster on Confluent Cloud.
2. Write producer code to generate and stream mock data into a Kafka topic in Avro format.
3. Read the schema from the schema registry in the producer code.
4. Set up a MongoDB cluster on MongoDB Atlas and create a database and collection for logistics data.
5. Use Kafka Connect to stream data from the Kafka topic to the MongoDB collection.
6. Create an insightful dashboard on MongoDB Atlas.

- **Tasks:**

1. Kafka Cluster Setup on Confluent Cloud:

- ❖ Create a new Kafka cluster on Confluent Cloud.
- ❖ Configure the cluster with necessary settings.
- ❖ Note the connection details for use in the producer and Kafka Connect configurations.

2. Kafka Producer Code:

- ❖ Write a producer code in Python to generate mock logistics data and send it to a Kafka topic in Avro format.
- ❖ Sample Data Object:

```
{  
    "shipment_id": "SH123456",  
    "origin": "New York, NY",  
    "destination": "Los Angeles, CA",  
    "status": "in-transit",  
    "timestamp": "2024-07-06T12:00:00Z"  
}
```

- ❖ Use the schema registry to serialize the data. The schema should include fields like shipment\_id, origin, destination, status, and timestamp.

3. Schema Registry:

- ❖ Set up a schema registry on Confluent Cloud.
- ❖ Define the Avro schema for the logistics data.
- ❖ Update the producer code to fetch and use the schema from the schema registry.

4. MongoDB Atlas Setup:

- ❖ Create a new MongoDB cluster on MongoDB Atlas.

- ❖ Create a database named logistics.
- ❖ Create a collection named shipments within the logistics database.

5. Kafka Connect Setup:

- ❖ Set up Kafka Connect to stream data from the Kafka topic to the MongoDB collection.
- ❖ Use the MongoDB Kafka Connector available in Confluent Hub.
- ❖ Configure the connector with necessary details like Kafka topic, MongoDB connection URI, database, and collection names.

6. MongoDB Atlas Dashboard:

- ❖ Create a dashboard on MongoDB Atlas to visualize the logistics data.
- ❖ The dashboard should include charts and graphs to provide insights such as shipment status distribution, origin-destination analysis, and shipment timelines.

● **Submission Requirements:**

1. Kafka Producer Code:

- ❖ Provide the complete Kafka producer code in Python.
- ❖ Ensure the code is well-documented and includes comments explaining each part.

2. Step-by-Step Setup Process:

- ❖ Document the entire setup process for Kafka Cluster, schema registry, MongoDB cluster, and Kafka Connect.
- ❖ Include screenshots for each step to demonstrate the setup and configuration.

3. MongoDB Dashboard Screenshots:

- ❖ Include screenshots of the MongoDB Atlas dashboard showing the different visualizations.
- ❖ Each screenshot should have a brief description explaining what is being shown.

● **Submission Format:**

- ❖ Create a single document (PDF or Word) containing all the code, documentation, and screenshots.
- ❖ Ensure the document is well-organized with clear headings and subheadings for each section.
- ❖ Submit the document through the designated submission platform or email by the specified deadline.