**MongoDB Assignment**

**Assignment: Integrating MongoDB with Kafka using Python**

**Objective:** Develop a Python-based application that integrates Kafka and MongoDB to process logistics data. The application will involve a Kafka producer and consumer, data serialization/deserialization with Avro, and data ingestion into MongoDB. Additionally, an API will be developed to interact with the data stored in MongoDB.

**Requirements:**

* Basic understanding of Python, Kafka, MongoDB, and Docker.
* Access to Confluent Kafka and MongoDB Atlas.
* Familiarity with Docker and containerization.

**Tasks**

1. **Kafka Producer in Python**

* Develop a Python script to act as a Kafka producer.
* Use Pandas to read logistics data from a CSV file.
* Serialize the data into Avro format and publish it to a Confluent Kafka topic.

1. **Schema Registry Integration**

* Establish a Schema Registry for managing Avro schemas.
* Ensure that the Kafka producer and consumer fetch the schema from the Schema Registry during serialization and deserialization.

1. **Logistics Data Information**

* Logistics data contains fields like (e.g., shipment details, tracking information).

1. **Kafka Consumer in Python**

* Write a Python script for the Kafka consumer.
* Deserialize the Avro data and ingest it into a MongoDB collection.

1. **Scaling Kafka Consumers**

* Utilize Docker to scale Kafka consumers.
* Provide instructions for deploying multiple instances of the Kafka consumer using Docker.

1. **Data Validation in Kafka Consumer**

* Implement data validation checks in the consumer script before ingesting data into MongoDB.
* Validations like checking for null values, data type validation, and format checks.
* More assumptions can be taken for data validation, make sure to list down your assumptions in the submission document.

1. **API Development using MongoDB Atlas**

* Create an API to interact with the MongoDB collection.
* Implement endpoints for filtering specific JSON documents and for aggregating data.
* More assumptions & use-cases can be considered for API creation, make sure to list down your assumptions & use-cases in the submission document.

1. **Deliverables**

* Python scripts for Kafka producer and consumer.
* Sample logistics data CSV file.
* Dockerfile for scaling Kafka consumers.
* API code for MongoDB interactions.
* Documentation explaining the setup and execution of the application.