Configuration for the Taxi Co-Op

1. Taxi Sensor Simulator:

This script will simulate taxis moving and updating their location.

EC2 : Taxi Sensor Simulator.py (Will be run from EC2)

2. Data Ingestion & Storage:

For this, we'll use AWS Lambda and API Gateway for serverless API endpoints and AWS DocumentDB for storage.

**Lambda Function:**TaxiLocation.py

**Configuration**:

* Deploy the above Lambda function in AWS.
* Create an API endpoint in API Gateway that triggers the Lambda function.
* Ensure the Lambda function has the necessary permissions to interact with DocumentDB.
* In DocumentDB, create a collection for taxis and set up a geospatial index on the location field.

3. User Registration and Taxi Registration:

For simplicity, we'll create two separate Lambda functions to handle user and taxi registrations. Each registration will store the user or taxi details in separate collections in DocumentDB.

**Lambda Function**: UserRegistraion.py

**Lambda Function**: TaxiRegistration.py

4. Handling User Taxi Requests:

When a user requests a taxi, the system should find the nearest available taxis based on the user's location and taxi type preference.

**Lambda Function:** UserTaxiRequests.py

**Configuration:**

1. **User and Taxi Registration**:
   * Deploy the user and taxi registration Lambda functions in AWS.
   * Create API endpoints in API Gateway for user and taxi registrations.
   * Link the API endpoints to the respective Lambda functions.
2. **User Taxi Request**:
   * Deploy the user taxi request Lambda function in AWS.
   * Create an API endpoint in API Gateway for user taxi requests.
   * Link the API endpoint to the Lambda function.
3. **Permissions**:
   * Ensure all Lambda functions have the necessary permissions to interact with DocumentDB.