NSP Bolt Ride – Real-Time Trip Processing Project

Objective: Design and implement a real-time trip data ingestion and analytics pipeline.

Project Context

NSP Bolt Ride is a ride-hailing company that operates through a mobile app. As part of the data engineering team, your role is to implement a system that processes and enriches trip data in near real time to support analytics and operations.

This project simulates a real-world, event-driven architecture and will test your ability to build scalable ingestion, processing, and aggregation pipelines using AWS-native services.

Requirements

Your system must:

- Ingest **trip start** and **trip end** events through streaming infrastructure.
- Store and update trip data in a single NoSQL database using the trip_id as the primary key.
- Trigger a transformation process when a trip is marked complete.
- Write the aggregated results to a downstream data store for analytics.

You are expected to use the following AWS services:

- **Amazon Kinesis** (for event ingestion)
- **AWS Lambda** (for stream processing)
- **Amazon DynamoDB** (for storage)
- **AWS Glue or custom logic** (for aggregation)
- **Amazon S3** (for storing final output)

You may design the structure and schema of the events and tables as you see fit.

KPI Requirements

The system must produce **daily-level metrics** for completed trips. These metrics must be written to Amazon S3 as a structured JSON file.

Minimum KPIs (per day):

- total_fare: The sum of all fares collected from completed trips on a given day.
- count_trips: The total number of completed trips for the day.
- average_fare: The average fare amount per trip for the day.
- max_fare: The highest single fare recorded among completed trips for the day.

- min_fare: The lowest fare among completed trips for the day.

The output file should be timestamped and organized for efficient access.

Constraints

- Trip start and end data will arrive independently and may not be perfectly ordered.
- Only completed trips (i.e., those with both start and end data) should be considered for aggregation.
- Your solution must be modular and scalable.