Workshop: **Unleashing Real-Time Insights Using Data Processing**

In this hands-on session, you need to tackle a real-world challenge faced by the Bangalore Traffic Police. Your mission is to design and implement a robust system that enables the monitoring of incidents in specific areas of Bangalore using real-time data.

Problem Overview:

The Bangalore Traffic Police is seeking a system to enhance their incident monitoring capabilities. The system should pull real-time data from a provided API, process the information, create aggregates, and store the results for effective monitoring. The goal is to empower the police force with timely insights into traffic incidents, enabling quicker response times and improved overall traffic management.

Key Requirements:

API Integration:

- Participants are required to interface with a provided API that streams real-time traffic data.
- When you query the API you will receive a signed S3 URL. You need to fetch the data from the URL and start processing it. The data will be in the form of a parquet file.
- There will be new data every 15 mins. Your system needs to query this API every 15 mins, get the fresh data and process it.

Real-time Data Processing:

- Implement a system capable of processing incoming data in real-time.
- Apply necessary transformations and aggregations to derive meaningful insights from the raw data.
- The aggregations we are interested in are given below.

Storage:

- Develop a mechanism to store the processed data efficiently for historical analysis and reporting.

- Choose an appropriate storage solution (e.g., database, cloud storage) for the aggregated information.
- You may not store all the data that comes in. We are only interested in storing the aggregated data.

Trigger an event:

- Participants need to alert the authorities upon certain events. The data they aggregate should contain the required information for the alerts
- The points when participants need to alert are provided below

Visualization (optional):

- Optionally, participants can incorporate a visualization component to present the processed data in a user-friendly format.
- Provide insights through charts, graphs, or any other suitable visualization method.

Cost, effort and scale:

- The solution needs to be cost effective.
- The effort required to set it up should be minimal
- Since they don't have a huge technical force, they need this to be low on maintenance and easily scalable without human intervention.

API Payload

Query the following service with a GET request to receive the current parquet file URL.

```
https://gikfcv41s7.execute-api.ap-south-1.amazonaws.com/production/
```

Response of the query will be as follows:

```
{"url":"https://dev-day-blr-feb-2024.s3.ap-south-1.amazonaws.com/11%3A30.parquet?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=ASIAU2FGRBGMUCG3IE3I%2F20240202%2Fap-south-1%2Fs3%2Faws4_request&X-Amz-Date=20240202T121245Z&X-Amz-Expires=..."}
```

Given below is the format of the payload you can expect to see in the parquet file you receive. (explain lat lon places and how they are stored)

```
{
  "id": "string",
  "parentId": "string",
  "type": "string",
  "time.startTime": "string",
  "time.endTime": "string",
  "time.entryTime": "string",
  "roadClosed": "string",
  "restrictions.vehicleType": "string",
  "restrictions.restrictionType": "string",
  "location.west": "float64", # Latitude and longitude coordinates
  "location.east": "float64",# bounding the event area
  "location.north": "float64",
  "location.south": "float64"
}
```

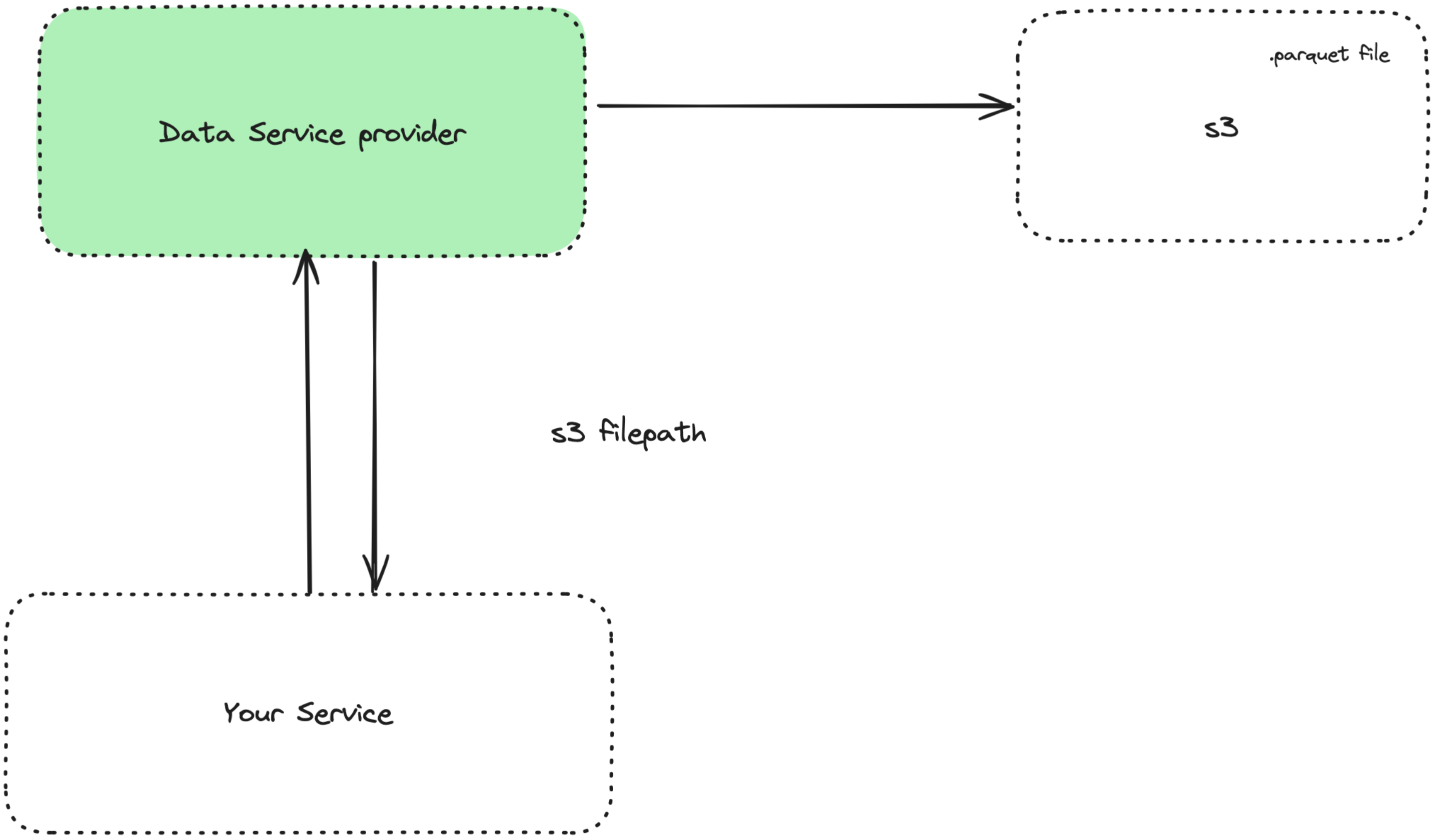
Aggregations & Event Triggers

1. Aggregate the number of incidents happening across each region. If the number of incidents exceed 1000 at any location, then send a get request to the following endpoint.

   GET https://skinny-siler-ixn-171.lockjaw.link/?lat=<loc.west>&lon=<loc.north>

2. Find the average time to resolve for events across locations
3. Given the following criteria classify the criticality of events in a location
   a. If incidents with roadClosed is true < 1,000 then criticality = low
   b. If incidents with roadClosed is true > 500 but less than 1,000 then classify it as minor and number of incidents > 10,000 then classify it as minor
   c. If incidents with roadClosed is true > 1,000 and number of incidents > 10,000 then classify it as major
4. If the criticality of a location is classified as major, you need to send a request to the URL specified above.

5. Find out the number of events at a location that propagated and caused other events.
6. Find out which type of event caused the most number of incidents at a location


Data Service provider
.parquet file
s3
s3 filepath
Your Service