PROJECT TITLE – 4

Application Monitoring Dashboards

1. Project Overview

The Log Analytics Platform is designed to collect, process, and visualize log data in real time. It leverages Grafana for visualization, Kafka for real-time log ingestion, and a relational database for storage.

2. Functional Requirements

The platform must be able to visualize the following metrics in Grafana:

- Request Count per Endpoint: Track the number of requests made to each API endpoint.
- Response Time Trends: Display response time patterns over different time periods.
- Most Frequent Errors in the Application: Identify and highlight recurring errors.
- Real-Time Logs: Provide a live feed of logs for monitoring purposes.

NOTE - You can add more dashboards if you think that it benifits your monitoring application.

3. Non-Functional Requirements

- All components must be containerized using Docker for easy deployment and scalability.

4. Technology Stack

The implementation is flexible in terms of programming languages and frameworks. However, the following components must be included:

- Containerization: Docker
- Message Broker: Apache Kafka for log ingestion
- Visualization: Grafana for querying and visualizing log data

5. Implementation Steps

Week 1: Infrastructure and API Development (15m)

- Develop/Use a REST API server with 5-10 endpoints.
- Create a script to simulate workload by generating API requests.
- Install and configure Apache Kafka with topics for different log types.
- Implement a producer that pushes logs into Kafka topics.
- Set up Docker containers for all components.

Week 2: Log Processing and Storage (10m)

- Develop a Kafka consumer that reads logs, processes them, and stores them onto a database(s).
- Design the schema for log storage and configure the database(s).

Week 3: Visualization (15m)

- Connect Grafana to the database and configure dashboards.
- Write queries to extract required metrics and visualize them.
- Implement monitoring mechanisms.

Resources

- In case you don't want to develop your own REST API server(Add more endpoints in case it's too less). This repo uses js. Note that you can use any language for your server development.
  - json-server (JS-based REST API)
- In case you want to use confluent kafka, you can check out the link below. They have repos that explain how to create producers and consumers in many different languages. You will have to search through their repos for these sample producers and consumers.
  - Confluent Kafka Repositories
- You can check out how to set up Kafka containers and use them.
  - Kafka on Docker
  - Confluent Kafka on Docker
- You can use more than a single database for the monitoring application you develop. Some good choices for the same is mentioned below. You can read more about why they are good choices in the **NOTE** section.
  - Prometheus - for telemetric data
  - Loki - for logs

NOTE

- The stack you will be using over here is not a classic monitoring stack that gets used most often. You can read more about them here
  - ELK Stack
  - PLG Stack