Department of Computer Science and Engineering (Data Science)

Subject: Big Data Engineering (DJ19DSL604)

AY: 2023-24

Experiment 10

(Mini Project)

Aim: Design the infrastructure of a Big Data Application.

Tasks to be completed by the students:

Task 1: Choose a problem definition which requires handling Big Data.
Task 2: Design the data pipeline for your application.
Task 3: Deploy your project on suitable platform.
Task 4: Test your application with different volume, variety and velocity of data.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)

Report on Mini Project

Big Data Engineering (DJ19DSL604)

AY: 2023-24

Real-Time Dashboard for Flight data

Tanisha Gandhi: 60009210197

Kresha Shah: 60009220080

Durva Patel: 60009220088

Guided By:

Mohammed Adil Shaikh

CHAPTER 1: INTRODUCTION

In this project, we will use a real-time flight tracking API, Apache Kafka, ElastichSearch and Kibana to create a real-time Flight-info data pipeline and track the flights in real-time. We will use a high-level architecture and corresponding configurations that will allow us to create this data pipeline. The end result will be a Kibana dashboard fetching real-time data from ElasticSearch.

We started by collecting in real-time Flight informations (Aircraft Registration Number,Aircraft Geo-Latitude,Aircraft Geo-Longitude,Aircraft elevation,Flight numbe...) and then we sent them to Kafka for analytics.

The data is ingested from the flight streaming data API and sent to a kafka topic. You need to run Kafka Server with Zookeeper and create a dedicated topic for data transport.

In Spark Streaming, Kafka consumer is created that periodically collect data in real time from the kafka topic and send them into an Elasticsearch index.

You need to enable and start Elasticsearch and run it to store the flight-info and their realtime information for further visualization purpose. You can navigate

Kibana is a visualization tool that can explore the data stored in elasticsearch. In our project, instead of directly output the result, we used this visualization tool to visualize the streaming data in a real-time manner.You can navigate to http://localhost:5601 to check if it's up and running.

CHAPTER 2: DATA DESCRIPTION AND ANALYSIS

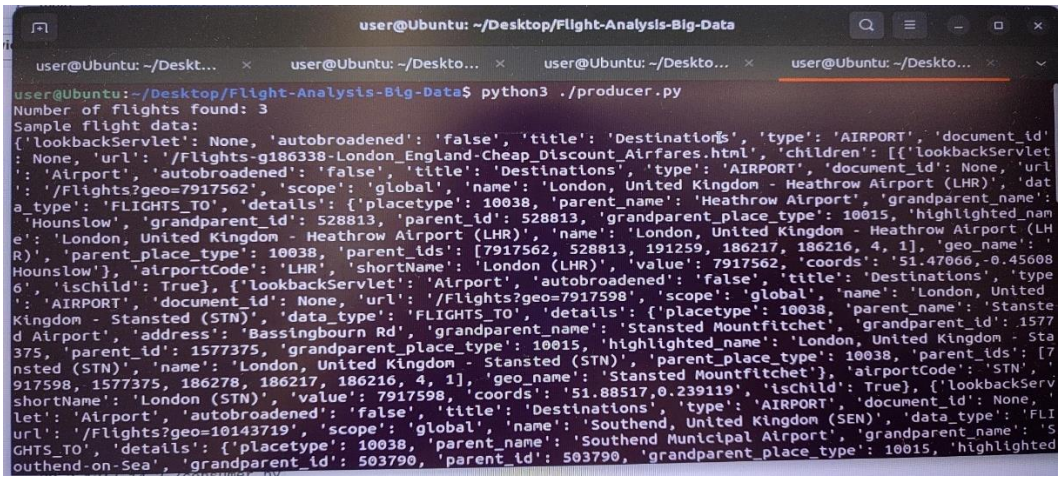
The data used in this project exhibits certain characteristics that influence its processing and analysis.

In terms of volume, the input data is sourced in real-time, indicating a continuous stream of information. Specifically, the API response volume amounts to approximately 1000 flights, denoting a considerable amount of data flowing into the system continuously.

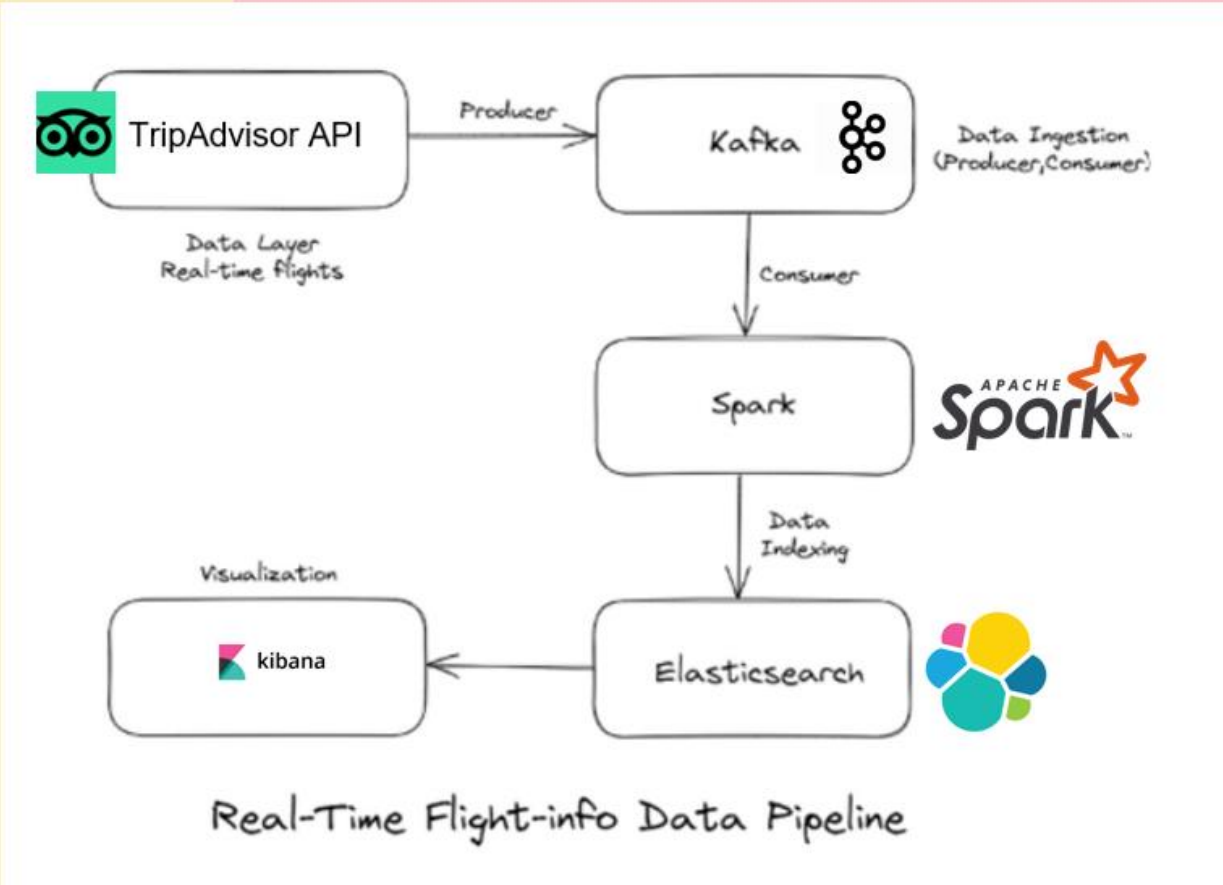
Regarding velocity, the use of RapidAPI facilitates swift analysis of text data, ensuring that insights can be derived promptly from the incoming stream of flight information.

Variety plays a significant role in the data landscape of this project. The data encompasses various aspects of flight information, including search flights, filters, airport searches, and multi-city flight searches. This diversity of data types enhances the depth and breadth of analysis that can be performed on the flight information.

Lastly, the veracity of the data is notably high, as indicated by an R-squared value of 0.9984. This metric signifies a high degree of accuracy and reliability in the data, instilling confidence in the insights derived from its analysis.



CHAPTER 3: DESIGN OF DATA PIPELINE



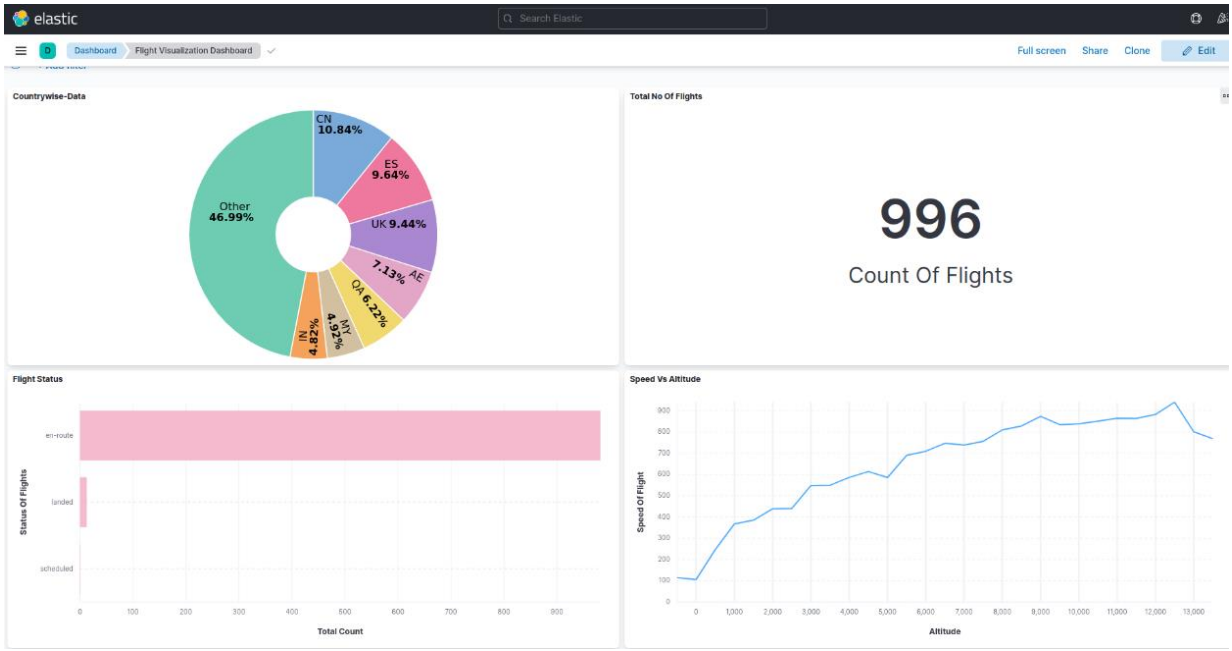
This pipeline extracts real-time flight data from TripAdvisor API, processes it, and stores it for visualization purposes.

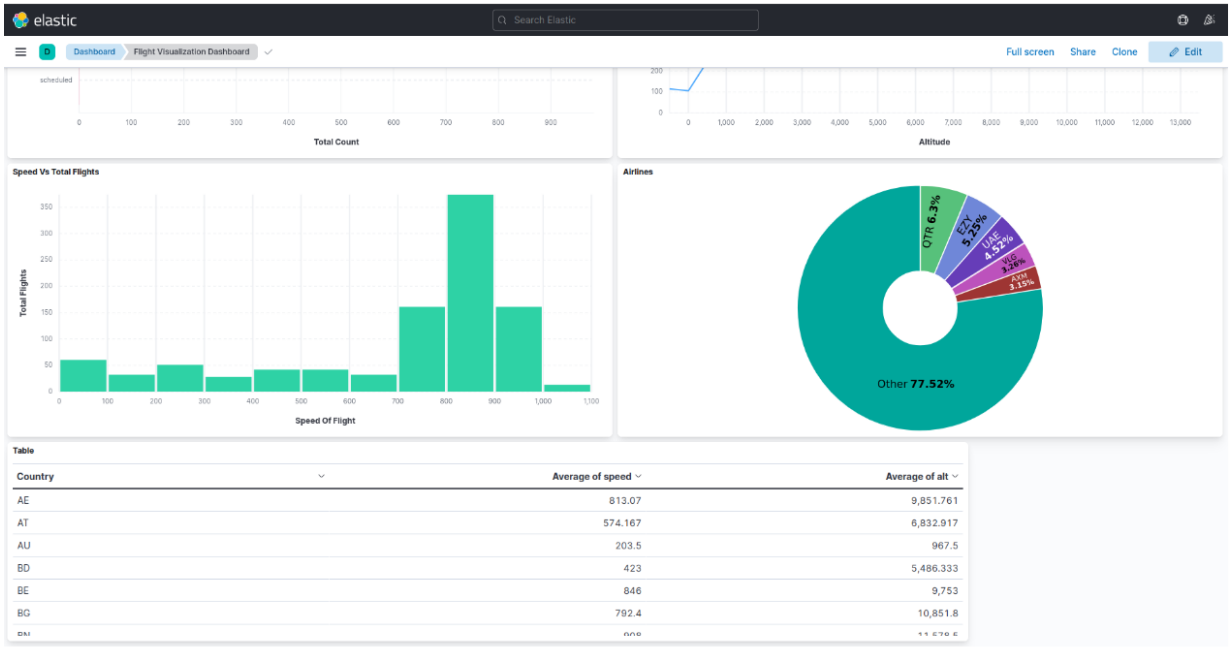
Here's a breakdown of the pipeline:

- **TripAdvisor API** acts as the **producer** in this pipeline. It continuously pushes real-time flight information into the pipeline.

- **Kafka** is a streaming platform that acts as an intermediary between the producer and the consumer. It ingests the data from the TripAdvisor API and distributes it to the consumer.
- **Data Layer (Consumer)**: This layer receives the real-time flight data from Kafka.
- **Spark** is used for data indexing. It processes the incoming flight data and prepares it for storage in Elasticsearch.
- **Elasticsearch** is a search engine that stores the processed flight data.
- **Kibana** is a data visualization tool that allows users to interact with and analyze the data stored in Elasticsearch. It essentially creates real-time flight information dashboards.

CHAPTER 4: RESULT ANALYSIS





CHAPTER 5: CONCLUSION AND FUTURE SCOPE

Conclusion:

In conclusion, the development of a real-time flight-information data pipeline utilizing Apache Kafka, Apache Spark, Elasticsearch, and Kibana has proven to be a successful endeavor. Through the integration of these cutting-edge technologies, we have achieved the creation of a dynamic system capable of ingesting, processing, storing, and visualizing real-time flight data efficiently.

The project has demonstrated the feasibility and effectiveness of leveraging modern data technologies to address the challenges associated with handling large volumes of real-time data. By harnessing the power of Apache Kafka for data ingestion, Apache Spark for data processing, Elasticsearch for data storage, and Kibana for data visualization, we have established a comprehensive solution for tracking and monitoring flights in real-time.

Future Scope:

While the current implementation of the real-time flight-information data pipeline has achieved its primary objectives, there are several avenues for future enhancement and expansion:

1. Advanced Analytic: Explore advanced analytics techniques to derive deeper insights from the flight data, such as predictive analytics for flight delays or anomaly detection for identifying unusual flight patterns.

2. Enhanced Visualization: Enhance the Kibana dashboard with additional visualizations and interactive features to provide more comprehensive insights into flight patterns and trends.

3. Integration with External Systems: Integrate the data pipeline with external systems, such as airline reservation systems or weather forecasting services, to enrich the flight data and provide more contextually relevant information.

4. Scalability and Performance Optimization: Optimize the performance and scalability of the data pipeline to handle even larger volumes of real-time data and accommodate future growth in data sources and users.

5. User Customization: Implement features that allow users to customize their dashboard views and set up alerts based on specific flight criteria or events.

6. Geospatial Analysis: Incorporate geospatial analysis capabilities to visualize flight routes on maps and analyze spatial relationships between flights and geographic features.