**Real-Time Race Lap Time Data Engineering Project using Kafka and AWS**

**🌐 Project Overview**

This project simulates a real-time data pipeline for racing lap times using Apache Kafka, Python, and AWS services including EC2, S3, Glue, and Athena. It demonstrates how to stream time-series data using Kafka and store, catalog, and query it on AWS.

**⚖️ Technologies Used**

* **Programming Language**: Python
* **Message Broker**: Apache Kafka (hosted on EC2)
* **Cloud Provider**: Amazon Web Services (AWS)
  + S3 (Simple Storage Service)
  + Glue Crawler
  + Glue Catalog
  + Athena
  + EC2 (Elastic Compute Cloud)
* **Data Format**: CSV

**📆 Architecture Breakdown**

**□ Step 1: Dataset**

* The process begins with a CSV dataset that simulates lap times of 20 drivers in a 52-lap race.

**□ Step 2: Python Race Simulation App**

* A Python script reads the CSV dataset and simulates real-time lap time data.
* Boto3 SDK is used for AWS interactions.
* This script acts as a **Kafka Producer**, sending data to Kafka topics hosted on an EC2 instance.

**□ Step 3: Apache Kafka on EC2**

* Kafka is installed and configured on an AWS EC2 instance.
* The EC2 instance hosts Kafka and acts as a broker.
* Producers send real-time race data to Kafka topics.

**□ Step 4: Kafka Consumer**

* Another Python script acts as a **Kafka Consumer**.
* It reads streaming lap time data from Kafka and writes it to **Amazon S3** in near real-time.

**□ Step 5: Amazon S3**

* Acts as a data lake where consumer data is stored.
* Data is partitioned based on date/time for better performance.

**□ Step 6: AWS Glue**

* **Glue Crawler** crawls the S3 bucket and identifies the schema.
* Populates the **AWS Glue Data Catalog**, which organizes metadata.

**□ Step 7: Amazon Athena**

* Athena queries the real-time data directly from S3 using SQL.
* Integrated with the Glue Catalog for schema referencing.

**📁 File Structure**

* Kafka\_Producer.ipynb: Jupyter notebook for the producer script.
* Kafka\_Consumer.ipynb: Jupyter notebook for the consumer script.
* lap\_times\_with\_precise\_milliseconds.csv: Dataset used for race simulation.
* Architecture.jpg: Architecture diagram.

**⚙️ How to Run the Project**

**1. Setup EC2 and Install Kafka**

* Launch an EC2 instance.
* Install Java, Zookeeper, and Kafka.
* Start Zookeeper and Kafka services.

**2. Kafka Producer**

* Load and preprocess the race dataset using Pandas.
* Create a Kafka producer with the topic race-lap-times.
* Simulate driver lap times and publish them at intervals.

**3. Kafka Consumer**

* Connect to the same Kafka topic.
* Fetch incoming lap time data and store it in a local file or S3 bucket.
* Ensure batching or time-window-based writes for efficiency.

**4. S3 Setup**

* Create an S3 bucket.
* Upload or stream data to it from the consumer script.

**5. Glue Crawler and Catalog**

* Create a Glue Crawler to scan the S3 bucket.
* Run the crawler to generate table schema.
* Verify in the Glue Data Catalog.

**6. Query with Athena**

* Open Athena in AWS console.
* Connect to the Glue catalog and run SQL queries on the lap time data.

**📈 Use Cases**

* Real-time race performance analytics
* Driver lap time tracking
* Race commentary dashboards
* Motorsport team strategy development

**📅 Future Improvements**

* Integrate with real-time telemetry or race APIs
* Add Spark Streaming for high-throughput analysis
* Set up visual dashboards using QuickSight or Grafana
* Add driver comparison insights and sector-based splits

**📍 Summary**

This project demonstrates a full pipeline from real-time lap time data ingestion (Kafka) to storage (S3), metadata management (Glue), and querying (Athena). It's a strong foundation for building more advanced data engineering and analytics solutions for motorsports.