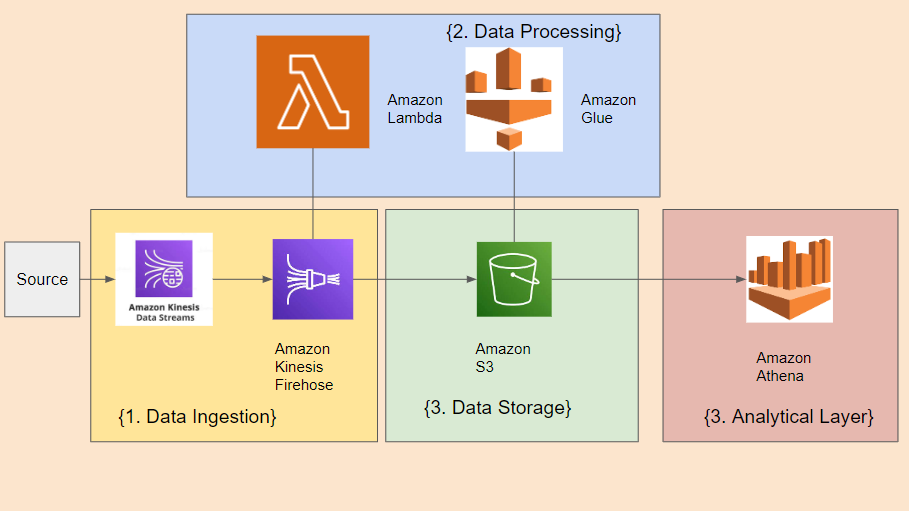
Q8) You are tasked with setting up a system to analyze real-time data using Athena. Describe the architecture of the system and how you would set up the ETL process to make the data available for analysis in Athena.

Real-time stream data with Athena



**Architecture Overview:**

1. Data Ingestion

Real-time data streams from various sources such as RDS, API gateway or other form of data sources are ingested into the data processing system.

2. Data Transformation

Before the data is stored, light data processing will be done. The data will be batched into small chunks. This is done with Lambda.

After the data is stored, further transformation is required.

ETL tool is used to pick up the data to process the data further for better analytical purposes.

3. Data Storage

Processed data is stored in Amazon S3, making it queryable by AWS Athena, which enables SQL-based ad-hoc querying and analysis.

4. Analytical Layer

Athena serves as the querying and analytical layer, allowing educators, data scientists, and analysts to run SQL queries on the data stored in S3 without the need for a traditional database.

**Setting Up the ETL Process:**

**Step 1: Data Ingestion**

- Utilize AWS services Amazon Kinesis Data Streams to ingest real-time data from sources such as API or dynamoDB.

- Ensure robust and reliable data ingestion mechanisms, capable of handling varying data rates and spikes.

**Step 2: Data Processing**

- Implement real-time data processing using tools AWS Lambda.

- Cleanse, transform, and enrich the data to derive meaningful insights. For example, calculate student engagement scores or identify at-risk students based on their behavior.

- it is helpful to include audit columns such as created\_time\_date, modified\_time\_date.

- Design data processing with scalability and fault tolerance in mind to accommodate growing data volumes.

**Step 3: Data Storage**

- Store the processed data in Amazon S3, adhering to an organized folder structure and utilizing columnar file formats (e.g., Parquet) for optimal query performance.

- Organize the S3 folder structure with data partition in mind.

- Adopt a data lake approach, allowing both raw and processed data to coexist, facilitating flexibility for future analyses and data exploration.

**Step 4: ETL for analytics**

- While lambda can be used to transform the data, it has limitations in terms of heavy transformation. Lambda also has a limit of 15 minutes runtime. With the distributed capabilities and spark, AWS glue is the suitable ETL tool.

- Script out the transformation, keep in mind that the data should be denormalized for analytical purposes.

**Step 5: AWS Athena Configuration**

- Define an AWS Glue Data Catalog database and tables that describe the schema of the data stored in S3. AWS Glue Crawlers can be employed to automate this process, especially if schema changes occur frequently.

- Optimize Athena queries by partitioning data based on relevant columns (e.g., timestamp, student ID) and employing columnar storage formats.

- Establish stringent access controls and permissions to safeguard Athena queries and data assets.

**Step 6: Query and Analysis**

- Enable educators and data analysts to leverage AWS Athena for SQL-based ad-hoc querying, exploration, and analysis.

- Athena delivers query results swiftly, facilitating real-time insights and informed decision-making.

- The query could be displayed in AWS quicksight for further insight.

- Moreover, the query could also be used for machine-learning with the help of data scientists.

Considerations:

1. Cost Optimization

Monitor and manage costs associated with data storage in S3 and query execution in Athena, utilizing features like result caching and cost allocation tags.

2. Data Quality

Implement robust data quality measures during ETL, including error handling, validation checks, and comprehensive logging of processing failures.

3. Scalability

Design the system architecture to horizontally scale to accommodate increased data volume and concurrent queries as the educational platform grows.

4. Real-time vs. Near-real-time

Define and meet latency requirements for real-time data analysis to ensure timely insights.

5. Data Retention

Establish data retention policies for both raw and processed data, addressing storage costs and compliance obligations.

6. Monitoring and Alerts

Implement comprehensive monitoring and alerting systems to promptly detect and respond to issues related to ETL, data ingestion, and query performance.

7. Data Security

Enforce robust data security practices, including encryption, access controls, and authentication, to safeguard sensitive PDPA data.

8. Athena in real-time analytics

Real-time data processing requires a high workload, while Athena is capable of providing insight, a real-time analytics insight such as AWS Glue is a better option. Athena is not built for real-time analytics as it has noticeable latency while it does the query especially if the query is complex with joins.