Build Real-Time Data Pipeline using AWS Kinesis and Snowflake

Business Overview

A real-time data pipeline is a critical data engineering practice designed to seamlessly integrate live data from diverse sources such as IoT devices, clickstream data, APIs, and web servers. This real time generated data is then transformed into meaningful insights for analytics and loaded into designated destination systems in real-time. This approach has become indispensable for organizations aiming to manage large volumes of real-time data efficiently and automatically.

Real-time data pipelines enable organizations to process data as soon as it is generated, facilitating instant insights and analytics. This capability is essential for a wide range of applications, including online stores, startups, and AI companies. By deploying real-time data pipelines, these organizations can drive insights rapidly and effectively.

Key features:

1. **Complex Workflow Handling:** Real-time pipelines manage intricate data workflows seamlessly.
2. **Data Quality Assurance:** They ensure data accuracy, completeness, and manage upserts during transformations/ loading.
3. **Unified Data Integration:** They consolidate data from diverse sources into a cohesive format.
4. **Cost Efficiency:** By updating destination systems with only new records, real-time pipelines optimize computational resources and reduce costs effectively.

Real-time ETL pipelines operate by applying predefined business rules to ensure high-quality outcomes, such as real-time customer demographic reports, live user engagement metrics, and continuous updates for autonomous vehicles. These pipelines also manage complex interactions with backend databases and systems to maintain throughput and data integrity.

While other data warehouses can be considered, Snowflake stands out for its ability to handle semi-structured data in real-time, making it a preferred choice. In this real-time data project, we automated processes using AWS services and Snowflake functionalities. This automation ensures reliability and reduces the risk of errors that can occur with manual programming.

One significant advantage of using AWS services and Snowflake is their robustness and reliability. By leveraging these in-house tools, we ensure the solution's availability, usability, and durability, which are crucial for maintaining a stable and efficient data pipeline.

Aim:

The aim of this project is to analyze real-time global market data using AWS Kinesis and Snowflake. We utilize CSV datasets extracted via API calls, stream them through Kinesis Firehose, and transform them with Snowflake. Our agile workflow ensures efficiency, providing a one stop comprehensive solution for real-time data insights. AWS services include S3 for storage, Kinesis Firehose for streaming, Lambda for serverless computing, AppRunner for deployment of API, Snowpipe for auto ingestion of raw data from destination S3 bucket, Snowpark for processing raw data into cleaned data and Power BI for data visualization. This project leverages AWS services like S3, Kinesis Firehose, Lambda, and AppRunner for data management and API construction. Snowflake acts as the robust data warehouse, while Power BI visualizes real-time global market insights.

Tech Stack:

- ➢ Programming : Python, SQL
- ➢ AWS Services : AWS Kinesis, AWS App Runner, AWS Kinesis Firehose, AWS S3, AWS LAMBDA
- ➢ Warehouse: Snowflake
- ➢ Visualization : Power BI
- ➢ Libraries : Boto3, requests, pandas

AWS KINESIS FIREHOSE:

AWS Kinesis acts as a key ETL (Extract, Transform, Load) tool of AWS Service to manage real-time data workflows with exceptional efficiency, scalability, and reliability.

AWS Kinesis served as the backbone of our data streaming architecture, enabling seamless extraction of data from multiple sources such as web servers and IoT devices in real-time. Its ability to automatically scale according to data volume fluctuations ensured that we could handle varying workloads without compromising performance or reliability. Overall, AWS Kinesis played a pivotal role in enhancing the efficiency and scalability of these real-time data workflows, enabling us to derive actionable insights and maintain high data quality throughout our project lifecycle.

AWS S3:

AWS S3, or Simple Storage Service, S3 Bucket is a versatile object storage solution provided by Amazon Web Services for securely storing and retrieving a wide range of data types, including semi-structured, unstructured, and structured data. It serves as a fundamental component for creating data lakes, which are repositories that store vast amounts of raw data in its native format for various analytical purposes.

In the context of this project, AWS S3 was utilized to store the data generated by our use case. This data was subsequently accessed and processed via APIs, demonstrating S3's capability to act as a reliable data source for real-time and batch processing scenarios. Moreover, AWS S3 served as a destination for streaming data ingestion from AWS Kinesis Firehose, facilitating efficient and scalable data storage and retrieval.

Special permissions were configured for S3 in this project to enable secure access to endpoints, ensuring that authorized users and organizations could interact with the data stored in S3 through APIs. These permissions went beyond the default settings of S3 to let anyone access their required endpoints from the dataset.

AWS LAMBDA :

AWS Lambda is a serverless service from Amazon Web Services (AWS) that lets you run code without worrying about servers. It automatically adjusts to handle more requests or real time event logs as needed and runs your code in response to events from other AWS services or HTTP requests.

In this project, AWS Lambda was utilized alongside AWS Kinesis Firehose, which streamed real-time logs in JSON format. The Lambda function was specifically designed to transform these JSON logs into CSV format. This transformation was crucial as the CSV format was required for further processing and analysis on Snowflake, a data warehouse. Ideally, the Lambda function should complete this transformation within a reasonable timeframe, such as greater than 400 milliseconds for larger log volumes, to ensure seamless data transformation

AWS APP RUNNER:

AWS App Runner is a deployment and API development service provided by AWS, developers can easily create APIs and web applications using popular frameworks like FastAPI and Flask. The service automatically handles the provisioning and scaling of infrastructure, manages load balancing, and orchestrates containers in the background. This automated approach ensures rapid and reliable deployment of APIs and web applications, catering to varying workload demands seamlessly.

In this project, AWS App Runner was used to create an API to demonstrate how a FastAPI application can be built and deployed seamlessly. Initially, a GitHub repository was set up containing two essential files: (the application code) and (dependencies). App Runner integrated with this GitHub repository, enabling automatic builds whenever changes were pushed to the repository. This automated the deployment process, ensuring that updates to the API could be deployed without manual intervention.This approach not only accelerates the deployment cycle but also enhances reliability and scalability, as AWS manages the underlying infrastructure, load balancing, and scaling based on traffic patterns.

Power BI:

Power BI is a robust business analytics tool renowned for its ability to create insightful visualizations and reports using data from various sources. It provides users with an interactive interface that simplifies the process of building visualizations by allowing them to drag and drop fields onto the canvas.

In this project, Power BI played a pivotal role in creating detailed reports and visualizations across multiple sheets, enabling organizations to extract valuable insights and delve deep into crucial business metrics to foster growth. By leveraging Power BI, organizations were empowered to base strategic decisions on clean and preprocessed data rather than relying solely on predictive models. This approach ensured that decision-makers had access to accurate and actionable information, facilitating informed choices that drive business success and operational efficiency.

Snowflake :

What sets Snowflake apart is its comprehensive support for various data types, including semi-structured and unstructured data, positioning it ideally for advanced applications such as machine learning and AI. Acting as both a data warehouse and a data lake, Snowflake incorporates external stages for preprocessing data before it reaches the main storage layer. This capability streamlines real-time and batch data handling, offering a versatile solution adaptable to dynamic business needs. Its

innovative three-tier architecture distinguishes it by separating storage from compute, which not only enhances scalability and concurrency but also supports simultaneous querying and analysis by multiple users. In this project, Snowflake was employed effectively, integrating with an S3 bucket as a destination for ETL pipeline data streams. Snowpipe automation was utilized to ingest raw data seamlessly into Snowflake's raw tables, streamlining the initial data loading process.

Data description:

The dataset constitutes a comprehensive view of the global food market across 36 countries and 2,200 markets spanning from January 1, 2007, to May 1, 2023. It includes detailed information on various aspects such as country specifics, market details, and diverse food items with associated price points (low, high, open, close), inflation rates, and trust indicators for each item.

To manage this extensive dataset efficiently, it was stored in AWS S3, leveraging its capabilities to handle large volumes of data effectively and ensuring rapid access speeds essential for transferring data to FastAPI. Real-time AWS services were subsequently employed to establish a seamless workflow, optimizing data retrieval through APIs only when needed. AWS S3 and real-time AWS services were instrumental in facilitating efficient data management and enabling responsive data access for analysis and decision-making purposes.

Approach :

1. The dataset was initially stored in an AWS S3 bucket for efficient storage and accessibility.A FastAPI application was created to establish an interface for accessing the data securely. Credentials to access the S3 bucket were integrated into the FastAPI application.

2. AWS Kinesis Data Streams collected real-time data from APIs, providing a scalable source for continuous data ingestion. AWS Kinesis Firehose processed these logs in real-time, using AWS Lambda to convert them into CSV format for improved readability and compatibility in downstream applications. This streamlined process ensured efficient data transformation and integration into the data pipeline.

3. Following data processing by AWS Kinesis Firehose, a new S3 bucket was designated as the destination for storing transformed data. Kinesis Firehose then automatically loaded the processed data into this bucket, ensuring smooth and uninterrupted data flow. This streamlined approach optimized data management and accessibility, supporting efficient retrieval and utilization for subsequent analysis and operations.

4. Snowpipe, an automated data ingestion service in Snowflake, was configured to ingest raw data directly from the specified S3 bucket. Additionally, Snowpark sessions in Python were utilized to cleanse and transform raw data, preparing it for insertion into separate Snowflake tables. These scheduled Snowpark sessions operated as stored procedures, automating the workflow from API data ingestion.

5. Cleaned data loaded into Power BI; multiple sheets created for diverse market insights. Reports and visualizations in Power BI focused on country-specific market rates and metrics for strategic insights.

Key Takeaways:

- Understanding of use cases and data implications.
- Understanding of AWS services and their applications.
- Implementation of API creation using FastAPI.
- Implementation of real-time data streams accessing API endpoints.
- Creation and configuration of Kinesis Firehose with data sources.
- Setting time triggers for data transformation in Lambda functions.
- Configuration of Snowpipe and external stages in Snowflake.
- Creation of Snowpark sessions in Python for data cleaning and column filtering.
- Implementation of Snowpark sessions deployment as stored procedures.
- Creation of comprehensive interactive dashboards on Power BI.

Architecture Diagram: