Real-time Data Streaming and Stock Data Analysis using AWS

GROUP NO. 6
JASH SHAH
PARTH BORKAR
PARTH KHARKAR

Section 1: Introduction

This project, "Real-Time Data Streaming and Stock Data Analysis using AWS," aims to harness the robust, scalable, and efficient capabilities of Amazon Web Services (AWS) for dual, interconnected purposes: the in-depth analysis of stock data and the implementation of real-time data streaming. We specifically target stock data of significant technology companies collectively termed 'FANGMANT' (Facebook, Apple, Netflix, Google, Microsoft, Amazon, Nvidia, and Tesla). It explores the fluctuations and trends of the stock market for these companies, focusing on the period from June 1, 1997, to December 12, 2023. This project covers the technical aspects of AWS services like Kafka, S3, EC2, Glue, Athena, and QuickSight and delves into their practical application in handling and interpreting large volumes of financial data.

Section 1.1: Project Structure

The project is methodically divided into two main sections:

1. **Analysis of Stock Data**: We have tried to combine advanced cloud technologies with financial analysis, providing a comprehensive and insightful look into the stock performance of some of the world's leading tech giants, especially during periods of economic uncertainty and recession. This section thoroughly examines the historical stock data of 'FANGMANT' companies. By leveraging a combination of AWS services such as S3, Glue, Athena, and QuickSight, we aim to dissect and understand these companies' market behaviors and financial patterns. The focus is on generating statistical insights and interpreting the economic implications during various market phases, including periods of economic uncertainty and recessions.
2. **Real-Time Streaming**: The second section pivots to the technical implementation and challenges of real-time data streaming. Utilizing AWS's potent tools like Kafka, S3, and EC2, we aim to simulate a real-time stock market environment. This involves streaming live data, managing it efficiently in a cloud environment, and analyzing it in real-time. This part of the project underscores AWS's technological prowess and adaptability in handling dynamic, high-volume data streams typical in financial markets.

Section 1.2: Objectives and Methodology

Our methodology integrates the Yahoo Finance API for extracting relevant financial data and capitalizes on AWS's advanced data streaming and analysis capabilities. This dual-faceted approach provides a granular view of stock market dynamics and showcases the transformative power of cutting-edge cloud technology in financial analysis.

The findings from both sections are expected to provide invaluable insights for investors, analysts, and technology enthusiasts, painting a comprehensive picture of the financial landscape of these leading tech companies. Beyond uncovering financial market patterns, this project illustrates how real-time analytics and sophisticated data processing can revolutionize decision-making in the fast-paced interplay between finance and technology.

In essence, "Real-Time Data Streaming and Stock Data Analysis using AWS" is more than a study; it explores the future of financial data analysis, where the confluence of cloud-based solutions and financial acumen opens new frontiers in the digital financial world.

Section 2: Data Exploration

Section 2.1: Introduction

Our data exploration process in the project "Real-Time Data Streaming and Stock Data Analysis using AWS" meticulously navigates through various stages, utilizing a suite of AWS services to analyze market trends during recession periods.

Section 2.2: Data Ingestion and Storage

The journey begins with data retrieval using a Python script, which fetches stock data from Yfinance and aggregates it into a consolidated CSV file. This file, containing data from companies like FANGMANT, is then uploaded to Amazon S3, which is known for its secure and scalable storage solutions.

Section 2.3: Data Processing and Schema Generation

AWS Glue plays a crucial role in transforming the raw data. It initiates a crawler on the S3 bucket to generate a schema automatically, facilitating a seamless Extract, Transform, Load (ETL) process. The transformed data, structured and cleaned, is then stored in S3.

Section 2.4: SQL Querying and Data Analysis

Amazon Athena steps in for in-depth data analysis, allowing us to execute SQL queries directly on the data stored in S3. This enables a flexible and efficient dataset exploration, which is vital for understanding market dynamics during the specified recession periods.

Section 2.5: Visualization and Insight Generation

The final stage of our data exploration involves Amazon QuickSight. It transforms our data into interactive dashboards, visually interpreting market trends in recession periods. QuickSight's SPICE engine supports extensive datasets and provides advanced analytical capabilities like forecasting and natural language querying.

Section 2.6: Conclusion

The integrated AWS services create a comprehensive pipeline from data ingestion to visualization. This end-to-end approach ensures efficient data processing and empowers us with actionable insights, crucial for decision-making in the volatile financial market.

Section 3: Technologies Used

Section 3.1: Services used in Structure 1 (HISTORICAL DATA TO SHOWCASE RECESSION INSIGHTS)

1. **Amazon S3:** As the primary data storage solution, S3 hosted our raw and processed stock data, ensuring secure and scalable storage. AWS S3 is a scalable object storage solution provided by Amazon Web Services. It offers an architecture that is dependable, safe, and highly accessible for storing and retrieving any volume of data from any online location. S3 is a popular storage tier for serverless applications, big data analytics, data lakes, backup and recovery, and data archiving.

2. **AWS Glue:** This service automated the ETL process, crawling data in S3, dynamically generating schemas, and transforming data for analysis. We easily prepared and loaded our data for analytics using AWS Glue, a fully managed extract, transform, and load (ETL) service. Its serverless data integration environment allows users to find, prepare, and integrate data from several sources for analytics, machine learning, and application development.

3. **AWS Glue Crawler:** Glue crawlers are a component of AWS Glue that searches different data stores and extracts the schema from your data. They make the data easily accessible for ETL tasks and query services like Amazon Athena by automatically identifying data types, inferring schemas, and adding metadata to the Glue Data Catalog.

4. **Amazon Athena:** Athena was crucial for querying data using SQL directly in S3, enabling efficient data analysis. Using standard SQL, Amazon Athena is an interactive query tool that facilitates data analysis directly in Amazon S3. Because Athena is serverless, you only pay for the queries you execute and do not need to set up or maintain any infrastructure. It is frequently utilized in intricate data processing pipelines, ad hoc analysis, and data reporting.

5. **Amazon QuickSight:** Employed for data visualization, QuickSight turned complex data sets into interactive dashboards, offering deep insights with its SPICE engine and ML capabilities. This cloud-based business intelligence tool simplifies creating visuals, doing ad hoc analysis, and deriving fast business insights from your data. Without installing or maintaining any software, it may grow automatically to tens of thousands of users.

6. **Python and Boto3:** A Python script with Boto3 library interfaced with AWS services for data retrieval and processing. Boto3 is the Python Software Development Kit (SDK) for Amazon Web Services (AWS). It enables programmers working with Python to create applications that utilize AWS S3, AWS EC2, AWS DynamoDB, and more services. Developers can design, set up, and control AWS services with Boto3.

This blend of technologies formed a robust framework for our data analysis pipeline, addressing various aspects of data handling, processing, and visualization.

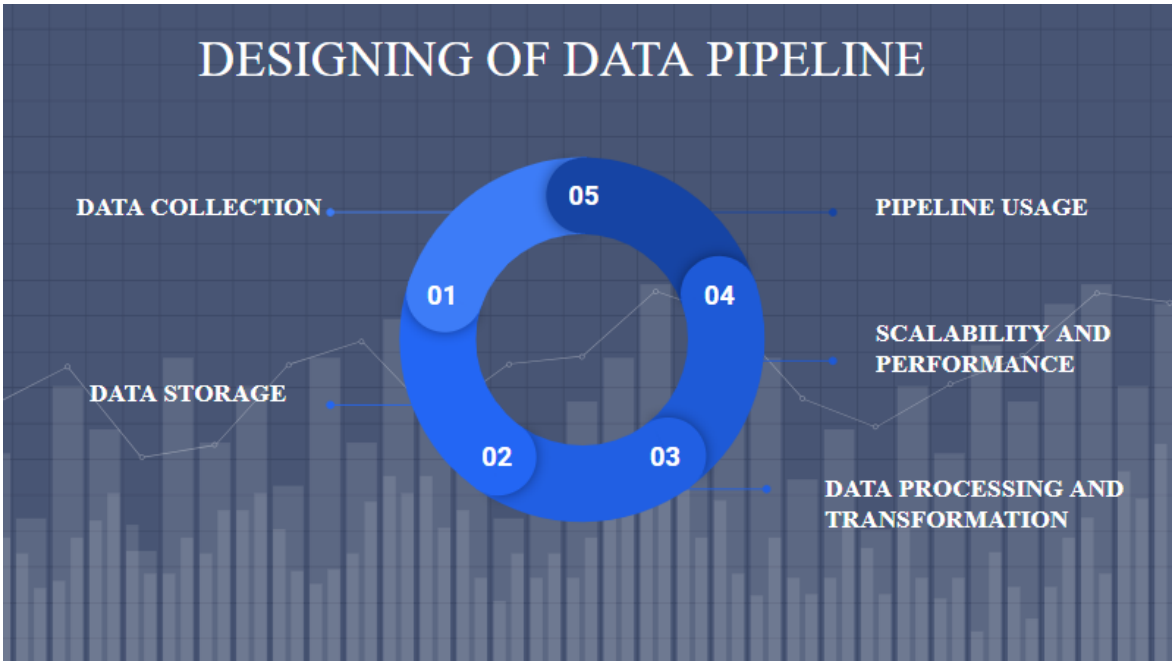
Section 3.2: Additional Services used in Structure 2 (REAL-TIME DATA STREAMING PIPELINE)

7. **Amazon EC2 (Elastic Compute Cloud)**: Elastic Compute Cloud, or AWS EC2, offers expandable processing power within the Amazon Web Services cloud. It speeds up the process of obtaining and starting up new server instances, enabling you to swiftly scale capacity up and down in response to changes in your computing needs.

8. **AWS's Apache Kafka:** Apache Kafka may be used to create streaming applications and real-time data pipelines. Running and maintaining Apache Kafka clusters on AWS infrastructure is called "Kafka on AWS." It may be used to create distributed, high-throughput, publish-subscribe messaging systems. Stream processing, log aggregation, and real-time analytics are some of its common uses in combination with other AWS services.

Section 4: Methodology

Section 4.1: Designing Data Pipeline



A data pipeline is a systematic set of processes and technologies designed to collect, process, and store data efficiently. It facilitates seamless and organized data flow from various sources to its final destination. In the context of our project, key considerations in designing this data pipeline include:

1. Data Collection:

- Leveraging Yahoo Finance's API, the pipeline initiates the collection process, efficiently gathering a substantial volume of stock data. The API serves as a reliable source, ensuring up-to-date and accurate information.

2. Data Storage:

- The pipeline utilizes AWS's serverless options for secure and scalable data storage. This ensures that data is stored in a manner that can readily adapt to changing volumes and demands while benefiting from the reliability and durability of AWS infrastructure.

3. Data Processing and Transformation:

- AWS services are harnessed within the pipeline to convert raw and unprocessed data into meaningful and comprehensible representations. This transformative step is crucial for preparing the data for further analysis, ensuring it is structured, cleaned, and ready for insights.

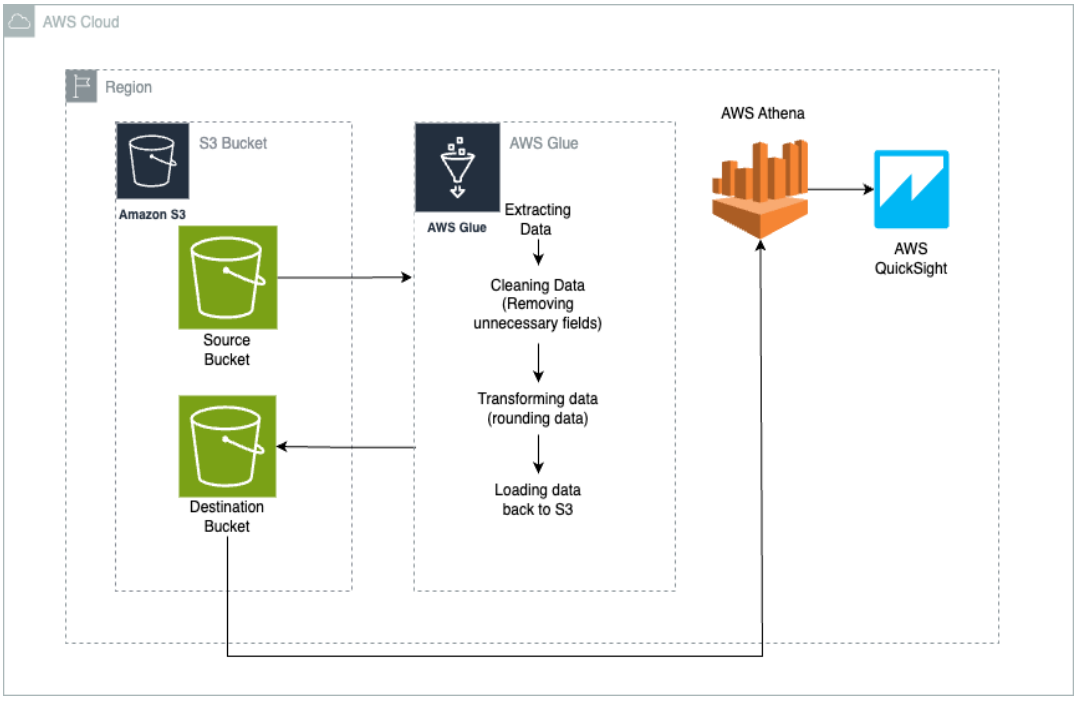
4. Scalability and Performance:

- Ensuring scalability and optimal performance is a core consideration. The pipeline is designed to efficiently handle massive amounts of data, reflecting the dynamic nature of financial markets and the need for timely analysis.

5. Pipeline Usage:

- The design prioritizes user accessibility, providing an easy-to-use interface for accessing and updating the pipeline. This user-friendly approach ensures ongoing, real-time analysis is not only feasible but also streamlined for users interacting with the data pipeline.

Section 4.2: Data Pipeline for Structure 1 (HISTORICAL DATA TO SHOWCASE RECESSION INSIGHTS)



➔ Data Storage in S3:

Our journey starts with Amazon S3, securely storing scraped data as the foundation for a scalable and accessible data processing workflow.

➔ Automated Data Ingestion with AWS Glue:

AWS Glue automates heavy lifting by crawling S3, dynamically forming a schema, and executing ETL operations seamlessly—streamlining the crucial Data Ingestion step.

➔ Efficient SQL Analysis with Athena:

Amazon Athena takes the stage as our SQL query powerhouse, enabling direct analysis of data in S3 using standard SQL for flexible and efficient exploration.

➔ Visual Insights in QuickSight:

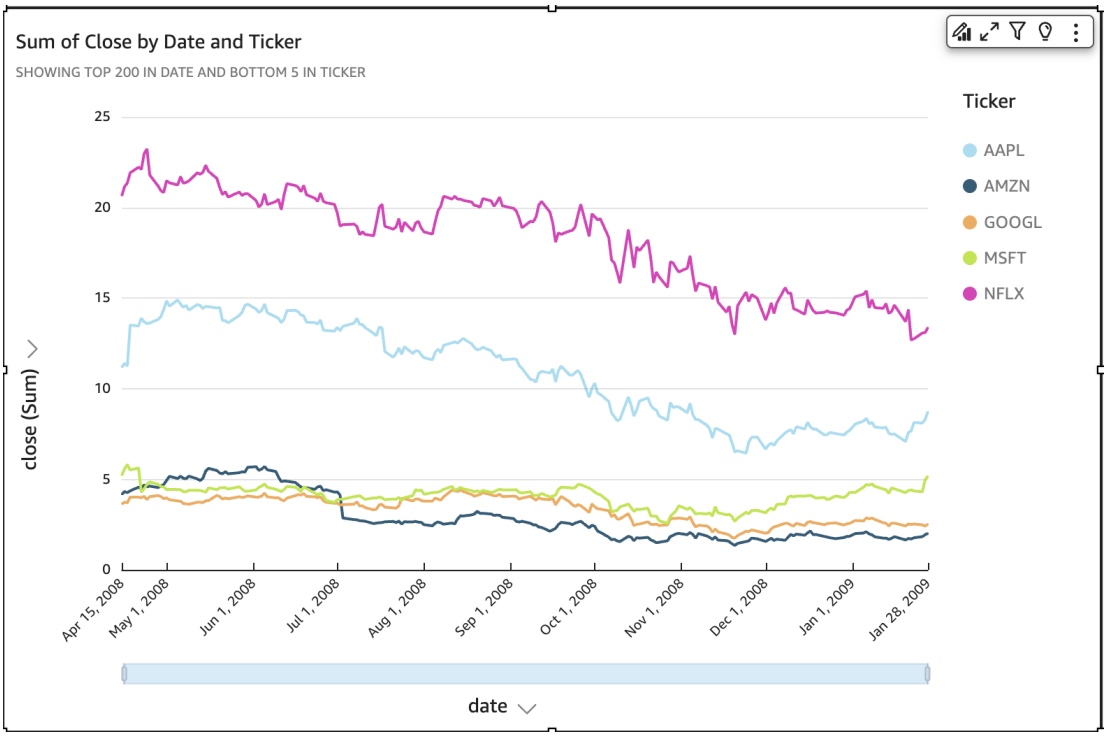
Concluding our journey in Amazon QuickSight, data comes to life through interactive analytics dashboards, transforming insights into a vivid visual representation of our data landscape.

Significance:

1. **Serverless:** All the services used are Serverless, so we don’t need to worry about managing the infrastructure. It can automatically scale to tens of thousands of users without any infrastructure to manage or capacity to plan for. You no longer need to apply software patches or upgrade hardware to meet increasing demand.
2. **Efficiency:** Automation through Glue reduces manual effort in data processing.
3. **Flexibility:** Athena's SQL querying allows us to adapt to changing analysis needs.
4. **Visualization:** QuickSight brings our data to life, facilitating easy interpretation and decision-making.

Section 4.3: QuickSight Visualizations

I. Recession Period-1 Visualization



Integration of AWS Athena and QuickSight:

- Quick integration of AWS Athena into QuickSight swiftly enhances our data visualization capabilities. It's crucial to note that Athena, being serverless, eliminates the need for infrastructure management, providing read-only access.

Focus on Recession Period 1 - March 2008 to Jan 2009:

- The graph illustrates the stock prices during Recession Period 1, specifically from March 2008 to January 2009. To optimize visualization, we concentrate on a subset from April 15, 2008, to Jan 28, 2009, given the data limitations for this period.

Key Insights from the Graph:

The graph reveals significant stock price drops across various companies during this recession. For instance, Netflix's valuation dropped from $23 in April 2008 to $13 by the end of January 2009, reflecting a consistent pattern across multiple companies.

Anomaly Detection - Oct 7, 2008:

- An intriguing finding emerges from anomaly insights, detecting an anomaly on Oct 7, 2008. Notably, the daily total close for Microsoft was lower than expected at 3.19 and Google at 2.93, signifying a deviation from the anticipated values.

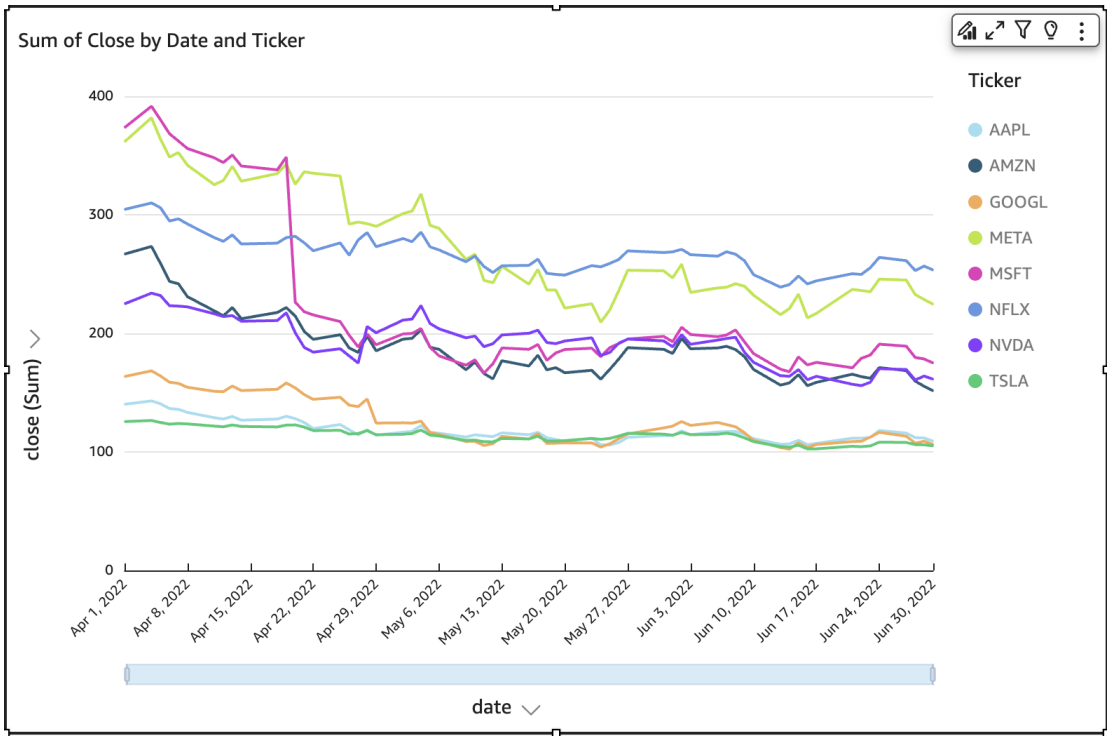
Top total close movers for Jan 28, 2009 are:
AAPL **increased by 5.18% (0.43)**, from 8.3 to 8.73.
GOOGL **increased by 4.13% (0.1)**, from 2.42 to 2.52.
AMZN **increased by 4.12% (0.08)**, from 1.94 to 2.02.

Anomaly Insight

-15.61% ↓

An anomaly was detected on Oct 7, 2008 primarily driven by **lower than expected daily total close for MSFT at 3.19**, and **GOOGL at 2.93**.

II. Recession Period-2 Visualization:



Analyzing the 2022 Recession Period:

Examining the graph below, we delve into the 2022 recession period. Notably, all companies have an initial uptick in stock values in the first week of April. However, a significant downturn follows, leading to a substantial decline in stock values across the board until June 30, 2022.

Anomaly Insights:

The anomaly insights highlight specific decreases in stock values:

- Google experiences a decrease of 2.49%.
- Amazon sees a decline of 2.47%.
- Apple's stock value drops by 2.45%.

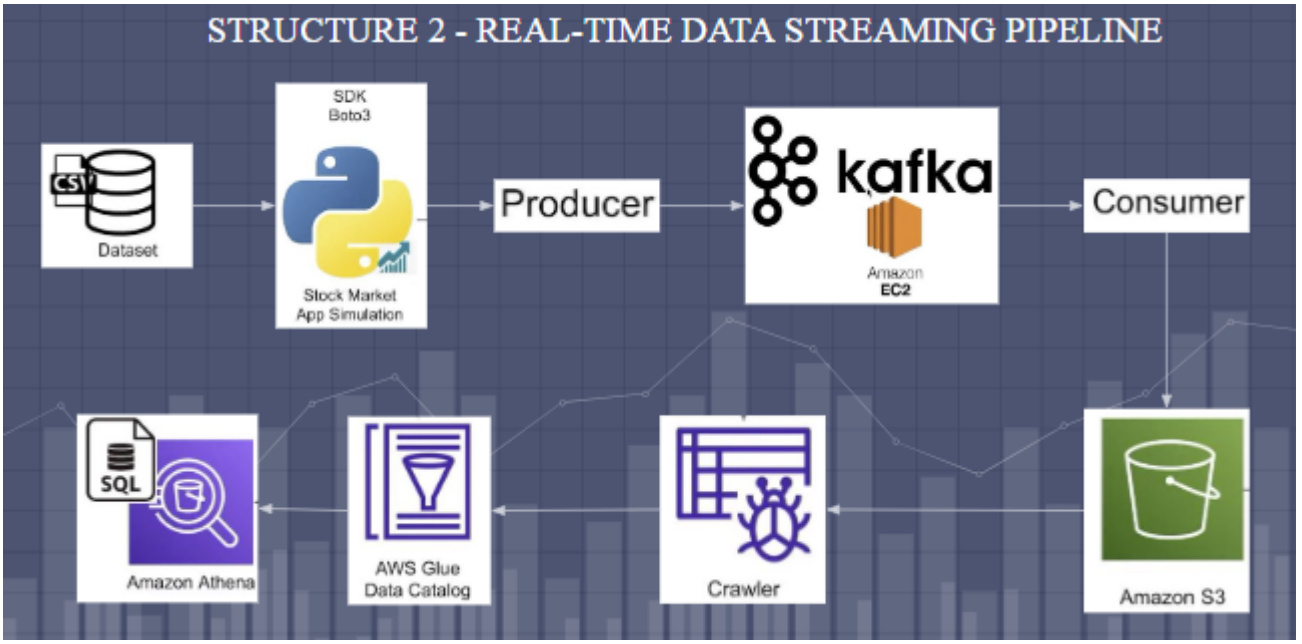
Bottom total close movers for Jun 30, 2022 are:
GOOGL **decreased by 2.49% (2.71)**, from 108.92 to 106.21.
AMZN **decreased by 2.47% (3.83)**, from 155.27 to 151.44.
AAPL **decreased by 2.45% (2.74)**, from 111.7 to 108.96.

Total Close Analysis for June:

The data reveals a noteworthy decrease in the total close for June, reflecting an overall downturn in the market. Specifically, the total close has decreased by 1.81%.

Total close for Jun 30, 2022 **decreased by 1.81% (-23.71)** from 1,309.31 to 1,285.6.

Section 4.4: Data Pipeline for Structure 2 (REAL-TIME DATA STREAMING PIPELINE)



Our goal for this project phase was to simulate real-time stock market data realistically. This was accomplished using a Python-based simulation to stream rows of our previously collected dataset sequentially. After that, the simulated data was easily included in an AWS EC2 instance-hosted Kafka streaming infrastructure. The ability to replicate the ever-changing and uninterrupted character of stock market data flows was crucial for the latter phases of our data processing and analytics pipeline.

➔ Data preparation:

To meet the needs of a real-time streaming situation, we pre-processed our dataset, which represents stock market transactions. Every entry in the dataset represents a distinct transaction on the stock market, complete with the stock ticker, transaction volume, price, and date.

➔ Python-Based Real-Time Simulation:

We created a Python script to mimic real-time data flow. The script mimics the behavior of real-time stock market data feeds by gradually reading rows from the dataset and emitting them at regular intervals. This methodology allowed us to test and improve our data processing techniques in a realistic yet controlled environment.

➔ Kafka Producer Integration:

A Python environment was used to create a Kafka Producer. Its job was to post the stock market simulation data on a Kafka topic. With built-in safeguards against potential network problems or data transmission faults, the Kafka Producer guarantees the rapid and dependable transfer of data messages.

➔ Hosting on EC2:

Our Producer and the entire Kafka ecosystem were housed on an AWS EC2 machine. EC2 offers a safe and scalable cloud computing environment. We ensured our Kafka solution was reliable and able to handle the high-throughput needs characteristic of stock market data by utilizing EC2. The option to scale up the processing capabilities as needed, based on the amount and velocity of incoming data, was another benefit of using AWS EC2.

➔ Data Streaming to Kafka:

The Python script started streaming data to the Kafka topic housed on the EC2 instance as soon as it was executed. This procedure produced a data flow that was almost constant and resembled the real-time stock market feed. The Kafka platform, renowned for its exceptional efficiency in managing real-time data streams, effectively handled the incoming data and prepared it for use by analytical tools and downstream applications.

➔ Uploading to S3:

After that, every data record in JSON format was transferred to an AWS S3 bucket. The object storage service AWS S3 (Simple Storage Service) provides performance, security, scalability, and data availability that are among the best in the business. This indicates that our stock market data, which is now stored in S3, is easily accessible and preserved for any future processing or analytical needs.

➔ AWS Glue for Data Cataloging:

The data kept in S3 was cataloged using Glue. Glue builds a metadata catalog, recognizes the data format, and automatically scans the S3 bucket. The data is easily discoverable and queryable, thanks to this catalog. Additionally, it makes managing the data structure easier, particularly when handling big data sets.

➔ Integration with AWS Athena:

Then, SQL queries were run directly on the S3 data using Athena, an interactive query tool. Because of Athena's interface with AWS Glue's data catalog, we could execute queries on our dataset rapidly and effectively without further data extraction or transformation procedures.

Significance

This setup was crucial for several reasons:

- **Realism in Simulation**: It offered an almost real-world setting for testing analytics and data processing techniques, which was essential for a financial data analysis project.
- **Seamless Data Flow and Storage:** Managing real-time data successfully requires a smooth data flow pipeline, represented by the change from streaming data to storing it in S3 and then querying it.
- **Scalability and Flexibility**: The system was made flexible and scalable to handle changing data volumes without sacrificing performance using AWS EC2 and Kafka.
- **Foundation for Analytics**: Our real-time streaming infrastructure can allow us to extract valuable insights from dynamic stock market data and carry out intricate calculations.

Section 5: Conclusion

- Our AWS-powered data processing pipeline integrates storage, curation, analysis, and visualization seamlessly.
- This end-to-end solution optimizes our data workflow, empowering us to derive meaningful insights from raw data.
- An essential step in our project was the integration of Kafka on AWS EC2 with a real-time data simulation written in Python.
- This setup, combining S3, Glue, and Athena, provided a serverless architecture that is highly scalable and cost-effective.
- It allowed us to focus on analyzing data without worrying about the underlying infrastructure management.
- It established the foundation for modern data processing and analytics to simulate an environment closely reflecting the high-stakes stock market trading world.