

Real-Time Data Streaming and Stock Data Analysis using AWS



GROUP NO 6:
JASH SHAH
PARTH KHARKAR
PARTH BORKAR

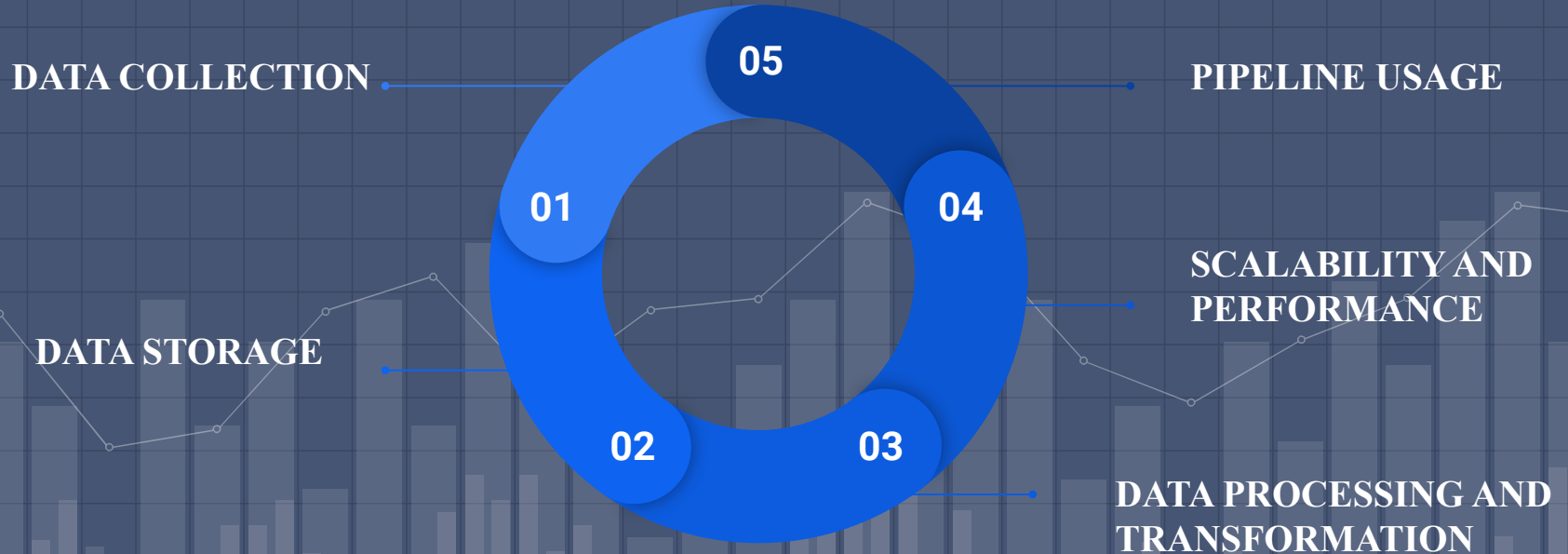
OVERVIEW

- ❑ INTRODUCTION
- ❑ DESIGNING OF DATA PIPELINE
- ❑ YAHOO FINANCE DATA
- ❑ STRUCTURE OF THE PROJECT
- ❑ AWS SERVICES
- ❑ STRUCTURE 1 - MODEL PIPELINE
- ❑ STRUCTURE 2 - MODEL PIPELINE

INTRODUCTION

- Utilizing AWS serverless capabilities to enhance and optimize the task of real-time data streaming as well as analyzing the effects of economic recession on the tech giants
- With the assistance of yahoo finances API (web-scraping) we obtained stock data to construct an ETL pipeline for analyzing it.
- The project focuses on 'FANGMANT' companies- Facebook, Amazon, Netflix, Google, Microsoft, Apple, Nvidia, and Tesla.

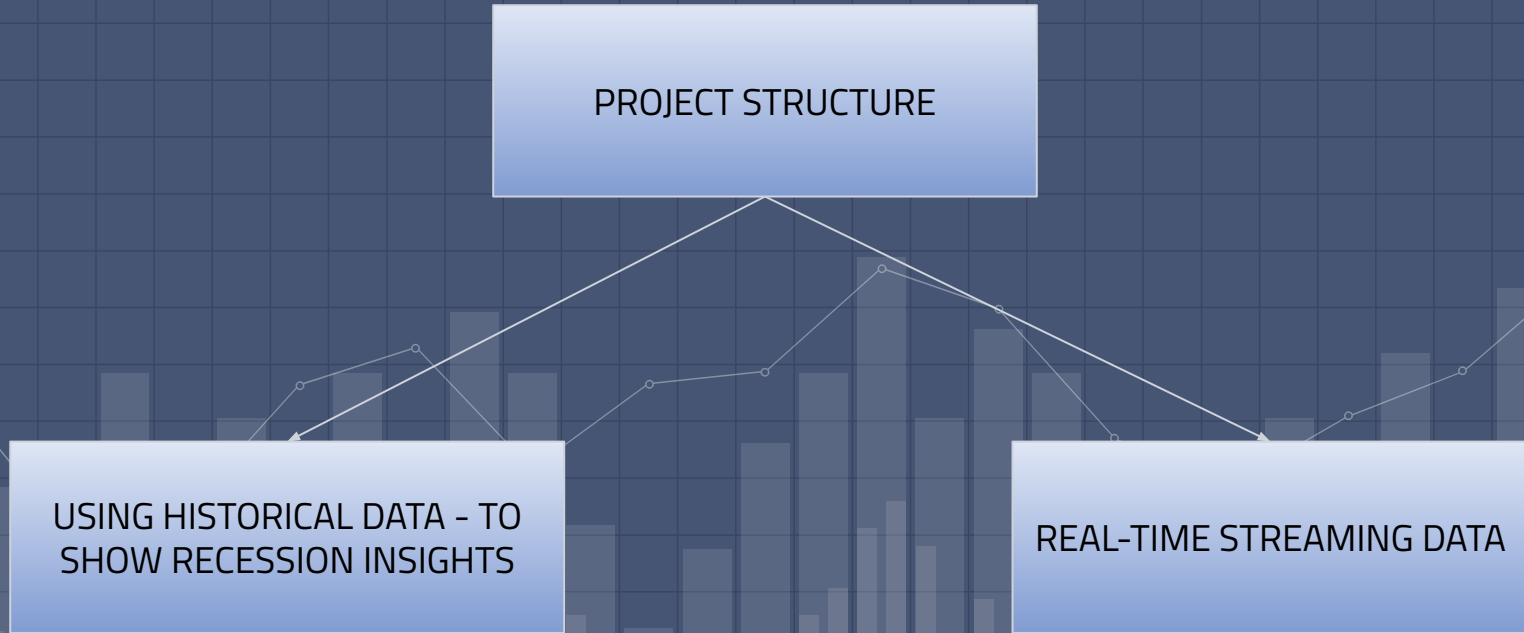
DESIGNING OF DATA PIPELINE



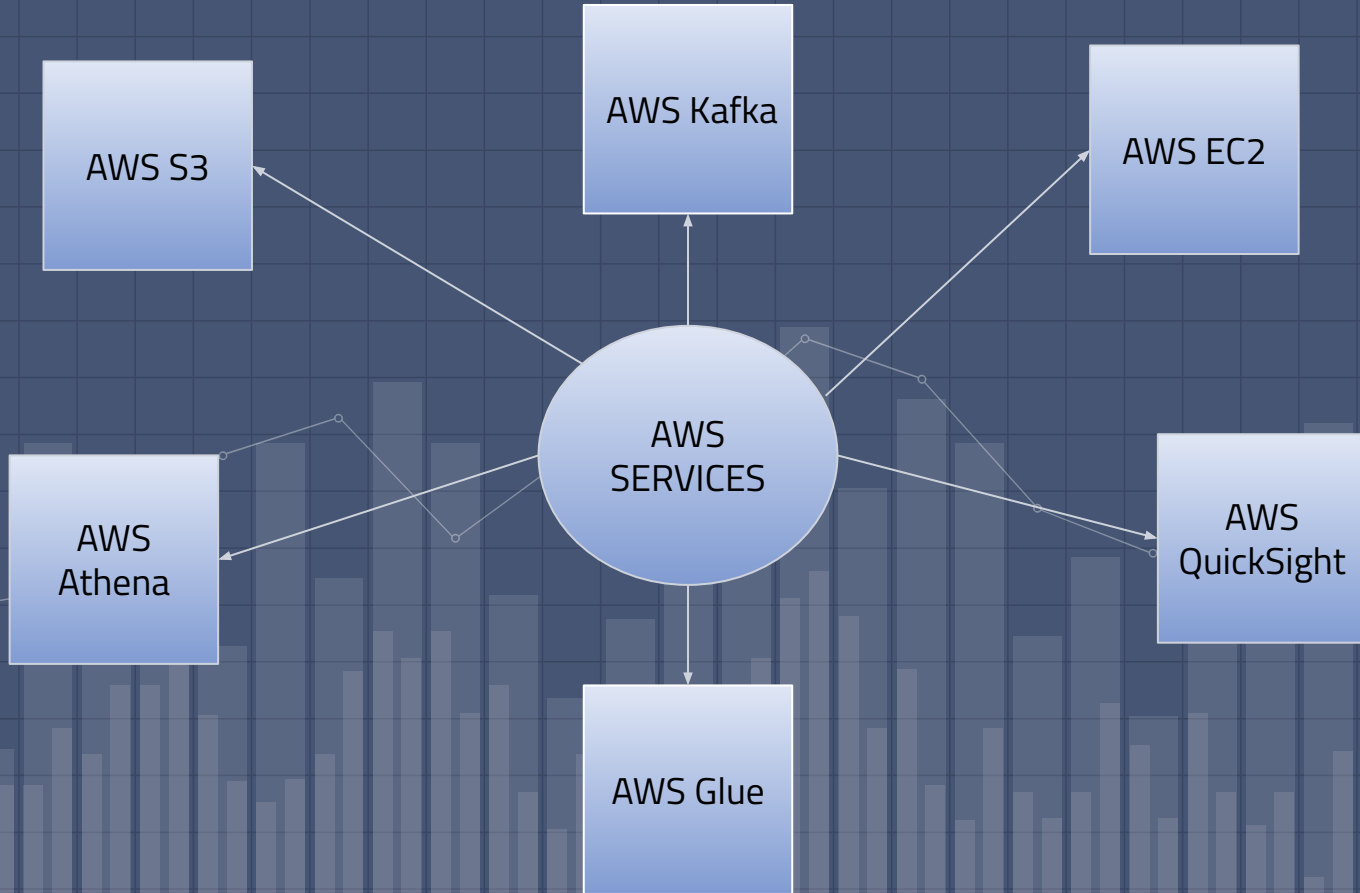
YAHOO FINANCE DATA

- **HISTORICAL SPAN:** A comprehensive timeline from June 1, 1997, to December 12, 2023, allowing us to analyze long-term trends and impacts.
- **TECH GIANTS:** The dataset includes the 'FANGMANT' group - an acronym representing eight of the most influential tech companies in the stock market
- **COLUMN PARAMETERS:** Date, Ticker, Open, High, Low, Close, Adj_Close, Volume
- **SIGNIFICANCE OF DATA:** Our real-time data streaming pipeline is built on this dataset, making it possible to analyze stock performance as well as using it across various economic conditions such as recession period.

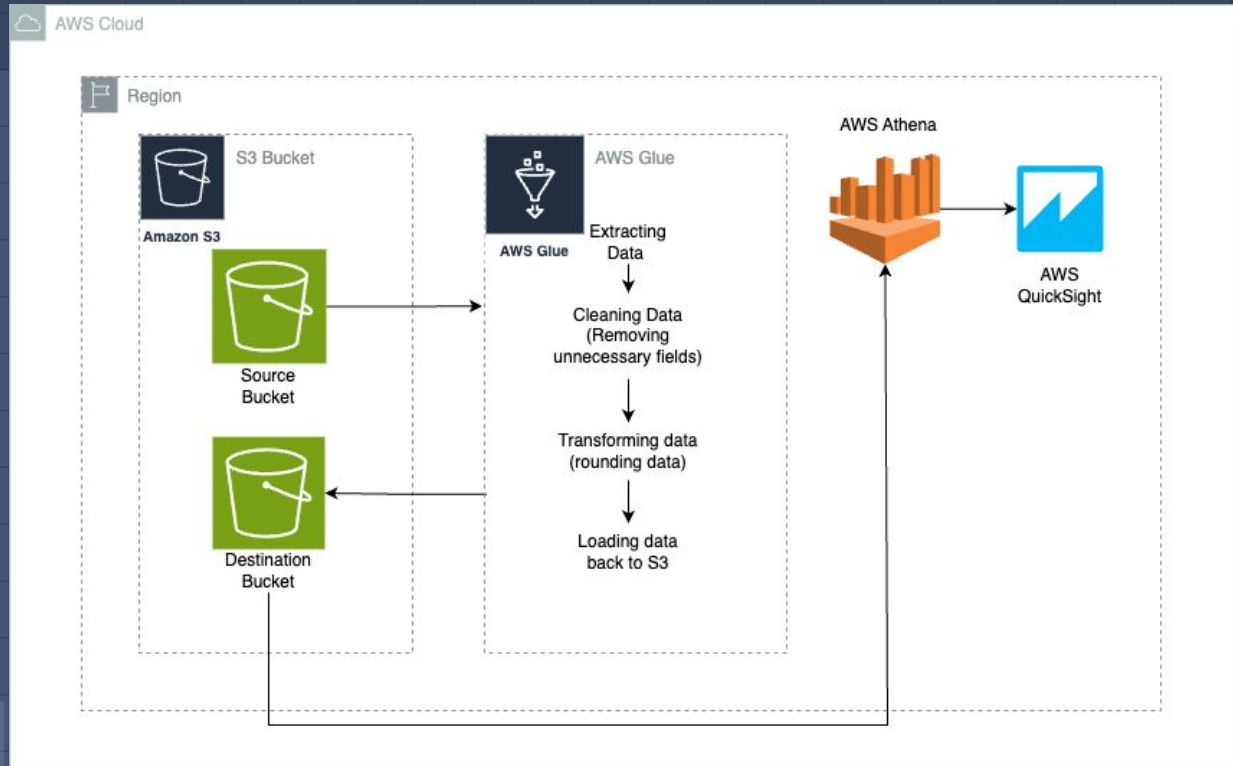
STRUCTURE OF THE PROJECT



AWS SERVICES



STRUCTURE 1 - HISTORICAL DATA TO SHOWCASE RECESSION INSIGHTS



FLOWCHART OF THE DATA PIPELINE

combined_df_data.csv

```

1  ticker,Date,Open,High,Low,Close,Volume,Dividends,Stock Splits
2  AAPL,2004-08-19,2.502502918243408,2.6041040420532227,2.4014010429382324,2.5110108852386475,893181924,0.0,0.0
3  AAPL,2004-08-20,2.527777910232544,2.7297298908233643,2.515014886856079,2.7104599475860596,456686856,0.0,0.0
4  AAPL,2004-08-23,2.771522045135498,2.8398399353027344,2.7289791107177734,2.7377378940582275,365122512,0.0,0.0
5  AAPL,2004-08-24,2.7837839126586914,2.792793035507202,2.59184193611145,2.6243739128112793,304946748,0.0,0.0
6  AAPL,2004-08-25,2.626626968383789,2.7027029991149902,2.599600076675415,2.6526529788970947,183772044,0.0,0.0
7  AAPL,2004-08-26,2.626375913619995,2.701451063156128,2.6191189289093018,2.7004499435424805,141897960,0.0,0.0
8  AAPL,2004-08-27,2.705204963684082,2.7182180881500244,2.644895076751709,2.6564059257507324,124235640,0.0,0.0
9  AAPL,2004-08-30,2.6346349716186523,2.63988995552063,2.5528030395507812,2.5528030395507812,103935960,0.0,0.0
10 AAPL,2004-08-31,2.5600600242614746,2.5953450202941895,2.5565569400787354,2.561811923980713,98357544,0.0,0.0
11 AAPL,2004-09-01,2.5700700283050537,2.576827049255371,2.49424409866333,2.508759021759033,182765052,0.0,0.0
12 AAPL,2004-09-02,2.482232093811035,2.561811923980713,2.47597599029541,2.540290117263794,302373324,0.0,0.0
13 AAPL,2004-09-03,2.526276111602783,2.546046018600464,2.485485076904297,2.5027530193328857,103048848,0.0,0.0
14 AAPL,2004-09-07,2.527777910232544,2.5525529384613037,2.4927430152893066,2.5420420169830322,116950932,0.0,0.0
15 AAPL,2004-09-08,2.5210208892822266,2.5783278942108154,2.515014886856079,2.5600600242614746,99712128,0.0,0.0
16 AAPL,2004-09-09,2.5658159255981445,2.570319890975952,2.5275280475616455,2.560309886932373,81234684,0.0,0.0
17 AAPL,2004-09-10,2.5425429344177246,2.6666669845581055,2.5350348949432373,2.6358859539031982,173977848,0.0,0.0
18 AAPL,2004-09-13,2.6684179306030273,2.712963104248047,2.6641640663146973,2.690190076828003,156882960,0.0,0.0
19 AAPL,2004-09-14,2.688939094543457,2.8028030395507812,2.672421932220459,2.7900400161743164,216579204,0.0,0.0
20 AAPL,2004-09-15,2.7667670249938965,2.8586089611053467,2.7577579021453857,2.8028030395507812,214261524,0.0,0.0
21 AAPL,2004-09-16,2.8113110065460205,2.897897958755493,2.794044017791748,2.852102041244507,185326488,0.0,0.0
22 AAPL,2004-09-17,2.863363027572632,2.940190076828003,2.8415920734405518,2.940190076828003,189450360,0.0,0.0
23 AAPL,2004-09-20,2.9266769886016846,3.0430428981781006,2.9221720695495605,2.9869871139526367,212575212,0.0,0.0
24 AAPL,2004-09-21,2.9982481002807617,3.0135140419006348,2.9406909942626953,2.948949098587036,144575280,0.0,0.0
25 AAPL,2004-09-22,2.9379379749298096,2.9947450160980225,2.923172950744629,2.9624619483947754,151624224,0.0,0.0
26 AAPL,2004-09-23,2.9739739894866943,3.068819046020508,2.9284279346466064,3.023524045944214,170713116,0.0,0.0
27 AAPL,2004-09-24,3.026526927947998,3.1056060791015625,2.996997117996216,2.998749017715454,182469348,0.0,0.0
28 AAPL,2004-09-27,3.0010010067651267,3.025024800800658,2.9470470780723887,3.050450066300001,141222526,0.0,0.0

```

RAW YAHOO FINANCE DATA FOR FANGMANT COMPANIES

AWS SERVICES USED TO PERFORM RECESSION INSIGHTS

- S3 - store scraped data
- Glue - crawl the data in S3 and form a schema and also to perform ETL and completing the Data Ingestion step
- Athena - analyze data directly in Amazon S3 using standard SQL and connect it to Amazon QuickSight
- QuickSight - analytics dashboard

AWS SIMPLE STORAGE SERVICE (AWS S3)

- Provides object storage through a web service interface
- We store the raw data from yFinance by writing a script in Python using Boto3 library
- Before sending it to S3, the stock data from FANGMANT is combined into one csv file.
- We use S3 to store data after performing ETL

AWS S3 BUCKETS

12

General purpose buckets (8) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

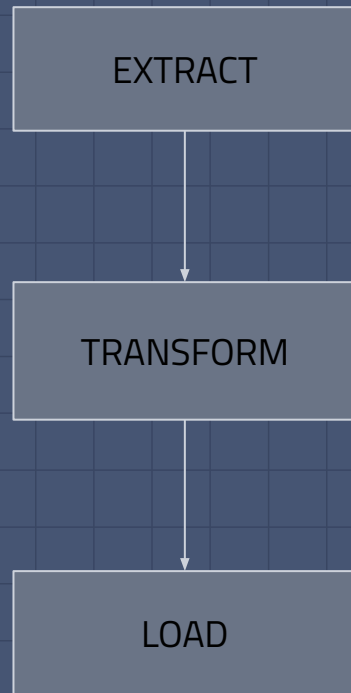
	Name ▲	AWS Region ▼	Access ▼	Creation date ▼
<input type="radio"/>	aws-athena-query-results-us-west-2-403325901382	US West (Oregon) us-west-2	Bucket and objects not public	December 17, 2023, 15:54:04 (UTC-05:00)
<input type="radio"/>	aws-glue-assets-403325901382-us-east-1	US East (N. Virginia) us-east-1	Bucket and objects not public	December 14, 2023, 16:00:01 (UTC-05:00)
<input type="radio"/>	aws-glue-assets-403325901382-us-west-2	US West (Oregon) us-west-2	Bucket and objects not public	December 15, 2023, 21:30:35 (UTC-05:00)
<input type="radio"/>	parthbucket-123	US East (N. Virginia) us-east-1	Bucket and objects not public	December 14, 2023, 23:04:19 (UTC-05:00)
<input type="radio"/>	parthdestinationbucket	US West (Oregon) us-west-2	Objects can be public	December 15, 2023, 21:25:19 (UTC-05:00)
<input type="radio"/>	parthsourcebucket	US West (Oregon) us-west-2	Bucket and objects not public	December 15, 2023, 21:22:34 (UTC-05:00)
<input type="radio"/>	recession-table-db	US West (Oregon) us-west-2	Public	December 16, 2023, 17:02:31 (UTC-05:00)
<input type="radio"/>	stockrawdatabucket	US East (N. Virginia) us-east-1	Bucket and objects not public	December 7, 2023, 19:59:33 (UTC-05:00)

DESTINATION BUCKET

SOURCE BUCKET

AWS GLUE - ETL PIPELINE

- AWS Glue features fall into three major categories:
 - Discover and organize data
 - Transform, prepare, and clean data for analysis
 - Build and monitor data pipelines
- For our project we made use of the second category extensively
- Initiated a crawler to run on our S3 bucket containing combined_df.csv
- Crawler automatically generates a schema for our data



TABLES IN AWS GLUE AFTER CRAWLING

AWS Glue > Databases > stock-data-recession-db

stock-data-recession-db

Last updated (UTC)
December 18, 2023 at 02:29:07

[Refresh](#) [Edit](#) [Delete](#)

Database properties

Name	Description	Location	Created on (UTC)
stock-data-recession-db	-	-	December 16, 2023 at 21:55:44

Tables (6)

Last updated (UTC)
December 18, 2023 at 02:29:08

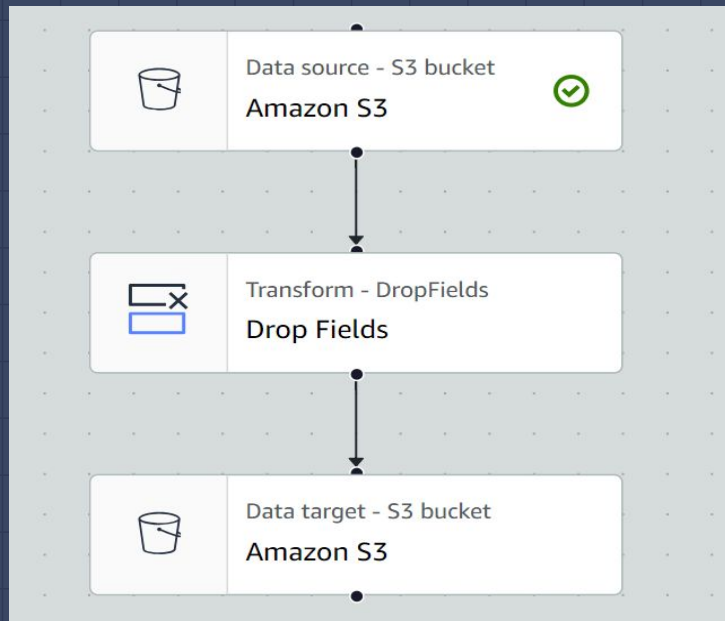
[Refresh](#) [Delete](#) [Add tables using crawler](#) [Add table](#)

View and manage all available tables.

<input type="checkbox"/>	Name	Database	Location	Classification	Deprecated	View data	Data quality
<input type="checkbox"/>	combined_stock_data	stock-data-recession	s3://parthdestination	CSV	-	Table data	View data quality
<input type="checkbox"/>	manifest_json	stock-data-recession	s3://parthdestination	JSON	-	Table data	View data quality
<input type="checkbox"/>	parth_athena	stock-data-recession	s3://parthdestination	CSV	-	Table data	View data quality
<input type="checkbox"/>	parth_second_try	stock-data-recession	s3://parthdestination	CSV	-	Table data	View data quality
<input type="checkbox"/>	run_1702840828167	stock-data-recession	s3://parthdestination	CSV	-	Table data	View data quality
<input type="checkbox"/>	stock_data_final	stock-data-recession	s3://parthdestination	CSV	-	Table data	View data quality

Database in which tables are formed

- We use Glue to perform ETL on the data using ETL jobs
- We use the visual functionality first for basic operations, and then modify the script according to our project
- Processed data is stored in a new destination S3 bucket



AWS ATHENA

1. OVERVIEW:

- Interactive query service for analyzing data in Amazon S3
- Serverless, no infrastructure setup, pay-as-you-go based on queries

2. SCHEMA DEFINITION OPTIONS:

- DDL Statements: Directly define schema using SQL
- AWS Glue: Automatically crawl data sources to discover data and populate your Data Catalog -> *our approach*

3. Serverless architecture hence no infrastructure management

4. Amazon QuickSight Dashboard: Directly access data for visualization

AWS ATHENA CONSOLE

17

The screenshot displays the AWS Athena console interface. At the top, the 'Services' menu and a search bar are visible. The main header shows 'Amazon Athena > Query editor'. Below this, there are tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. The 'Workgroup' is set to 'primary'.

On the left sidebar, under the 'Data' section, the 'Data source' is 'AwsDataCatalog' and the 'Database' is 'stock-data-recession-db'. The 'Tables and views' section shows a list of tables, including 'combined_stock_data_athena', 'manifest_json', 'parth_athena', 'parth_second_try', 'run_1702840828167_part_r_00000', and 'stock_data_final'.

The main query editor area shows a SQL query: `SELECT * FROM "stock-data-recession-db"."combined_stock_data_athena" limit 10;`. The query is labeled 'Query 41'.

Below the query editor, the 'Query results' tab is active, showing the execution status as 'Completed'. The query results are displayed in a table with 1 column and 1 row. The table is titled 'combined_stock_data_athena'.

At the bottom, the query execution details are shown: 'Time in queue: 112 ms', 'Run time: 751 ms', and 'Data scanned: 0.35 KB'.

An arrow points from the text 'QUERY TO PREVIEW TABLE' to the query editor area.

AWS QUICKSIGHT

- We use QuickSight for visualization of our data:
 - Recession periods
 - Real-time stock data streaming
- Support for big datasets: SPICE now supports datasets up to billion row
- **SERVERLESS**: Can automatically scale to tens of thousands of users, without any infrastructure to manage
- **REASONS TO USE AWS QUICKSIGHT**: Deeper data insights through as-needed analysis and machine learning (ML) capabilities such as anomaly detection, forecasting, and natural language querying

BENEFITS OF USING AWS ATHENA

SERVERLESS

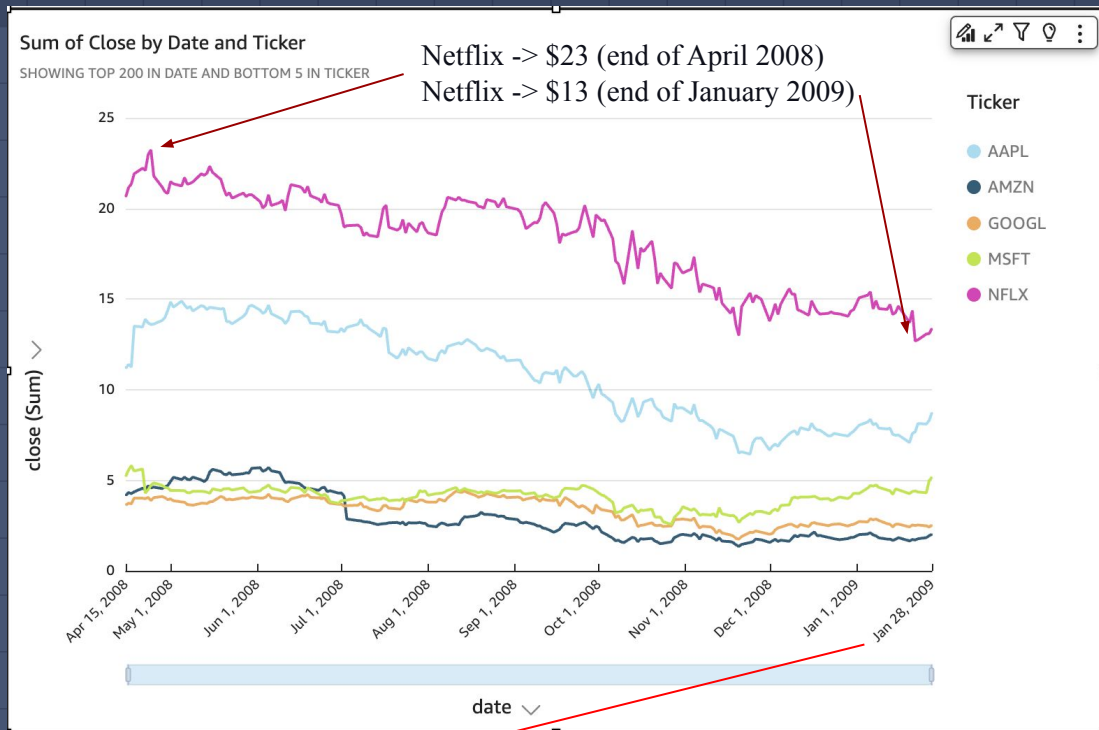
EFFICIENCY

FLEXIBILITY

VISUALIZATIONS

RECESSION PERIOD - 1: VISUALIZATION

20



Total close for Jan 28, 2009 decreases by 2.27%

ANOMALIES:

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.

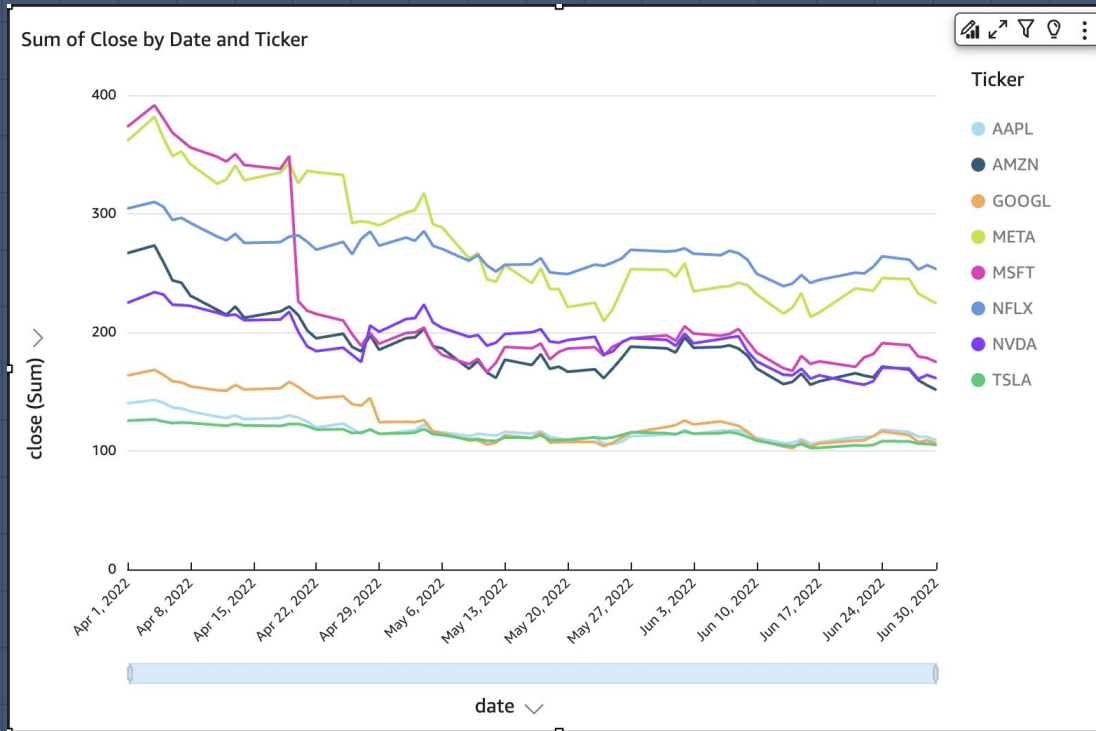
Anomaly Insight

-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.

RECESSION PERIOD-2 VISUALIZATION

21



ANOMALIES:

Bottom total close movers for Jun 30, 2022 are:

GOOGL **decreased by 2.49% (2.71)**, from 108.92 to 106.21.

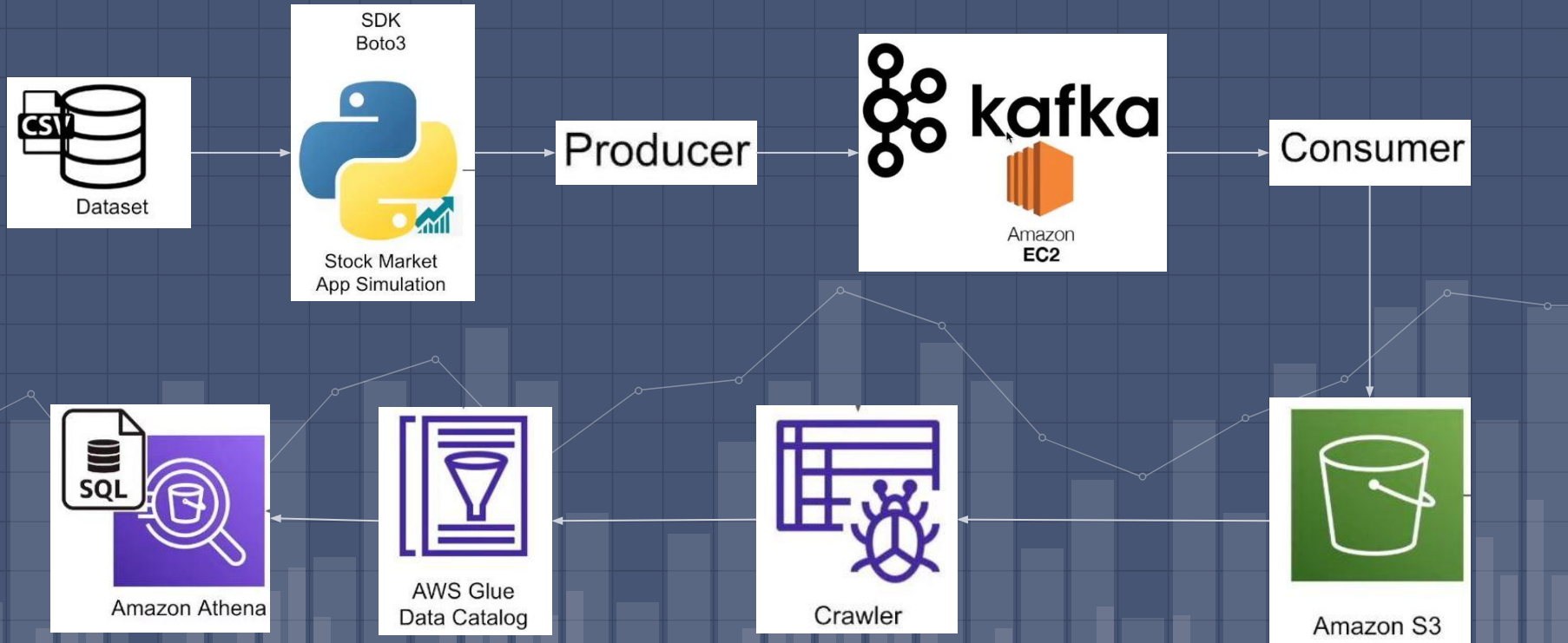
AMZN **decreased by 2.47% (3.83)**, from 155.27 to 151.44.

AAPL **decreased by 2.45% (2.74)**, from 111.7 to 108.96.

Total close for Jun 30, 2022

decreased by 1.81% (-23.71) from 1,309.31 to 1,285.6.

STRUCTURE 2 - REAL-TIME DATA STREAMING PIPELINE



AWS EC2 - Elastic Compute Cloud

- **Elastic Compute Power:** It automatically adjusts computer capacity to match application demands in real-time.
- **Diverse Instance Types:** A wide range of instance types designed to accommodate various workloads, such as GPU-, compute-, or memory-intensive operations in addition to general-purpose jobs.
- **Cost-Effective Pricing:** Pay-as-you-go pricing that takes consumption patterns into account by offering options for Reserved, On-Demand, and Spot Instances.
- **Enhanced Security:** Strong security groups and key pairs are integrated with AWS VPC to provide network isolation and safe instance access.
- **Global Infrastructure:** To increase redundancy and lower latency, deploy applications across several Availability Zones and geographies.

AWS EC2 - Console

24

arch

[Alt+S]

Ohio ▾parthborkar ▾

Instances (1) [Info](#)

↻

Connect

Instance state ▾

Actions ▾

Launch instances ▾

Q

Find Instance by attribute or tag (case-sensitive)

< 1 > ⚙

<input type="checkbox"/>	Name ✎ ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS ▾	Public IPv4
<input type="checkbox"/>	kafka-stock-market-project	i-09fce7d4c2b15e614	⊖ Stopped 🔍 🔍	t2.micro	-	No alarms +	us-east-2a	-	-

Select an instance

⚙ ✕

Apache Kafka

- **High-Throughput Data Pipeline:** Kafka functions as a high-availability system that can process billions of events daily.
- **Distributed Streaming Platform:** Kafka is perfect for large-scale message processing applications since it is built to be a distributed, fault-tolerant, and horizontally scalable system.
- **Real-Time Processing:** Kafka facilitates stream processing, which enables instantaneous data modification as it comes in and allows real-time data streaming into analytics systems.
- **Reliability and Durability:** Kafka uses retention and replication strategies to provide reliability while storing streams of records in topics.
- **Flexible Integrations:** Kafka can be easily integrated with a wide range of databases, storage systems, and live dashboards since it is compatible with many producer and consumer interfaces.

Kafka: Topics, Producers, Consumers

- **Kafka Topics:** Essential to Kafka's communications, topics are feed names or categories to which records are posted. Topics in Amazon Kafka are divided, distributed, and extremely resilient logs.
- **Producers:** Producers are programs or services that post events or messages related to Kafka subjects. AWS Kafka guarantees producers a smooth integration experience so they can connect and stream data effectively.
- **Customers:** Customers peruse information from subjects to which they have subscriptions. With Amazon Kafka, users can efficiently manage streams using consumer groups for maximum throughput while scaling horizontally.
- **Consoles for Producers and Consumers:** AWS offers intuitive user interfaces for setting up and controlling producers and consumers of Kafka, making real-time data processing and streaming easier.

```

[ec2-user@ip-172-31-1-10 ~]$ cat /etc/os-release
NAME="Amazon Linux 2"
VERSION="2.0.20230310.1"
ID="amzn"
ID_LIKE="fedora"
BOUT="AL2"
VERSION_ID="20230310"
PRETTY_NAME="Amazon Linux 2"
ANSI_COLOR="0;32"
LOGO="https://amazonlinux.github.io/images/logo/AL2/amazonlinux.png"
CPE_NAME="cpe:2.3:os:amazonlinux:2:0:GA:Amazon Linux 2"

```










Simulating upload of real time data into S3 Bucket


```
ec2-user@ip-172-31-1-12:~/kafka_2.12-3.4.1
05855, "Adj_Close": 0.1158939227461814, "Volume": 111148800}
{"Date": "06/25/1997", "Ticker": "AMZN", "Open": 0.0752599984407424, "High": 0.0763019993901252, "Low": 0.07395800203
08494, "Adj_Close": 0.0755209997296333, "Volume": 15024000}
{"Date": "06/26/1997", "Ticker": "AMZN", "Open": 0.0763019993901252, "High": 0.0763019993901252, "Low": 0.07447899878
02505, "Adj_Close": 0.0755209997296333, "Volume": 42120000}
{"Date": "06/26/1997", "Ticker": "AAPL", "Open": 0.1158939278255825, "High": 0.1163669363219095, "Low": 0.11352889797
53597, "Adj_Close": 0.1144749149680137, "Volume": 198632000}
{"Date": "06/27/1997", "Ticker": "AAPL", "Open": 0.1144749055586689, "High": 0.1144749055586689, "Low": 0.11069000422
59438, "Adj_Close": 0.1111630126833915, "Volume": 381987200}
{"Date": "06/27/1997", "Ticker": "AMZN", "Open": 0.0760419964790344, "High": 0.0760419964790344, "Low": 0.07525999844
07424, "Adj_Close": 0.0755209997296333, "Volume": 63792000}
{"Date": "06/28/1997", "Ticker": "AMZN", "Open": 0.0757810026407241, "High": 0.0757810026407241, "Low": 0.07395800203
08494, "Adj_Close": 0.0744789987802505, "Volume": 23760000}
{"Date": "06/28/1997", "Ticker": "AAPL", "Open": 0.1111630126833915, "High": 0.112109029598287, "Low": 0.110690004225
9438, "Adj_Close": 0.1111630126833915, "Volume": 157953600}
{"Date": "07/01/1997", "Ticker": "AAPL", "Open": 0.1116360323437768, "High": 0.1116360323437768, "Low": 0.10595996817
90313, "Adj_Close": 0.1078519895672798, "Volume": 171180800}
{"Date": "07/01/1997", "Ticker": "AMZN", "Open": 0.0755209997296333, "High": 0.0799480006098747, "Low": 0.07395800203
08494, "Adj_Close": 0.0770829990506172, "Volume": 54936000}
{"Date": "07/02/1997", "Ticker": "AAPL", "Open": 0.1054869780170561, "High": 0.1059599802601951, "Low": 0.09933790464
63008, "Adj_Close": 0.0998109132051467, "Volume": 450676800}
{"Date": "07/02/1997", "Ticker": "AMZN", "Open": 0.0770829990506172, "High": 0.0770829990506172, "Low": 0.07552099972
96333, "Adj_Close": 0.0757810026407241, "Volume": 25848000}
{"Date": "07/03/1997", "Ticker": "AMZN", "Open": 0.0757810026407241, "High": 0.0796879976987838, "Low": 0.07552099972
96333, "Adj_Close": 0.079426996409893, "Volume": 77640000}
{"Date": "07/03/1997", "Ticker": "AAPL", "Open": 0.1002839143697017, "High": 0.1012299251615119, "Low": 0.09839104648
13544, "Adj_Close": 0.0988640487194061, "Volume": 249961600}
{"Date": "07/04/1997", "Ticker": "AAPL", "Open": 0.0993378997736439, "High": 0.1050139643071348, "Low": 0.09839104271
33678, "Adj_Close": 0.1035949513316154, "Volume": 186782400}
```

















Simulating upload of real time data into S3 Bucket

Objects (224) Info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly










  Copy S3 URI  Copy URL  Download  Open  Delete  Actions  Create folder  Upload


 Find objects by prefix
















<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	 stock_market_223.json	json	December 18, 2023, 03:24:05 (UTC-05:00)
<input type="checkbox"/>	 stock_market_222.json	json	December 18, 2023, 03:24:04 (UTC-05:00)
<input type="checkbox"/>	 stock_market_221.json	json	December 18, 2023, 03:24:03 (UTC-05:00)
<input type="checkbox"/>	 stock_market_220.json	json	December 18, 2023, 03:24:02 (UTC-05:00)
<input type="checkbox"/>	 stock_market_219.json	json	December 18, 2023, 03:24:01 (UTC-05:00)
<input type="checkbox"/>	 stock_market_218.json	json	December 18, 2023, 03:24:00 (UTC-05:00)
<input type="checkbox"/>	 stock_market_217.json	json	December 18, 2023, 03:23:59 (UTC-05:00)
<input type="checkbox"/>	 stock_market_216.json	json	December 18, 2023, 03:23:58 (UTC-05:00)
<input type="checkbox"/>	 stock_market_215.json	json	December 18, 2023, 03:23:57 (UTC-05:00)
<input type="checkbox"/>	 stock_market_214.json	json	December 18, 2023, 03:23:56 (UTC-05:00)
<input type="checkbox"/>	 stock_market_213.json	json	December 18, 2023, 03:23:55 (UTC-05:00)
<input type="checkbox"/>	 stock_market_212.json	json	December 18, 2023, 03:23:54 (UTC-05:00)
<input type="checkbox"/>	 stock_market_211.json	json	December 18, 2023, 03:23:53 (UTC-05:00)
<input type="checkbox"/>	 stock_market_210.json	json	December 18, 2023, 03:23:52 (UTC-05:00)
<input type="checkbox"/>	 stock_market_209.json	json	December 18, 2023, 03:23:51 (UTC-05:00)

Objects (294) Info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly

  Copy S3 URI  Copy URL  Download  Open  Delete  Actions  Create folder  Upload

 Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	 stock_market_293.json	json	December 18, 2023, 03:25:16 (UTC-05:00)
<input type="checkbox"/>	 stock_market_292.json	json	December 18, 2023, 03:25:15 (UTC-05:00)
<input type="checkbox"/>	 stock_market_291.json	json	December 18, 2023, 03:25:14 (UTC-05:00)
<input type="checkbox"/>	 stock_market_290.json	json	December 18, 2023, 03:25:13 (UTC-05:00)
<input type="checkbox"/>	 stock_market_289.json	json	December 18, 2023, 03:25:12 (UTC-05:00)
<input type="checkbox"/>	 stock_market_288.json	json	December 18, 2023, 03:25:11 (UTC-05:00)
<input type="checkbox"/>	 stock_market_287.json	json	December 18, 2023, 03:25:10 (UTC-05:00)
<input type="checkbox"/>	 stock_market_286.json	json	December 18, 2023, 03:25:09 (UTC-05:00)
<input type="checkbox"/>	 stock_market_285.json	json	December 18, 2023, 03:25:08 (UTC-05:00)
<input type="checkbox"/>	 stock_market_284.json	json	December 18, 2023, 03:25:07 (UTC-05:00)
<input type="checkbox"/>	 stock_market_283.json	json	December 18, 2023, 03:25:06 (UTC-05:00)
<input type="checkbox"/>	 stock_market_282.json	json	December 18, 2023, 03:25:05 (UTC-05:00)
<input type="checkbox"/>	 stock_market_281.json	json	December 18, 2023, 03:25:04 (UTC-05:00)
<input type="checkbox"/>	 stock_market_280.json	json	December 18, 2023, 03:25:03 (UTC-05:00)
<input type="checkbox"/>	 stock_market_279.json	json	December 18, 2023, 03:25:02 (UTC-05:00)

Movement of data into Athena in real time

Query 2 :

```
1 SELECT count(*) FROM "stock_market_kafka"."kafka_cloud_comp_project" limit 10;
```

SQL Ln 1, Col 79

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☐ Reuse query results up to 60 minutes ago

Query results | Query stats

Completed Time in queue: 66 ms Run time: 595 ms Data scanned: 26.51 KB

Results (1)

[Copy](#) [Download results](#)

#	_col0
1	155

Query 2 :

```
1 SELECT count(*) FROM "stock_market_kafka"."kafka_cloud_comp_project" limit 10;
```

SQL Ln 1, Col 79

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☐ Reuse query results up to 60 minutes ago

Query results | Query stats

Completed Time in queue: 57 ms Run time: 653 ms Data scanned: 33.53 KB

Results (1)

[Copy](#) [Download results](#)

#	_col0
1	196

THANK YOU!