**Report**

**Real time stock market data analysis**

**Introduction**

This report is on real-time stock market data analysis using Apache Kafka. Technologies that are used are Python, AWS(Amazon Web services),Apache Kafka,Glue,Athena and SQL.

The Stock Market Kafka Real-Time Data Engineering Project is an ambitious endeavor aimed at executing an End-To-End Data Engineering Project focused on harnessing Real-Time Stock Market Data through the utilization of Kafka. This report provides an in-depth analysis of the project's architecture, the technologies employed, and the dataset used in the process.

Amazon Web services:

1. S3 (Simple Storage Service)
2. Athena
3. Glue Crawler
4. Glue Catalog
5. EC2

**Architecture:**

The project's architecture is designed to facilitate seamless data flow and processing. The core components include Python for programming, Amazon Web Services (AWS) for cloud infrastructure, S3 for data storage, Athena for querying data in S3, Glue for ETL operations, EC2 for hosting, and Apache Kafka for real-time data streaming.

Workflow:

* Real-time stock market data is ingested into the system through Kafka, ensuring a continuous and reliable stream.
* Python processes the incoming data, performing necessary transformations and cleaning operations.
* Processed data is stored securely in S3, providing a scalable and durable storage solution.
* Glue Crawler automatically discovers and catalogs metadata, enhancing data accessibility and simplifying ETL operations.
* Glue Catalog serves as a centralized repository for metadata, ensuring consistent and accurate data management.
* Athena enables users to interactively query and analyze the real-time stock market data stored in S3, providing valuable insights.

A diagram of a company

Description automatically generated

Mainly focused on operation side of Data Engineering.

Programming Language - Python: Python serves as the primary programming language, offering versatility and ease of integration with various components.

Amazon Web Service (AWS): AWS acts as the backbone, providing a scalable and reliable cloud infrastructure.

S3 (Simple Storage Service): S3 is employed for efficient and durable data storage, ensuring accessibility and durability.

Athena: Athena is utilized for querying data stored in S3, offering a serverless and interactive query service.

Glue Crawler: Glue Crawler automatically discovers and catalogs metadata from diverse data sources, enhancing data accessibility.

Glue Catalog: Glue Catalog is employed for maintaining the metadata repository, facilitating seamless ETL operations.

EC2: EC2 instances are used for hosting and running applications, ensuring flexibility and control.

Apache Kafka: Kafka serves as the central component for real-time data streaming, enabling efficient and reliable data transfer.

**Codes and Commands:**

#To download Apache kafka

wget https://downloads.apache.org/kafka/3.3.1/kafka\_2.12-3.3.1.tgz

tar -xvf kafka\_2.12-3.3.1.tgz

-----------------------

java -version

sudo yum install java-1.8.0-openjdk

java -version

cd kafka\_2.12-3.3.1

Start Zoo-keeper:

-------------------------------

bin/zookeeper-server-start.sh config/zookeeper.properties

Open another window to start kafka

But first ssh to to your ec2 machine as done above

Start Kafka-server:

----------------------------------------

Duplicate the session & enter in a new console --

export KAFKA\_HEAP\_OPTS="-Xmx256M -Xms128M"

cd kafka\_2.12-3.3.1

bin/kafka-server-start.sh config/server.properties

Do a "sudo nano config/server.properties" - change ADVERTISED\_LISTENERS to public ip of the EC2 instance

**For creating topic**

bin/kafka-topics.sh --create --topic demo\_testing2 --bootstrap-server 3.19.28.46:9092

**for creating producer:**

bin/kafka-console-producer.sh --topic demo\_test --bootstrap-server 3.19.28.46:9092

**for creating consumer:**

bin/kafka-console-consumer.sh --topic demo\_test --bootstrap-server 3.19.28.46:9092

Dataset Used:

While the project allows flexibility in the choice of datasets, the emphasis lies on the operational aspects of Data Engineering, particularly the construction of a robust data pipeline. So the simulation of existing data is done.

Project Execution:

The project execution involves the creation of an end-to-end data pipeline that seamlessly integrates Python for data processing, AWS for cloud services, S3 for data storage, and Kafka for real-time data streaming. This ensures a continuous flow of stock market data, enabling timely and informed decision-making.

Data Processing and Storage:

Python's versatility enables efficient data processing, while AWS S3 ensures secure and durable storage. The integration of Glue Crawler and Glue Catalog automates the discovery and cataloging of metadata, simplifying the ETL process and enhancing overall data accessibility.

Real-Time Data Streaming:

Apache Kafka plays a pivotal role in enabling real-time data streaming. Its distributed architecture ensures high throughput and fault tolerance, making it an ideal choice for handling the dynamic nature of stock market data.

AWS Services Integration:

The integration of AWS services such as Athena facilitates seamless querying of data stored in S3. This serverless approach enhances efficiency, allowing users to interactively analyze and derive insights from the real-time stock market data.

Flexibility in Dataset Selection:

The project's flexibility in dataset selection emphasizes the operational aspects of Data Engineering. This approach ensures that the chosen dataset aligns with the project's objectives, fostering the development of a robust and adaptive data pipeline.

**Conclusion:**

this project's workflow seamlessly ingests real-time stock market data through Kafka, undergoes precise processing in Python, securely stores the processed data in S3, automates metadata cataloging with Glue Crawler, maintains a centralized repository of metadata in Glue Catalog, and enables interactive querying and analysis through Athena. This holistic approach harmonizes each component's strengths, resulting in an effective and scalable real-time data engineering solution that meets the dynamic challenges of the stock market data landscape.

References:

* <https://kafka.apache.org/24/documentation.html>
* <https://docs.aws.amazon.com/>