Architecture Proposal for Data Analytics

Harshit Singhai

Packt

May 8, 2023

Introduction & Background

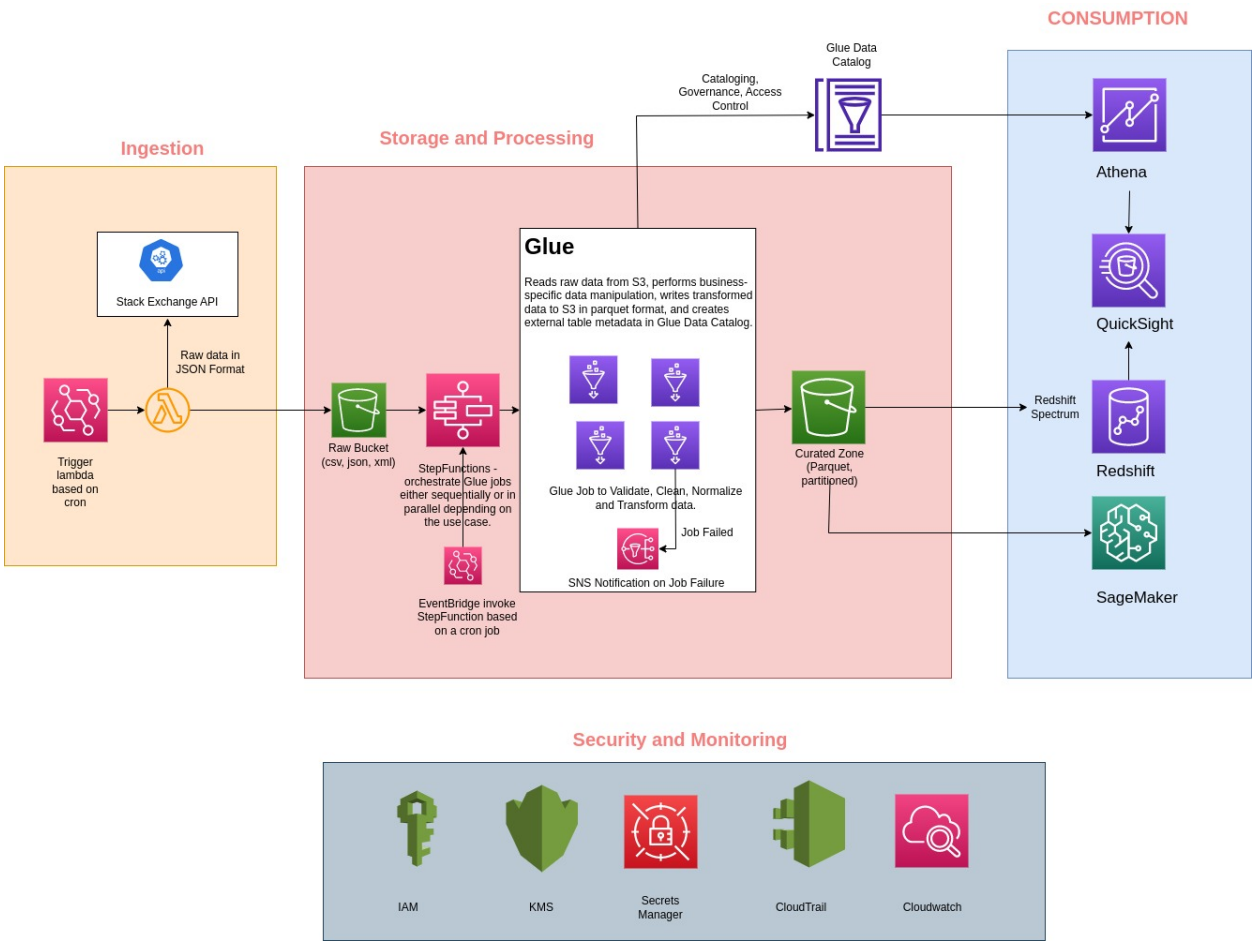
Building an end-to-end data pipeline solution for collecting data from Stack Overflow to be able to understand current sentiment and developments across the tech community. Ingesting it into the system, storing it in an database or data lake, retrieving it when necessary, and generating insights from the data.

We explored three different architecture designs to meet the requirements and constraints of the project. Each design has its own set of advantages and disadvantages, which we analyzed to determine the best approach for the given problem statement.

The first architecture design provides a serverless architecture that is cost-effective and scalable. The second architecture design uses an EMR cluster to process the data.
The third architecture design is an event-driven approach that uses a producer-consumer paradigm.

In conclusion, each architecture design has its own set of advantages and disadvantages, and the best approach will depend on the specific requirements and constraints of the project. By analyzing the different architecture designs, we can choose the best approach that meets the project's needs and delivers the desired outcomes.

Architecture 1 - Serverless Data Analytics



Explanation of Workflow steps

1. Event Bridge triggers Lambda based on a cron job frequency, such as once a day or every 30 minutes.
2. Lambda makes an API call to Stack Exchange API to get raw data in JSON/CSV format and dumps it into the S3 raw bucket without processing or validation.
3. Event Bridge triggers StepFunction based on the cron job.
4. StepFunction orchestrates the Glue Workflow, which can invoke Glue jobs sequentially or in parallel, either synchronously or asynchronously.

5. The Glue Job leverages PySpark scripts to access data from the raw bucket, apply data transformations and manipulations, and then write the transformed data to the curated/transformed S3 location in the parquet format. The data may also be partitioned if necessary to optimize performance. In addition, the Glue Job creates external tables and uses the Glue Data Catalog as the Hive Metastore for data cataloging.
6. Metadata of external tables is stored in Data Catalog and the data is stored in the S3 data lake.
7. On Glue Job failure, StepFunction sends an email notification alert through SNS to notify the team.
8. Athena uses Glue Data Catalog to read the database and table for serverless ad-hoc SQL execution.
9. QuickSight or Tableau can be used with Athena as a business intelligence (BI) tool for analytics, data visualization, and reporting.
10. Redshift Spectrum enables querying data directly from files on Amazon S3 to perform SQL queries on data stored in Amazon S3 buckets.
11. SageMaker can read from the S3 data lake if the data scientist wants to run Machine Learning workloads.
12. IAM, KMS, Secrets Manager, Cloudwatch, and CloudTrail are used for security, monitoring, and access control.

Pros

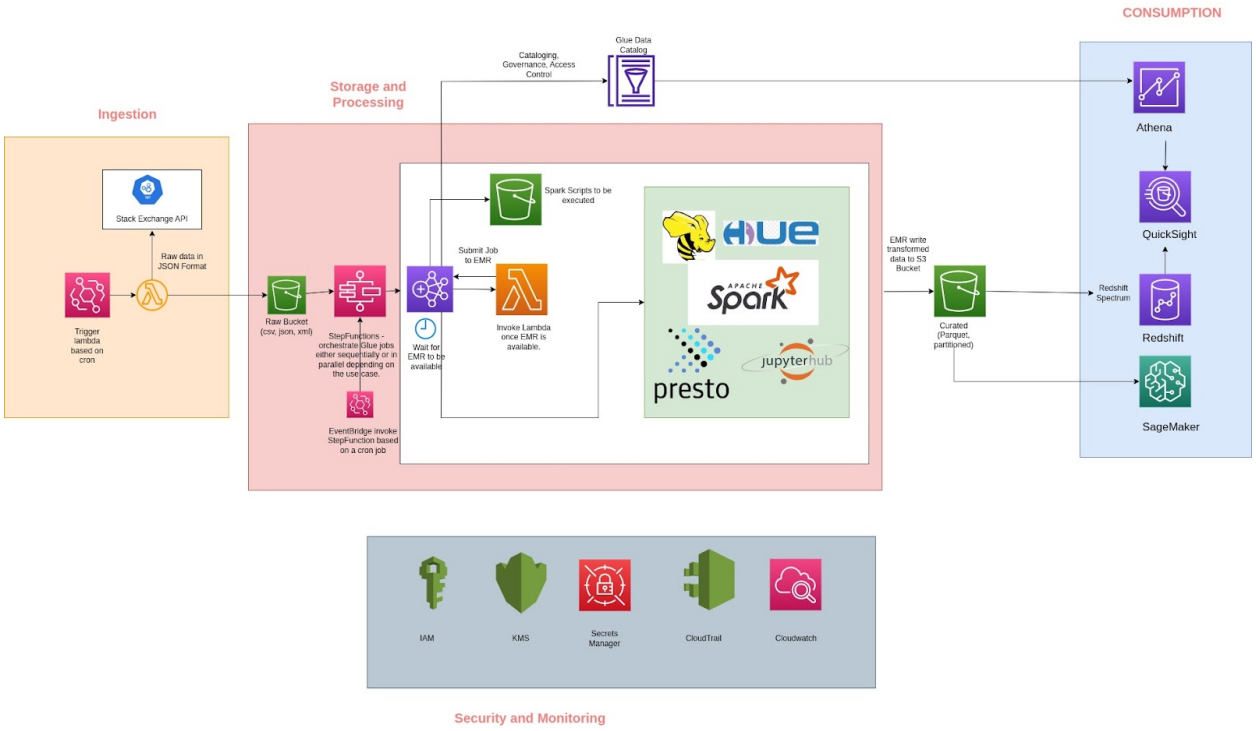
1. Serverless architecture allows for automatic scaling and low operational overhead.
2. Glue is a cost-effective solution for periodic data retrieval.
3. AWS Glue simplifies the ETL process and integrates with other AWS services.
4. AWS Redshift is a fully managed, scalable data warehouse solution that can handle large amounts of data.
5. AWS Athena is a cost-effective solution for ad hoc querying of data in S3.

Cons

1. AWS Glue has limitations on the number of concurrent jobs and processing capacity, which may limit scalability for very larger datasets. For our use case, Glue would be fine.
2. AWS Redshift has higher operational overhead compared to Athena.
3. AWS QuickSight may have additional costs associated with it for data visualization.

Athena is a cost-effective solution for ad hoc querying of data in S3, which may be a good fit for smaller datasets. On the other hand, using Redshift as a scalable data warehouse solution that can handle larger datasets, but has higher operational overhead.

Architecture 2 - Using EMR



`

Explanation of Workflow steps

1. Data ingestion starts the same way as the first architecture
2. StepFunction invokes the EMR cluster and waits for it to be available
3. Once available, StepFunction triggers Lambda to submit job to the EMR cluster using add_job_flow_steps boto3 API
4. Lambda passes s3_script_bucket_path to EMR, which reads the scripts from the S3 bucket and executes them automatically
5. EMR uses S3 as HDFS and Glue Data Catalog as Hive Metastore
6. EMR exposes a URL with JupyterHub Notebook and Apache Hue web app.

7. Hue will offer the ability to use different query engines like Presto, Hive, and SparkSQL to explore data and get insights
8. JupyterHub for the development of batch processing Spark scripts.
9. Data scientists can also use Jupyter Notebook for training and developing light weighted ML models.
10. The results from automatic jobs (step 4) are written to the transformed bucket
11. Metadata is stored in Glue Data Catalog for tables and database.
12. Athena can use Glue Data Catalog to read the database and table for serverless ad-hoc SQL execution
13. QuickSight or Tableau can be used with Athena as a business intelligence (BI) tool for analytics, data visualization, and reporting.
14. Redshift Spectrum enables querying data directly from files on Amazon S3
15. SageMaker can read from the S3 data lake if the data scientist wants to run Machine Learning workloads
16. IAM, KMS, Secrets Manager, Cloudwatch, and CloudTrail are used for security, monitoring, and access control.

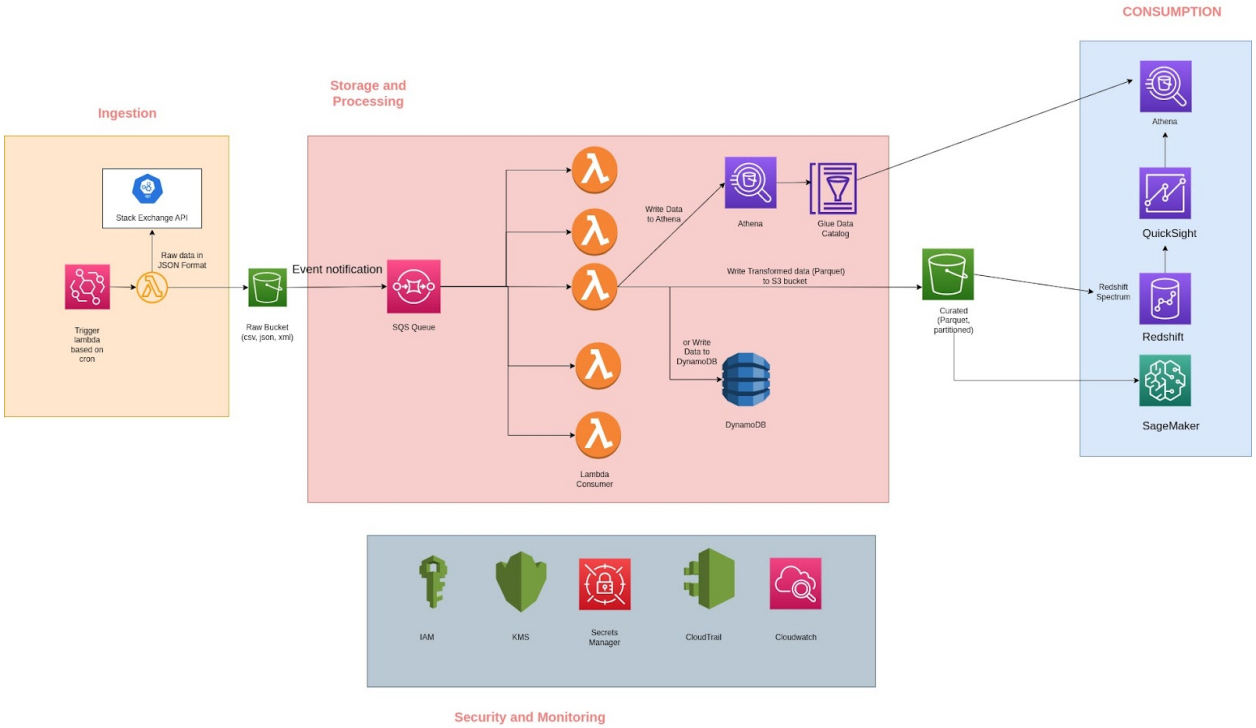
Pros

1. The architecture is highly scalable, as it uses EMR to process large amounts of data in parallel, and S3 as a data lake.
2. Using Glue Data Catalog as a metadata repository makes it easier to manage the data and the associated metadata, reducing the risk of inconsistency.
3. The architecture allows for a variety of query engines to be used, including Presto, Hive, and SparkSQL, providing analysts with more options to explore the data and get insights.
4. JupyterHub provides a great environment for data engineers to develop batch processing scripts, and data scientists to do EDA and POC for light machine learning models.
5. The architecture allows for seamless integration with various BI tools such as QuickSight and Tableau using Athena, querying data directly from S3 using Redshift Spectrum, and enabling data scientists to run machine learning workloads through SageMaker, making it flexible and adaptable to a variety of analytical workflows.

Cons

1. The architecture requires more upfront work to set up, as it involves setting up an EMR cluster.
2. The architecture requires a higher level of technical expertise to set up and maintain compared to simpler architectures. Since the architecture uses multiple services and components, there is a greater risk of failures and complexity.
3. With EMR, you need to manually provision and configure nodes to handle the workload, which can be time-consuming and error-prone.
4. EMR may be less cost-effective compared to Glue for certain use cases. Especially for this use case where Glue will be a more cost-effective choice and can scale up and down automatically to handle changes in workload.

Architecture 3 - Event Driven Architecture



Explanation of Workflow steps

1. Data ingestion starts the same way as the first architecture.
2. Once the file is added, S3 Bucket generates an event, which is sent to an Amazon SQS queue.
3. Lambda function is triggered to read from the Amazon SQS queue, which acts as a consumer of the queue.
4. Lambda function reads data from the S3 bucket and processes it.
5. Light transformations can be performed on the raw data using pandas or **awswrangler**.
6. Transformed data is written to an S3 bucket Data in Parquet format is INSERTED into the table using boto3 Athena API
7. Metadata is stored in the Glue Data Catalog
8. Data can also be written to DynamoDB for later analysis.
9. The consumer remains the same as in the previous architectures. Other services such as Amazon QuickSight or Tableau can use the OLAP database for analytical purposes.
10. Flexibility in terms of Data solution. We can use RDS, No-SQL, SQL, OLTP, OLAP(Amazon Redshift, Amazon Athena), or any other database/datawarehouse/data lake solution.

Pros

1. Horizontal scaling, allows for easy handling of an increase in the number of files.
2. Events can be processed by multiple systems, allowing for greater flexibility in implementing new features or changing existing ones.
3. Raw data can be tracked, managed, and replayed if necessary
4. Data can be processed in real-time, enabling quick and efficient analytics.
5. Decoupling of system.

Cons

1. Event-driven architectures can be more complex than traditional architectures, as they require additional components such as message queues, event processing systems, and event producers and consumers.
2. The order in which events are processed can be important, leading to potential issues if same events are processed multiple times at the same time by consumers.
3. Troubleshooting event-driven systems can be more challenging than traditional systems, as there are multiple components and layers involved.

4. Events can be lost if they are not properly handled or if there are system failures, leading to potential data loss or inconsistencies.

Data Consumers

1. Data Analyst - Ad-Hoc SQL Queries using Athena or similar.
2. Data Scientist - Applied Machine Learning and EDA using SageMaker or EMR Jupyter Notebooks.
3. Business User - Data Visualization using QuickSight or Tableau.