Project Report

on

“Real Time Data Streaming Application”

CPSC 531-03 22470

Advanced Database Management

Fall, 2022

Under Guidance Of

Prof. Tseng-Ching James Shen

Department of Computer Science

California State University

Fullerton CA – 92831

December 2022

Prepared By:-

Onkar Muttemwar(885199950) - onkar.muttemwar@csu.fullerton.edu

Sambhaji Ippar(885865899) - sambhaji@csu.fullerton.edu

Aman Rathore(885186841) - aman.r@csu.fullerton.edu

Contents

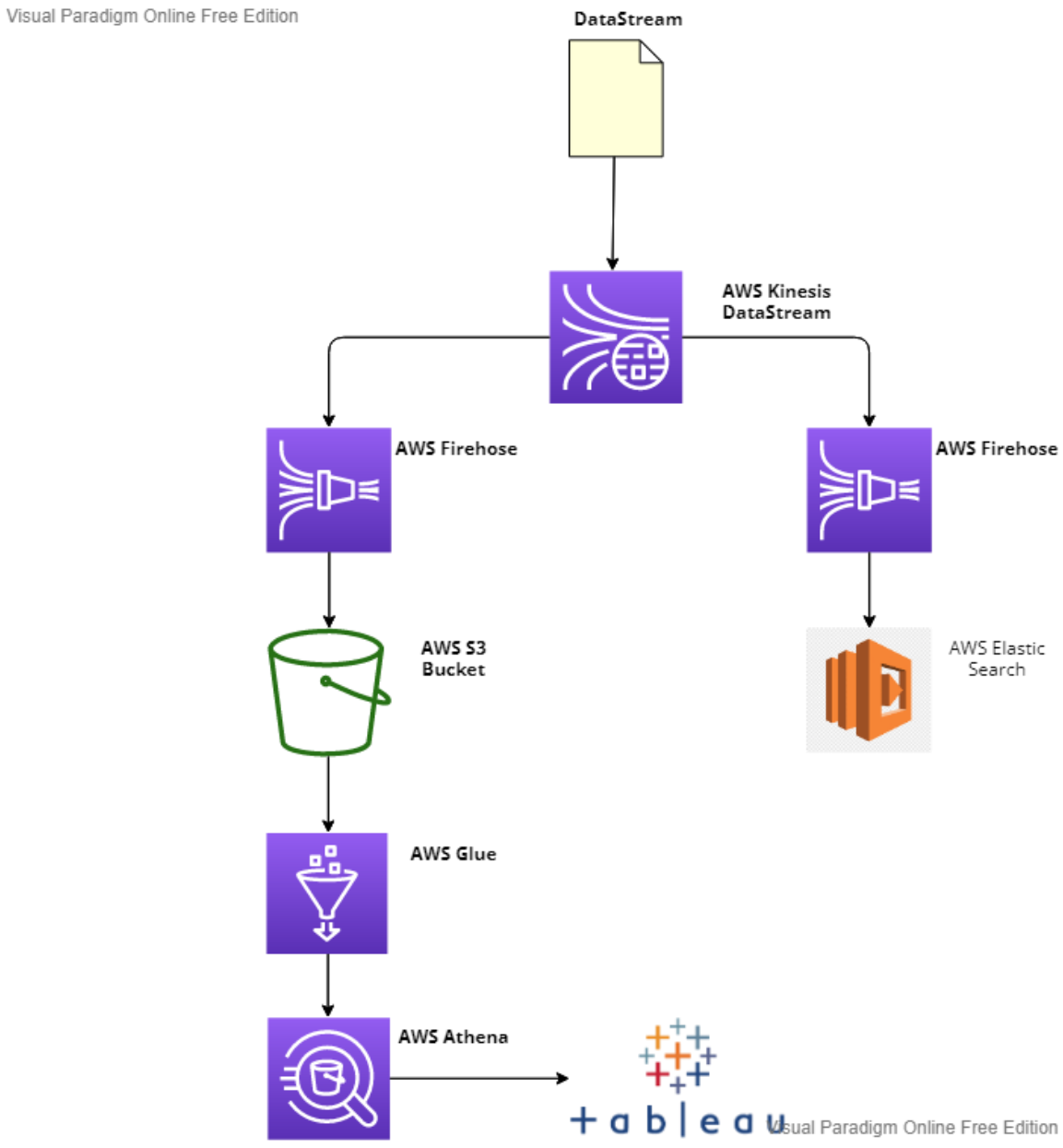
Introduction

- Real-time Data streaming is a process by which big volumes of data are processed as soon as they are generated as continuous streams.
- According to reports, more than a quarter of the data created would be real-time. There are a lot of data sources that create wuch type of data, which are IoT sensors, smart devices, and gaming applications which produce data at high volumes and high velocity.
- So there is a need to process this data in real-time as there is some business where real-time processing and analytics are of crucial importance to get an edge over competitors, and also it enables faster decision making along with various other advantages.
- Detecting fraud in real-time, ride-share apps, and e-commerce apps are very important examples of real-time processing.

Functionalities

- Producers are simulating the data stream rapidly, which is then getting ingested by AWS Kinesis in real-time.
- The ingested data in AWS Kinesis is partitioned using shards and sent to Firehose.
- The two connected AWS Firehose ingested the data from Kinesis. One of them is continuously loading the real-time data to Elasticsearch or OpenSearch while the other one is loading the raw data into our S3 data bucket.
- The AWS Elasticsearch or OpenSearch is monitoring our data and producing visual insights using Kibana
- The raw data within the S3 bucket is getting transformed by glue crawlers and moved to Athena
- The AWS Athena is then querying the data and producing the visual insights in Tableau which is connected to Athena server.

Architecture Overview:



- The data Stream is a python program which is simulating the data in real-time
- The output data stream is ingested by DataStream which is then processed by KinesisFirehose
- It is then used by OpenSearch for real-time analysis using Kibana
- The other Firehose is used by S3 for Batch processing which is then used to add data in glue tables and analysis using Athena and Tableau

Technologies and tools used:

1. Python

2. Tableau

3. AWS Kinesis

4. Amazon Kinesis

5. AWS S3

6. AWS Glue

7. AWS Athena

8. AWS ElasticSearch

Project skills needed but not limited to:

To work on the project, one must have the following skills but not limited to.

- Having a basic understanding of cloud computing.
- Experience with the AWS platform and its various services.
- Knowledge of python programming or, as an alternative, creating a producer code using JAVA.
- Understanding of security best practices and how they apply to the cloud.

Dataset:

- The architecture of the application is such that it would work on any real-time data set. But for the project's scope, we have used the Bank Marketing Dataset from Kaggle.
- The dataset has a lot of columns from which we can extract a lot of insights that would be extracted to analyze in real-time.
- The dataset can be downloaded from Kaggle https://www.kaggle.com/datasets/janiobachmann/bank-marketing-dataset

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	deposit
2	59	admin.	married	secondary	no	2343	yes	no	unknown	5	may	1042	1	-1	0	unknown	yes
3	56	admin.	married	secondary	no	45	no	no	unknown	5	may	1467	1	-1	0	unknown	yes
4	41	technician	married	secondary	no	1270	yes	no	unknown	5	may	1389	1	-1	0	unknown	yes
5	55	services	married	secondary	no	2476	yes	no	unknown	5	may	579	1	-1	0	unknown	yes
6	54	admin.	married	tertiary	no	184	no	no	unknown	5	may	673	2	-1	0	unknown	yes
7	42	managemer	single	tertiary	no	0	yes	yes	unknown	5	may	562	2	-1	0	unknown	yes
8	56	managemer	married	tertiary	no	830	yes	yes	unknown	6	may	1201	1	-1	0	unknown	yes
9	60	retired	divorced	secondary	no	545	yes	no	unknown	6	may	1030	1	-1	0	unknown	yes
10	37	technician	married	secondary	no	1	yes	no	unknown	6	may	608	1	-1	0	unknown	yes
11	28	services	single	secondary	no	5090	yes	no	unknown	6	may	1297	3	-1	0	unknown	yes
12	38	admin.	single	secondary	no	100	yes	no	unknown	7	may	786	1	-1	0	unknown	yes
13	30	blue-collar	married	secondary	no	309	yes	no	unknown	7	may	1574	2	-1	0	unknown	yes

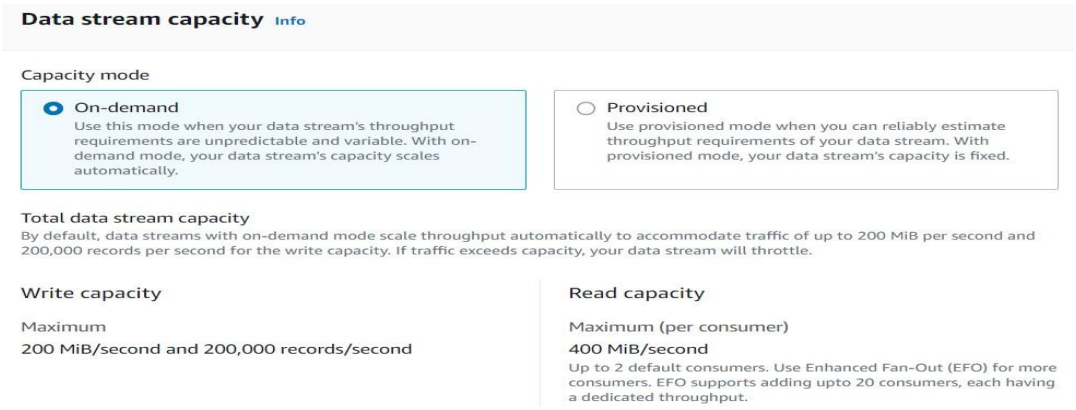
GitHub Location of Code:

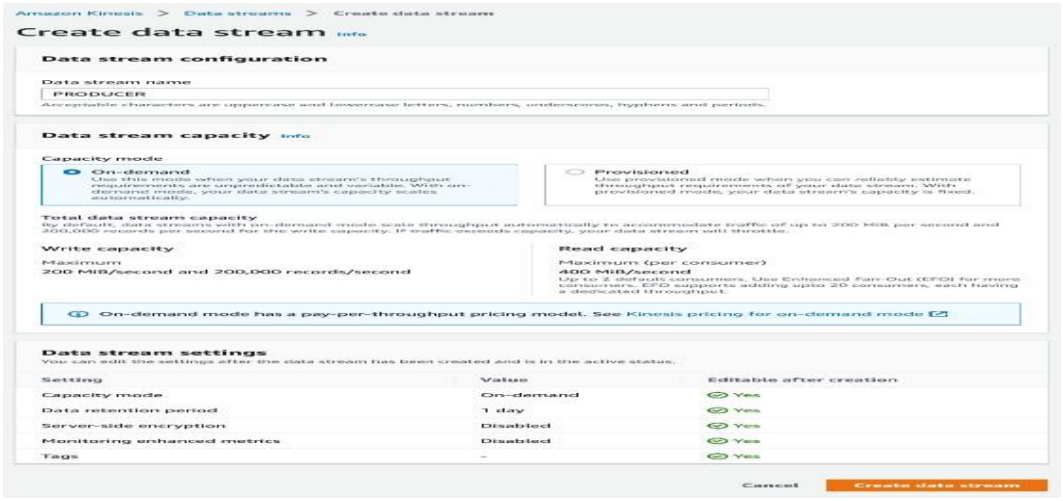
Click Logo

Deployment:

AWS Infrastructure

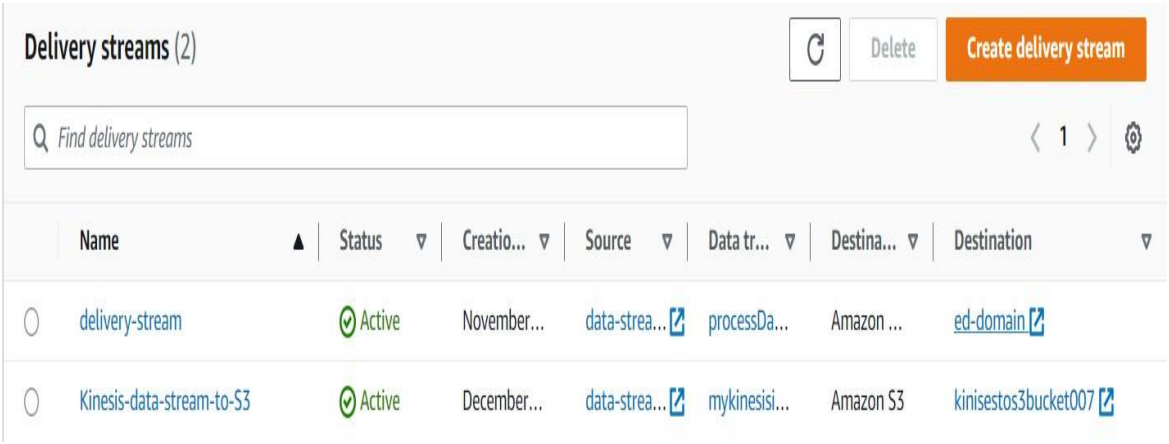
1. **Log in to the AWS portal**
2. **Creating a Data Stream**
    a. After logging in to the AWS portal, search Kinesis and open the Kinesis Portal, now click on the Data Stream tab to create the Data Stream.
    b. This process is relatively straightforward; there are two capacity modes which are on-demand and provisioned, where on-demand helps us to automatically scale the data stream and provisioned mode uses an initial set of set resources.





3. Creating a Delivery Stream

- After Data Stream is created click on the Data Stream inside Kinesis to create the new Delivery Stream.
- There are two delivery streams to be created one for the real-time scenario where it is used for ElasticSearch, another one is used for the batch flow, and S3 is the Destination.

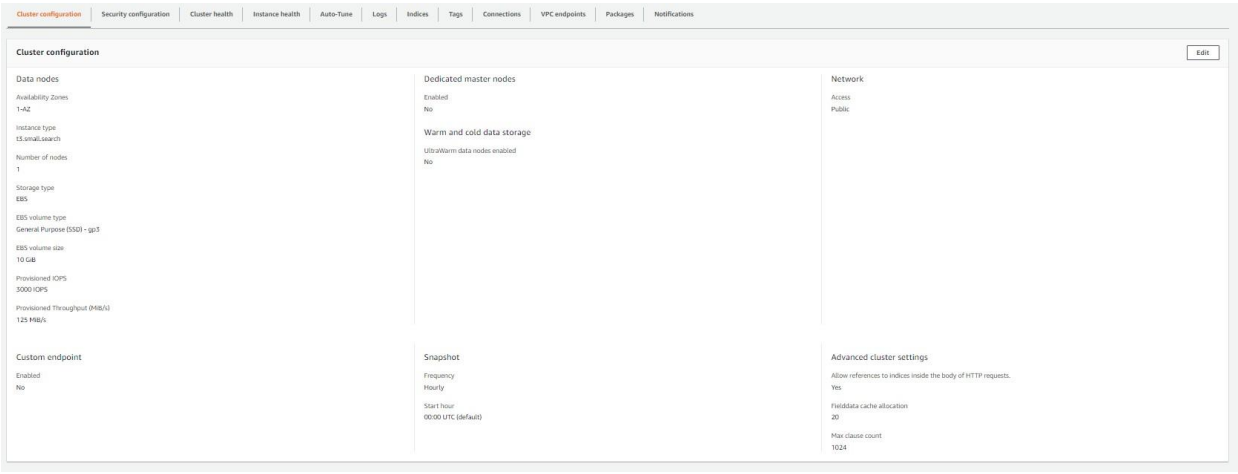


- The configuration of the first delivery stream is as shown in the image
  1) Source would be the Data Stream we just created

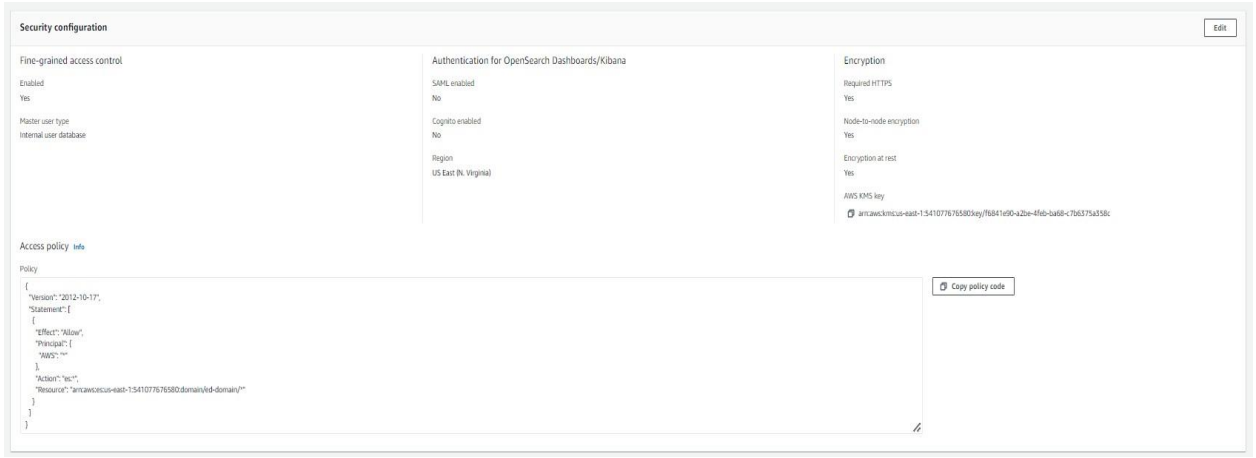
2) Transform records are enabled which uses Amazon Lambda to transform the data during the process
3) Destination is an OpenSearch domain which is similar to the cluster which has the following Cluster and Security configuration.
4) We have added an S3 location as a backup and also the error logs would be saved in the S3 location which was created
5) Delivery Stream2 is similar to the first delivery stream only the destination is S3 in the second delivery stream

**** Use the following configuration while creating the services**

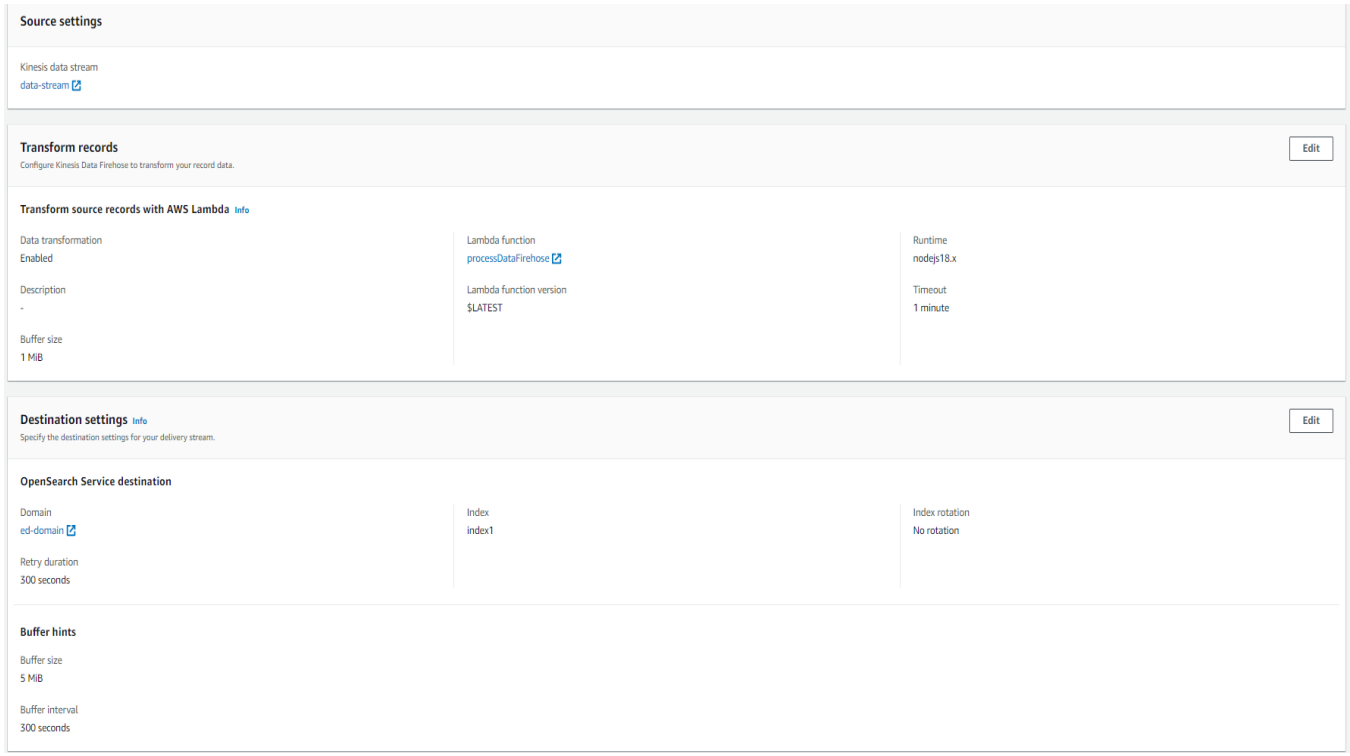
ElasticSearch Domain Cluster configuration



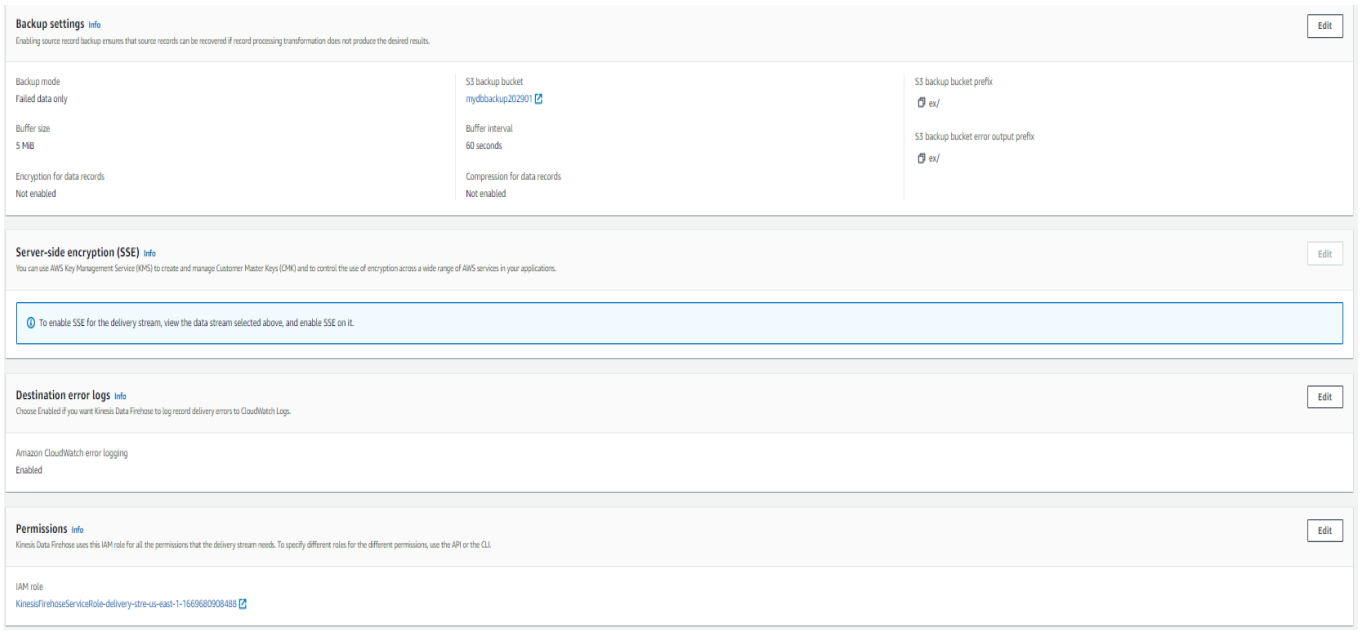
ElasticSearch Domain Security configuration



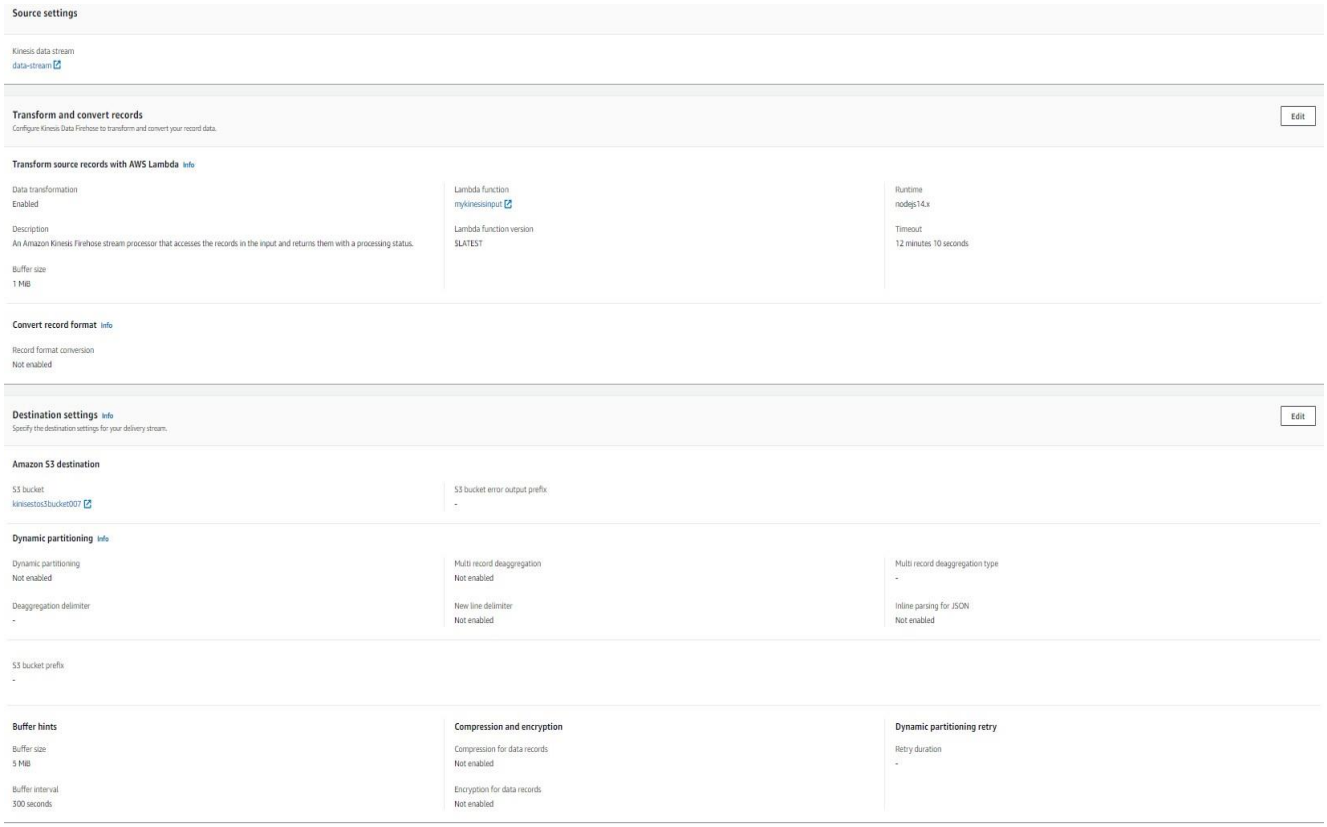
Delivery Stream1 (Part A)



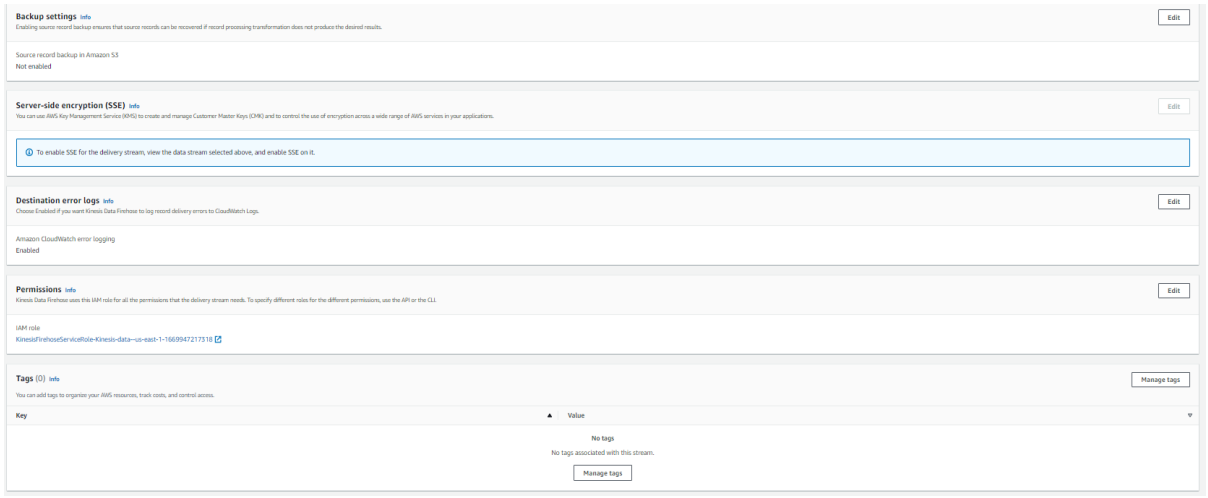
Delivery Stream1 (Part B)



Delivery Stream2 (Part A)



Delivery Stream2 (Part B)

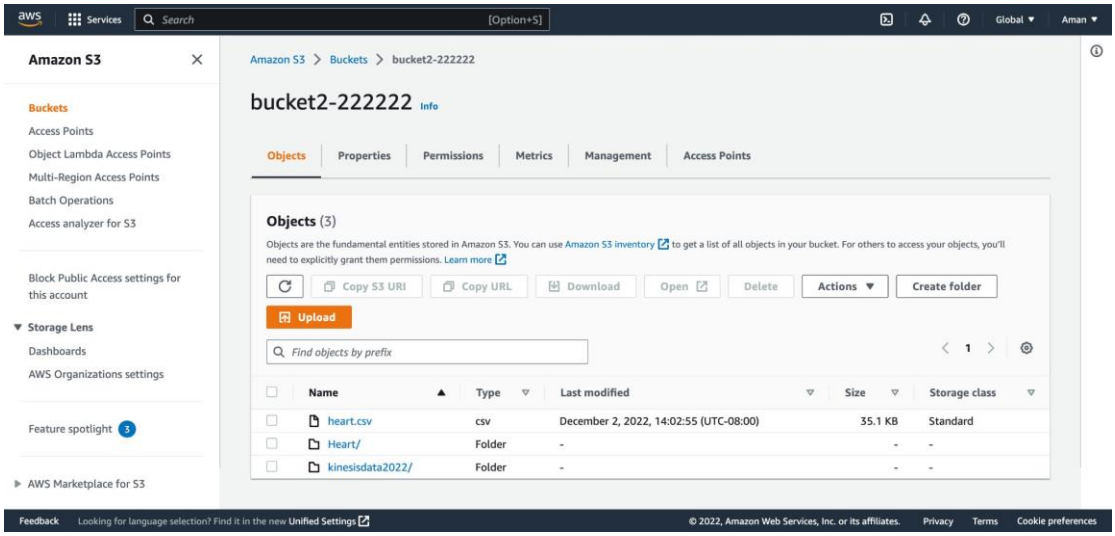


4. Creating a Lambda Function

- After Delivery Stream is created, search for Lambda, click on create a new function and then create and deploy the function.

5. Creating S3 storage

- After Lambda Function is Created, search for Lambda, click on create a new function and then create and deploy the function.



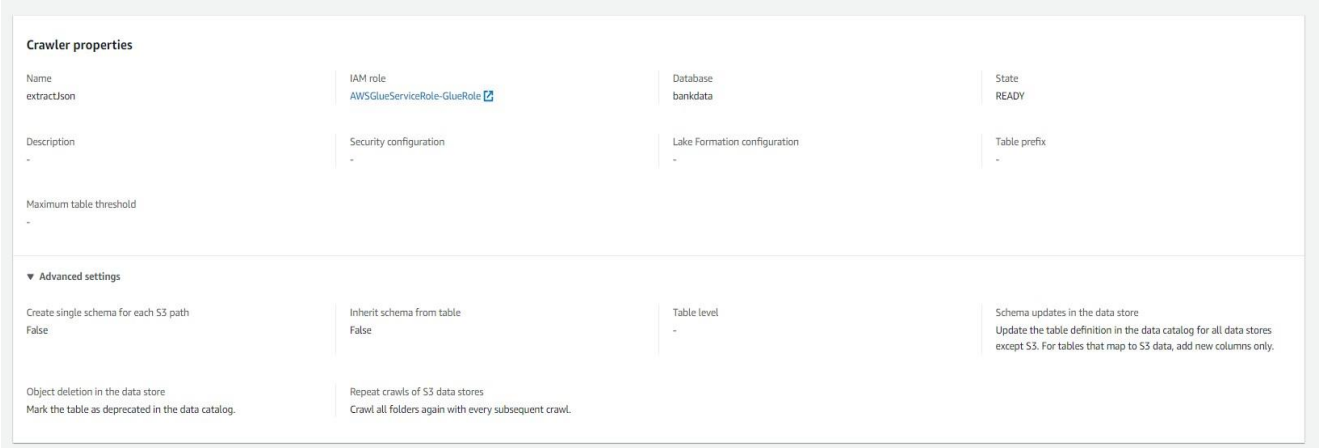
6. Finish creating the delivery stream

- After S3 and Lambda are created we can finish creating the delivery stream

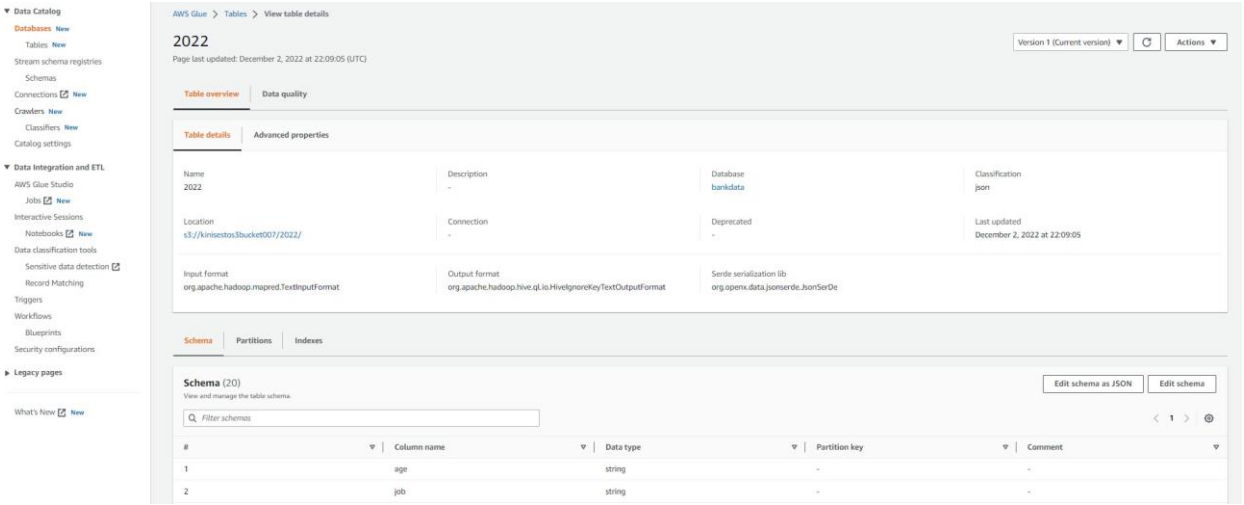
7. Create a Glue Crawler

- Create a new Glue Crawler with the following configuration which would create a GLUE database and tables/

Glue Crawler



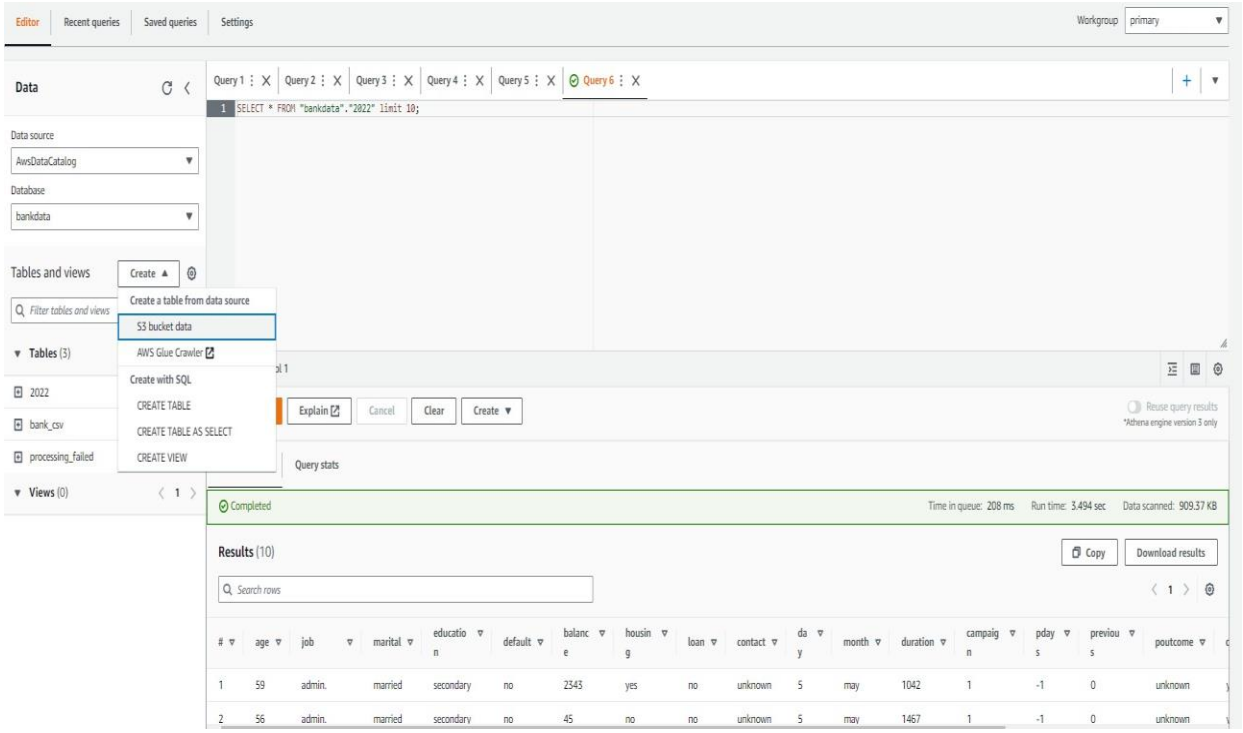
Glue Database



8. Setting up Amazon Athena

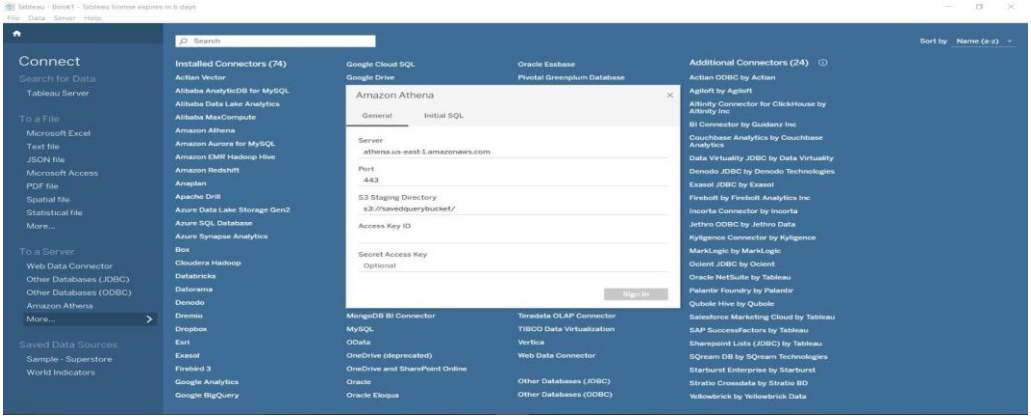
- After Creating AWS Glue Crawler, we need to search for Athena

- After a portal similar to the below image is opened we need to click on AWS Glue Crawler and select the already created Glue crawler.



9. Setting up Tableau

- After completing all the above steps, we need to install tableau and use the below configuration to connect Tableau to the Glue tables.



Steps to Run the Application

1) Download AWS CLI from https://AWS.amazon.com/cli/
2) Create an IAM role and give administrator access (for development) or give suitable access and note down the Access Key ID and the Access key
3) Open the Terminal(Command Prompt) and navigate to the project directory.
4) Type AWS configure and add the previously noted AWS Key ID and AWS Key, region, and output format.
5) Now set up the infrastructure in AWS, as explained above in the deployment stage.
6) After AWS has been set up, run the python file, which has the producer code, and will start adding the data into the AWS Data Streams and eventually in the AWS Firehose.
   ***** Run **python <Python File name>.py**
   ***** Generator Code and Amazon Lambda code can be found in the mentioned GitHub repository.
7) To view the Dashboard, we have to click on the Kibana URL inside the Domain, which we will find inside the created Delivery Stream

Test Results

Kibana Analysis

Below are the results of the live data stream analysis in the form of Dashboard

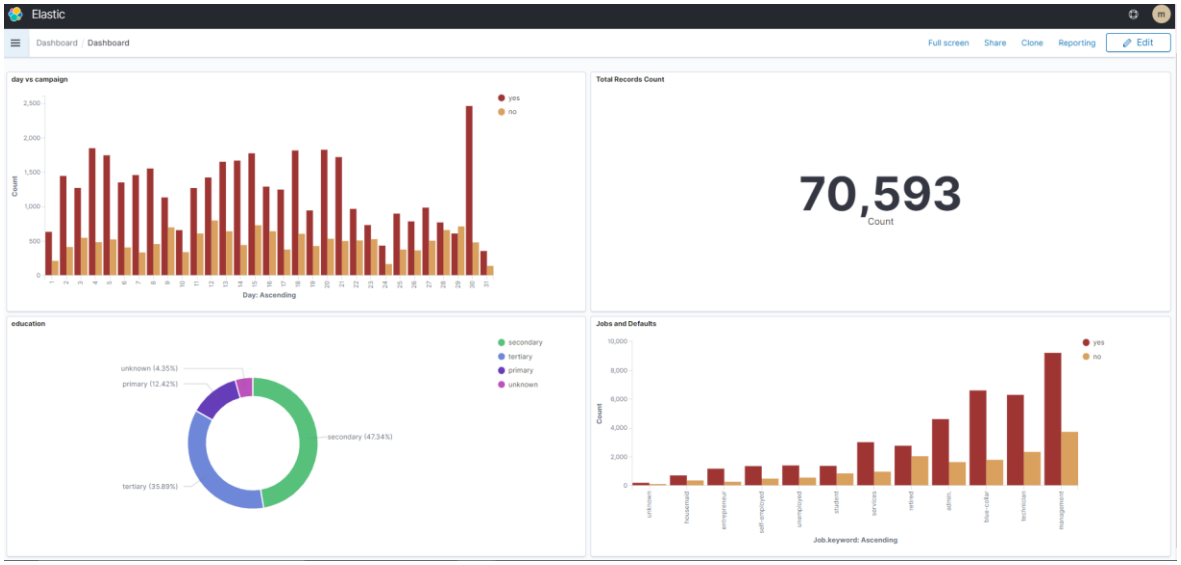
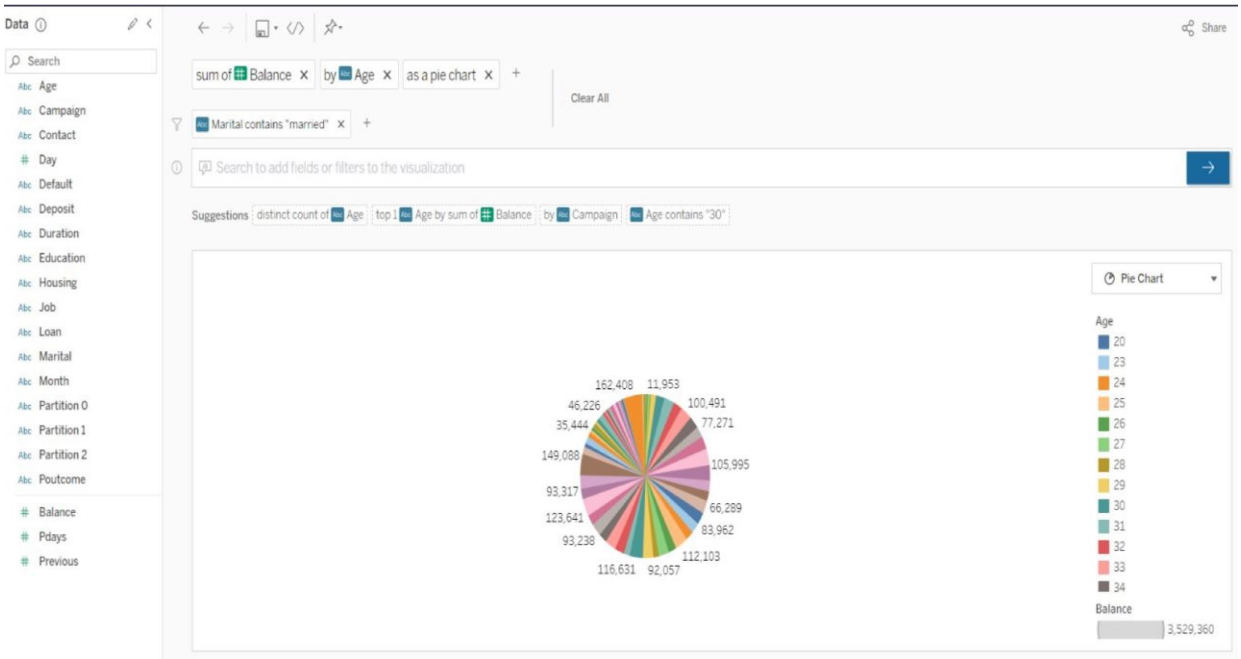
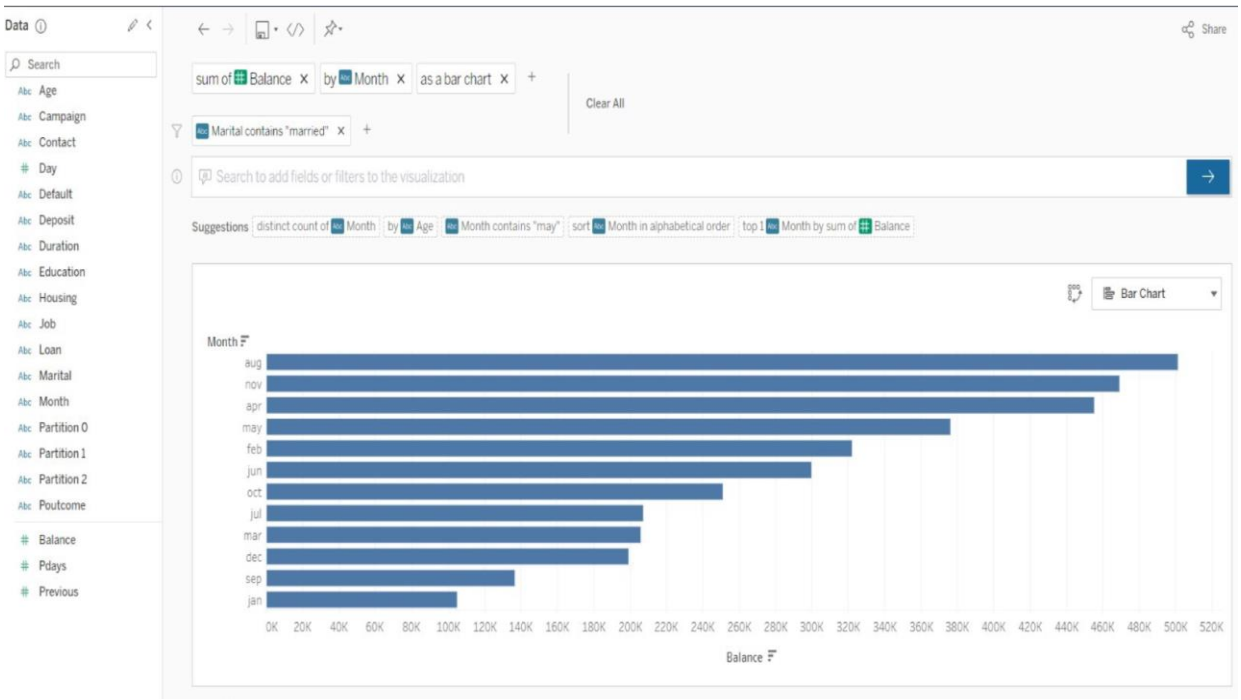


Tableau Analysis

Below are the results for the raw data analysis





References

https://aws.amazon.com/kinesis/

https://aws.amazon.com/kinesis/data-firehose/

https://aws.amazon.com/opensearch-service/

https://aws.amazon.com/s3/

https://aws.amazon.com/glue/

https://aws.amazon.com/lambda/

https://aws.amazon.com/athena/

https://www.tableau.com/