A blue and black logo

Description automatically generated

**CAPSTONE**

JULY’23 | TEAM-3

Special Thanks to our Training Co-ordinator

**Jayanth P Reddy**

|  |
| --- |
| **MENTORS** |
| Jayita Mandal |

|  |
| --- |
| **TEAM MEMBERS** |
| Akshat Kumar Singh |
| Raj Kumar |
| Utsav Sandeep Lakhlani |
| Sambangi Hima Bindu |

1. **Objective**

* Store all these data files in an AWS S3 bucket. Do the entire data profiling to understand the trends in data. Build QC checklist based on the data profiling done. Create data catalogues preferably using glue.
* Build custom data processing scripts to clean, process & aggregate data. The scripts should have failure alerts and should also provide a summary report over email.
* The processed can be loaded in S3 or moved to a data warehouse. The processed data should be able to be directly consumed by the visualization tool.
* Build a live dashboard to show the various trends. You can create any visualizations that might be able to actor/character/platform comparison.

1. **Data Profiling**
   1. **Data Set**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FileType** | **FileName** | | | |
| A green and white file with a green rectangle and black text  Description automatically generated | credits | links\_small | links | movies\_metadata |
| keywords | ratings\_small | ratings |  |

* 1. **Data Dictionary**
     1. Data Dictionary : credits.csv

**A screenshot of a computer

Description automatically generated**

* + 1. Data Dictionary : keywords.csv

A blue and white box with black text

Description automatically generated

* + 1. Dictionary : links.csv

A blue and white rectangular box with white text

Description automatically generated

* + 1. Dictionary : links\_small.csv

A blue and white rectangular box with white text

Description automatically generated

* + 1. Dictionary : rating.csv

A screenshot of a computer

Description automatically generated

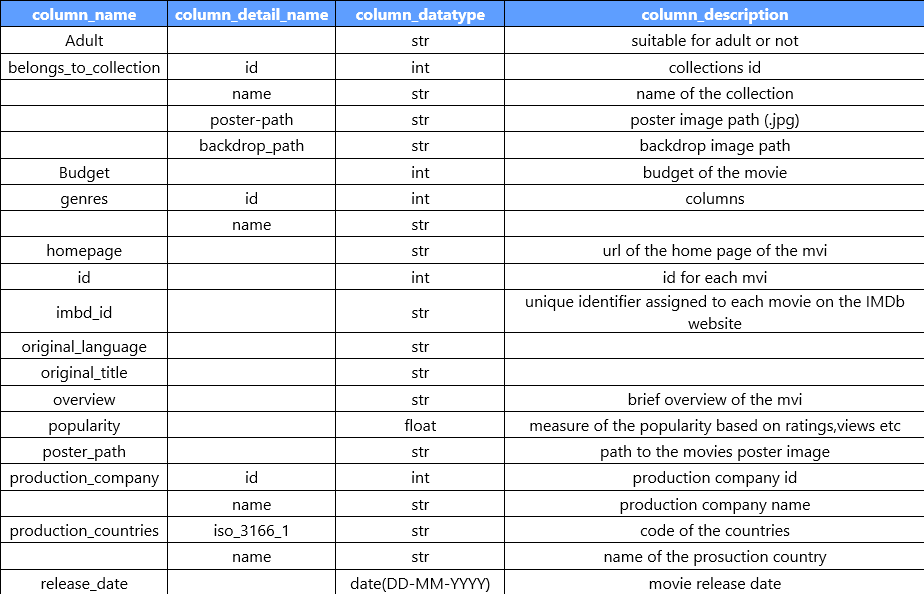
* + 1. Dictionary : ratings\_small.csv

A screenshot of a computer

Description automatically generated

* + 1. Dictionary : movies\_metadata.csv

**A table of words on a white background

Description automatically generated with medium confidence**

1. **Data Pipeline**

Data Pipeline should be in such a way that if raw data comes in S3 Bucket it will be automatically processed through lambda. Glue ETL Job will clean and process the data to processed bucket. If Glue Job Fails then mail will be sent to the concerened person. The Processed data will be then added to Data Warehouse(Redshift) and the same Data Warehouse will be used for visualization in PowerBI and QuickSight.

1. **Approach Architecture**

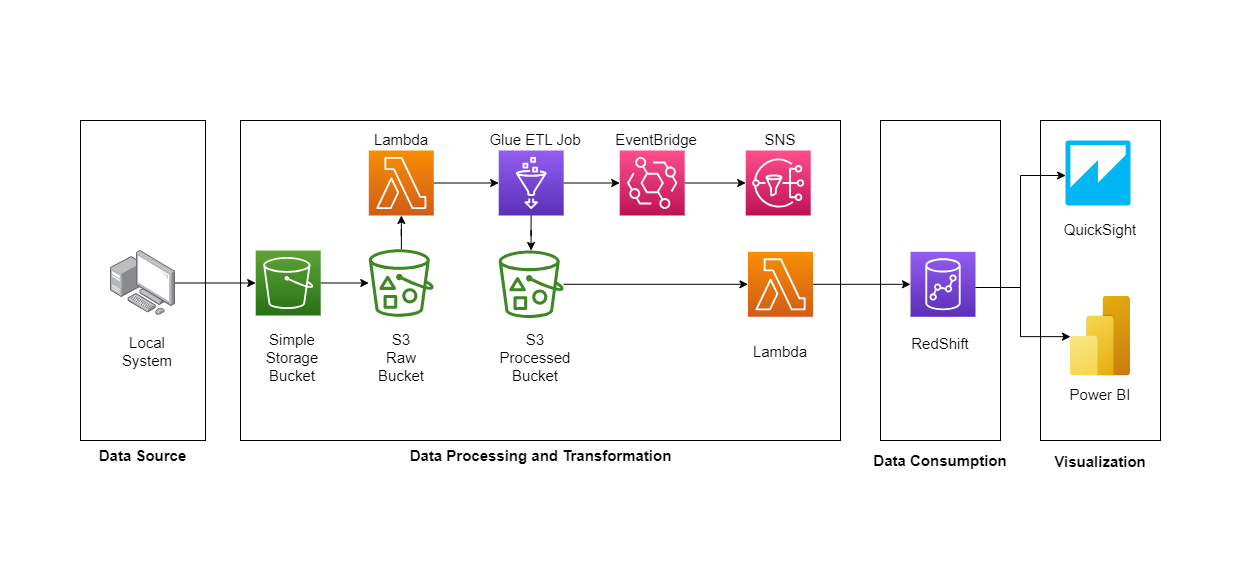


Fig 1: Approach Architecture in Amazon Web Services (AWS)

1. **AWS Services Used**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Simple Storage Service (S3) | Lambda | IAM | Glue |
|  |  |  |  |
| EventBridge | RedShift | Simple Notification Service (SNS) | QuickSight |

* 1. **Steps & Explanation**

1. **Identity Access Management (IAM)**
2. Go to Amazon Web Service Management Console and select IAM under Security, Identity & Compliance in Services.



1. Click on User Groups and create a group by *capstone*.

A screenshot of a computer

Description automatically generated

1. Click on Users and create users to access your account and assign user group as *capstone*.

A screenshot of a computer

Description automatically generated

1. Click on Policies and create policy to attach it with *capstone* user group.

A screenshot of a computer

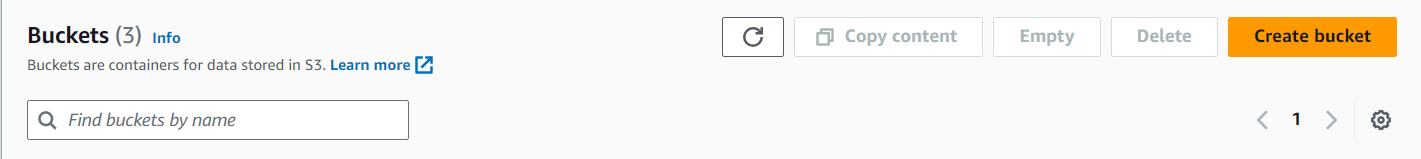
Description automatically generated

1. Now login as IAM user using root’s account number and IAM’s users’ credentials.

A screenshot of a login form

Description automatically generated

1. **Simple Storage Service (S3)**
2. Go to Amazon Web Service Management Console and select S3 under Storage in Services.
3. Click on Create Bucket and Provide necessary details as per the requirements.



After creating the Bucket, the BucketList will appear, click on your bucket.

A screenshot of a computer

Description automatically generated

1. Create 3 folder in the bucket. i.e., input, output, and script

A screenshot of a computer

Description automatically generated

After creating the folders, the bucket will look like:

A screenshot of a computer

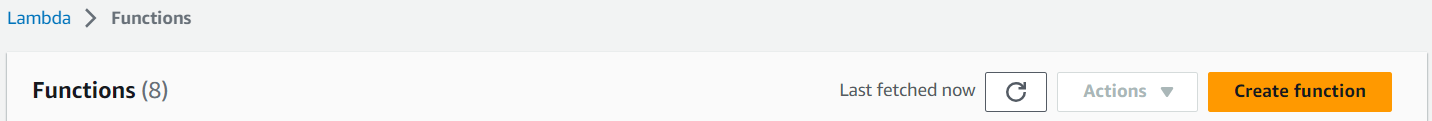
Description automatically generated

Input: To Store Raw Data

Output: To Store Processed Data

Script: To Store Glue ETL Job Script

1. **Lambda**
2. Go to Amazon Web Service Management Console and select Lambda under Compute in Services.
3. Click on Create Function and provide necessary details and create a specific Lambda Role which will have access to Read S3 Objects and GlueExecutionRole.



After Function Creation, Screen will look like:

A screenshot of a computer

Description automatically generated

Click on your function.

1. Click on Create Trigger. In trigger configuration put Source as S3 choose the bucket that you’ve created. Choose Event Types as PUT, prefix as input and suffix as .csv and then click on ADD.

So, what this trigger will do is whenever we’ll upload a file in our S3 input bucket which will have extension as .csv lambda will get to know about this event. Now, we have to write few codes to call Glue ETL Job whenever the event happens. And this is how our first phase of pipeline will work.

A screenshot of a computer code

Description automatically generated

1. **Glue ETL Job**
2. Go to Amazon Web Service Management Console and select AWS Glue under Analytics in Services.
3. Click and ETL Jobs and select Python Shell script editor and create.

A screenshot of a computer

Description automatically generated

1. After creation of job click on your job and write the code for data-cleaning and other operations that you want to do.

A screen shot of a computer

Description automatically generated

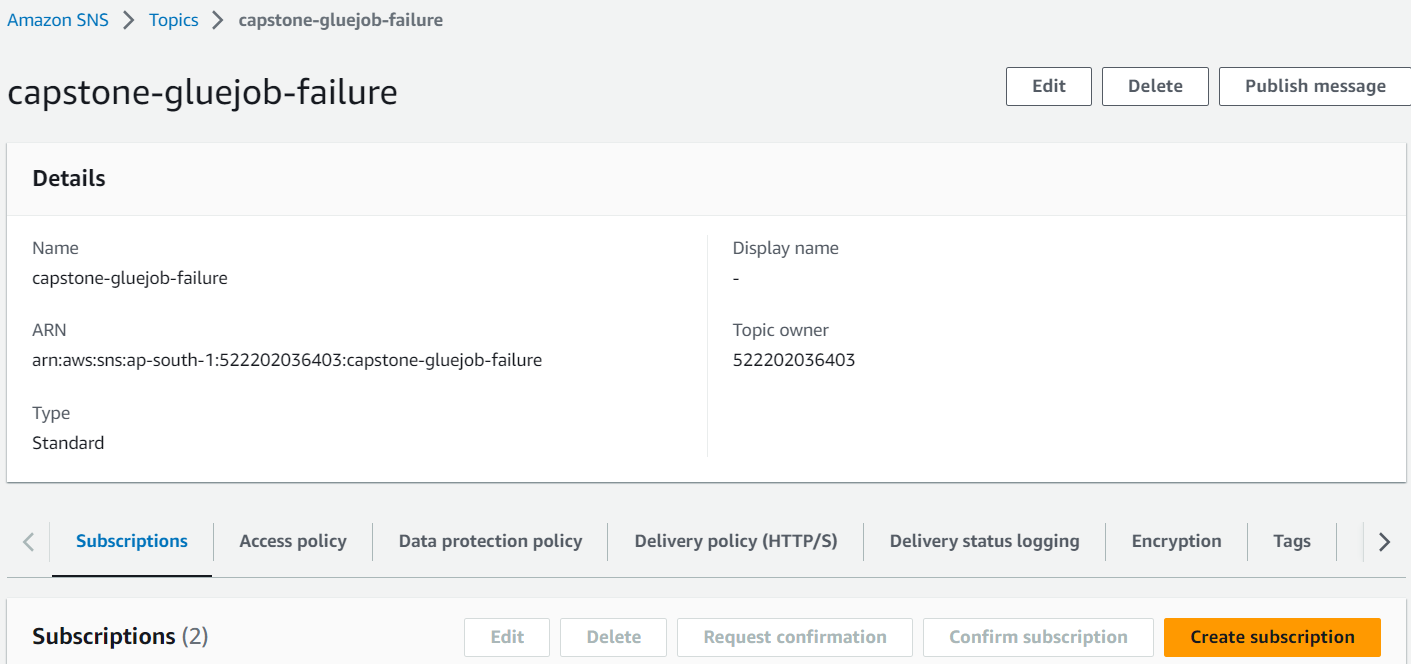
A screenshot of a computer

Description automatically generated

The code will clean the data process it and will store it into S3 Output Folder. And If my Glue Job fails, I must receive a notification. So, for that I’ve created an SNS Topic

And an EventBridge. EventBridge is basically connected to Glue Job and whenever my Glue Job Run Status will change it will automatically trigger SNS to send the mail with the specific error mentioned in it.

1. **Simple Notification Service (SNS)**
2. Go to Amazon Web Service Management Console and select Simple Notification Service under Application Integration in Services.
3. Click on Topics and then Create Topic by the name *capstone-gluejob-failure*. Now your screen will look like this.



Your screen will look like this and now click on Create Subscription. Choose Protocol as Email and in Endpoint mention your email-id. And now click on Create Subscription.

A screenshot of a computer

Description automatically generated

After creating subscription, a mail will be sent to the mentioned email-id for confirming the subscription. After confirmation user will start receiving notification.

1. **EventBridge**
2. Go to Amazon Web Service Management Console and select EventBridge under Application Integration in Services.
3. Click on Rules under Buses and then Create Rule.
4. Mention *capstone-gluejob-failure* as rule name and then click on next.
5. In creation method choose Custom Pattern (JSON Editor).
6. Insert the mentioned JSON Code and then click on Next.

A screenshot of a computer program

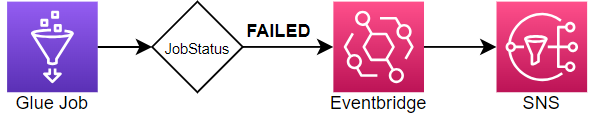
Description automatically generated

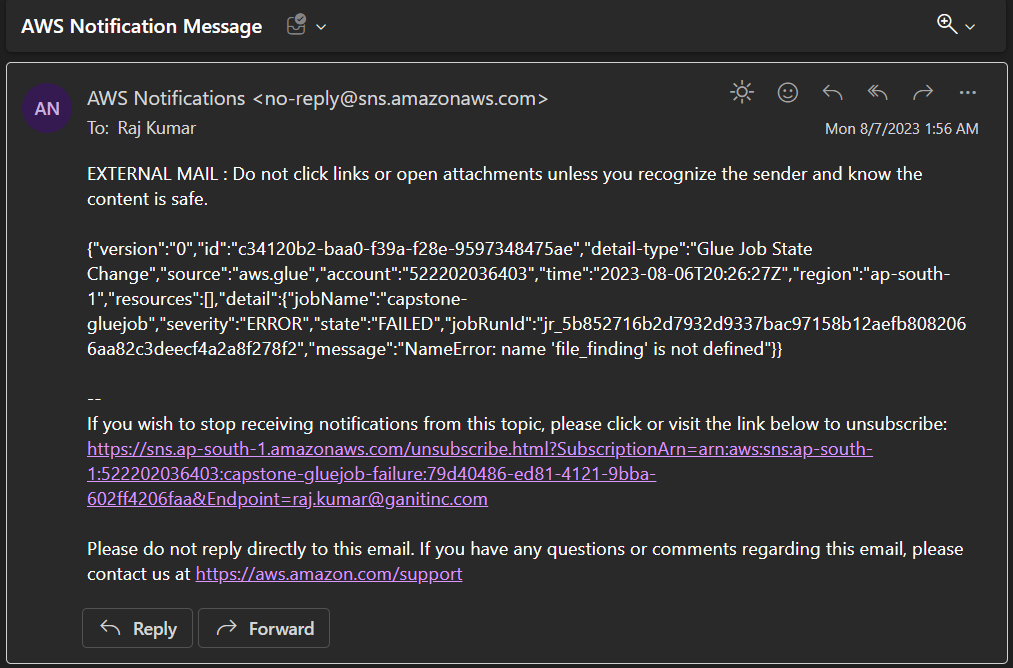
1. For Target 1 choose AWS Service as Target Types.
2. In select a target choose SNS Topic and the select *capstone-gluejob-failure*. Click on next until you get Create Rule.
3. After creation your screen will appear like this.

A screenshot of a computer

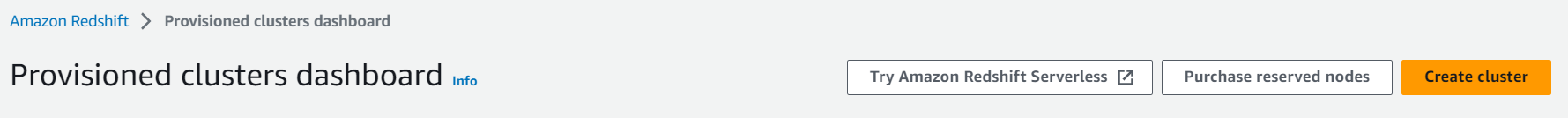
Description automatically generated

So, EventBridge is connecting 2 services for us. Firstly, it’s connected to Glue and if the Glue Job Status Changes it will target the AWS Service SNS.





1. **Redshift**
2. Go to Amazon Web Service Management Console and select Amazon RedShift under Analyticsin Services.
3. Click on Create Cluster.

****

1. Give necessary information and provide credentials for your cluster.
2. In Associate IAM role, create a RedShift Role from IAM and give S3 Read Access.
3. After providing the necessary details click on Create Cluster. And after some time, cluster will be visible in the RedShift Dashboard.

A screenshot of a computer

Description automatically generated

1. Click on your cluster name and then hover over to Query Data and select Query in query editor v2.

A white background with a black border

Description automatically generated with medium confidence

Query Editor v2 Page:

A screenshot of a computer

Description automatically generated

1. **S3 to Redshift**
2. We’re using Lambda to Automatically trigger when processed data is available in the S3 Output Folder.
3. Using Lambda Code, we’re inserting value in tables.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

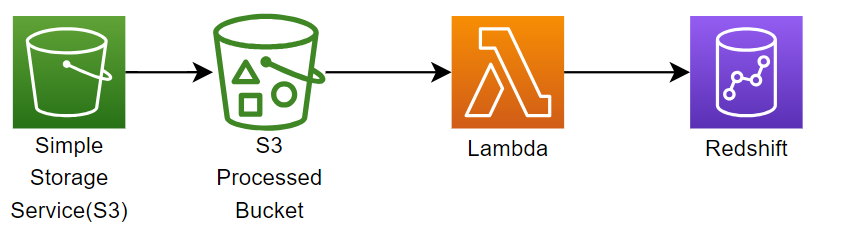
Description automatically generated

S3 Processed Data into RedShift Cluster:

A screenshot of a computer

Description automatically generated

Flow:



1. **RedShift to Visualization**
2. To connect RedShift Cluster to Visualization Tool we’ve enabled Public Access and changed VPC Inbound rules.
3. VPC Inbound rules is right now allow all the traffic to access my cluster.

* Things Needed to Connect Cluster to Visualization Tool

1. Endpoint (Hostname: Port)
2. Username
3. Password

Example in Power BI:

1. Search for RedShift in Get Data

A screenshot of a computer

Description automatically generated

Select Amazon RedShift and Click on Connect.

1. Provide Hostname and Port.

A screenshot of a login box

Description automatically generated

You’ll get this in cluster dashboard in RedShift.

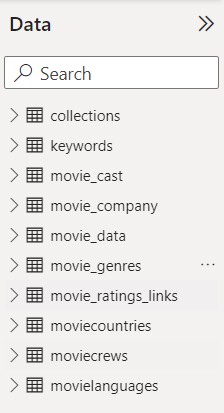
1. Provide db credentials.

A screenshot of a computer login

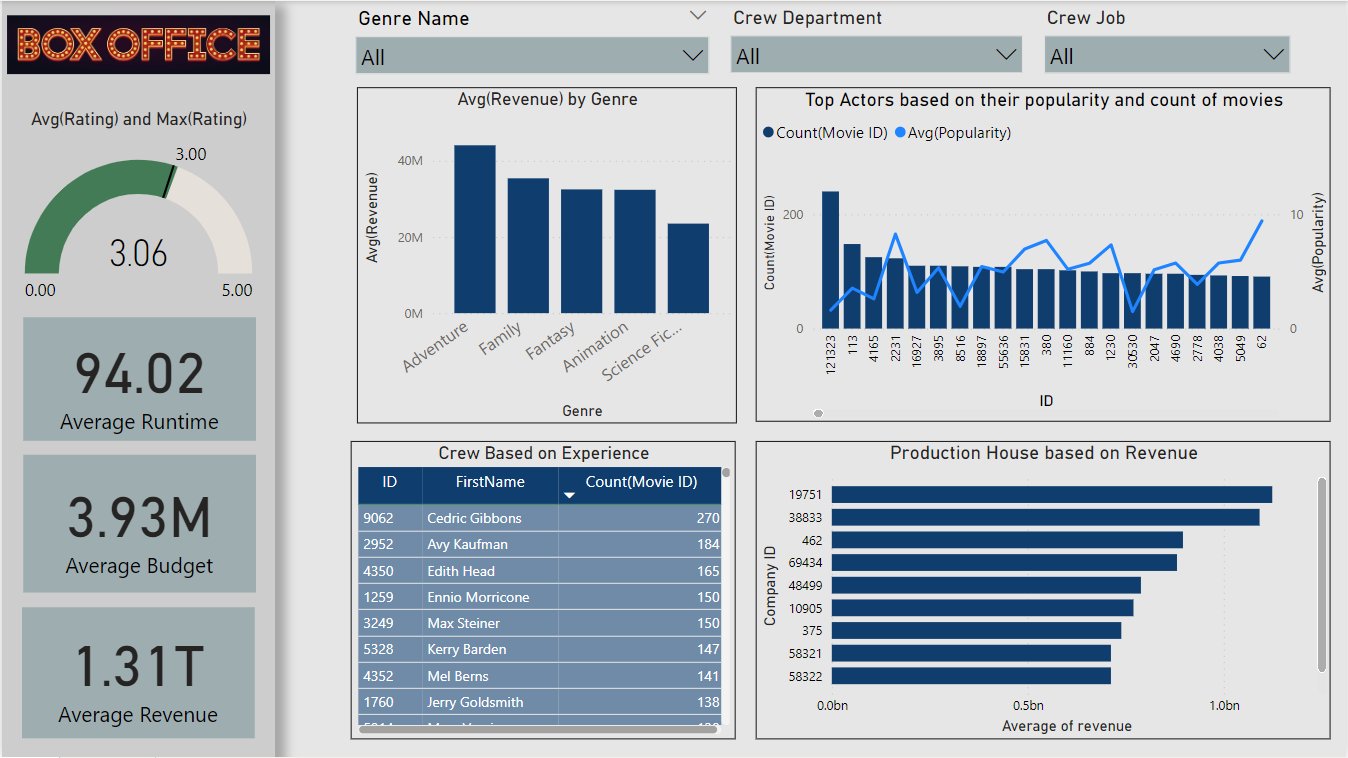
Description automatically generated

And click on connect. If credentials are valid the cluster will be loaded into the PowerBI Dashboard otherwise error will be shown.

1. Data into the PowerBI Dashboard.



1. **Visualization**

****

**A screenshot of a computer

Description automatically generated**