Project Report – Fake News Detection (Data Warehouse)

Aniket Kinage

Chapter 1- Introduction

Motivation

In this project, I have used the Cloud Data warehouse solution for Fake News Analysis. Nowadays, technology has advanced, and data warehouses have evolved. With the fast expansion in data utilization and the move to the big data era, data warehouse design has shifted dramatically from traditional on-site warehouses to cloud-based data warehouses. The cloud service is a critical component influencing the progress of modern data warehousing. A vast number of businesses employ cloud-service data warehouses to achieve low-cost storage, quick scale up/scale down capabilities, flexibility, loss prevention, sustainability, and other benefits.

Objectives

The main objective of this project is to utilize the advantages of cloud-base data warehouse so as to solve the fake news detection problem. To be more specific, we are looking for a data warehouse solution like AWS Redshift that will help us to analyse fake news from large datasets . The solution would improve business decision making and subsequently return more trustworthy outcomes.

Why in your view this is interesting?

The introduction of the cloud data warehouse is one of the most notable recent changes in data warehousing. With the rapid expansion of the digital era, more data typically offers up additional options while also posing significant hurdles for businesses. However, in the past, the data warehouse system struggled under the burden of a large accumulation of useful data. Companies fundamentally want an effective manner that can stock data in many formats and give an advantageous approach to it in order to be able to manage and evaluate a large amount of data efficiently. To overcome classic data warehouse concerns, cloud data warehousing platforms like as Amazon Redshift have stepped in. Cloud data warehousing is widely regarded as a cost-effective way for businesses to leverage cutting-edge technology without making large investments, among other advancements.

Explain briefly how this useful in implementing it on Cloud Computing

If you compare the cloud data warehouse with the traditional one, there are many advantages. The first point to consider when comparing on-premise versus cloud data warehouses is setup time. Before extracting insights that might aid executives in their decision making and business support, the organization has to implement a standard data warehouse for at least a year. In this exponential digital era, such a long period might potentially expose the project to a business decline.

Second, on-premise data warehouses are costly in terms of IT expenditures such as software, hardware, management and security costs. Cloud data warehousing, on the other hand, provides a fantastic option for data security, protection, and recovery. Its services are targeted for a large number of clients. As a result, cloud data warehousing providers may provide strong security at a far cheaper cost than on-premise data warehouses.

The third major factor would be elasticity, flexibility, robustness, and concurrency. For optimum results, organizations must adjust on-premise data warehouses during peak consumption, which may only be a tiny part of the year. Companies frequently require a large investment, as well as administrative costs, to execute it. Meanwhile, cloud data warehousing has significant benefits, such as essentially infinite storage/compute and scalability in users and workload. They can quickly scale up or down storage and computation to meet changing demands. Furthermore, it can quickly increase users and workload without compromising performance, as well as distribute virtual data across several compute clusters, implying concurrency.

Chapter 2- Project's Specifications and Requirements

Project Specifications

- Create an amazon account and navigate to console in AMAZON AWS.
- Create a bucket in S3 and upload files and check for configuration in S3 and set lifecycle (S3->S3IA->Glacier).
- Link your S3 bucket with Redshift to offload traffic on S3 bucket.
- This Dataset is stored in AWS Redshift (Data Warehouse) and retrieved using the Query Editor.
- AWS Quicksight is used to visualise the data in different formats.
- Kindly refer (https://aws.amazon.com/documentation/) for more details about every configuration

Requirements

1) AWS Account
2) Data pre-processing – Python (Jupyter Notebook)
3) Data Storage – Amazon S3, AWS Redshift
4) Data Visualisation – AWS Quicksight
5) Active Internet Connection

Chapter 3- Methodology & System Architecture

Methodology :

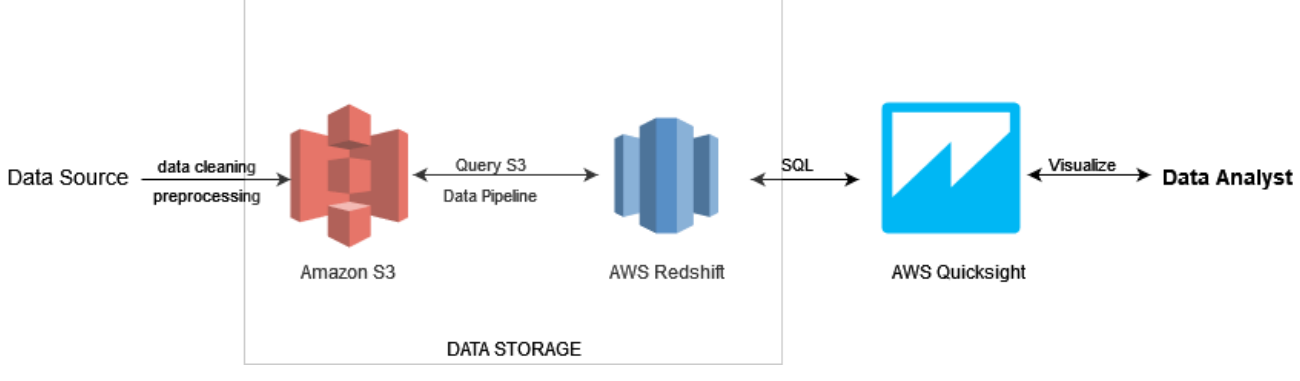
The project is developed in the following stages:

Data Extraction/ Cleaning - Firstly, we collected news data from various sources and stored it in a CSV file. Our data consists of URLs, news headline and its body and the label which defines whether the news is fake or not. In the pre-processing phase, we cleaned the data, that is, we handled the missing values(replaying it by NULL) and removed the irrelevant entries from the data using python. Along with that, we removed the punctuation and stop words present in the news body as it is not required for analysis.

Data Storage – Create an S3 bucket and load the dataset in it. We are going to use the AWS Redshift Data Warehouse solution for storing our data. Once the cluster is running, go to the query editor v2 and create a table with a public schema and add all the columns in it. After creating the table, load the dataset from S3 bucket and use the IAM role you created earlier to read the data into Redshift. Now you can execute SQL queries and ensure that the data is loaded correctly.

Data Visualisation – Finally, we have to visualise the data using AWS Quicksight. First we need to establish connection between the Redshift and Quicksight. For that purpose, we need to create a security rule called redshift and configure it by allowing all TCP connections at the respected port. After connecting to Quicksight we can create dashboards and visualise the data in different forms.

System Architecture –

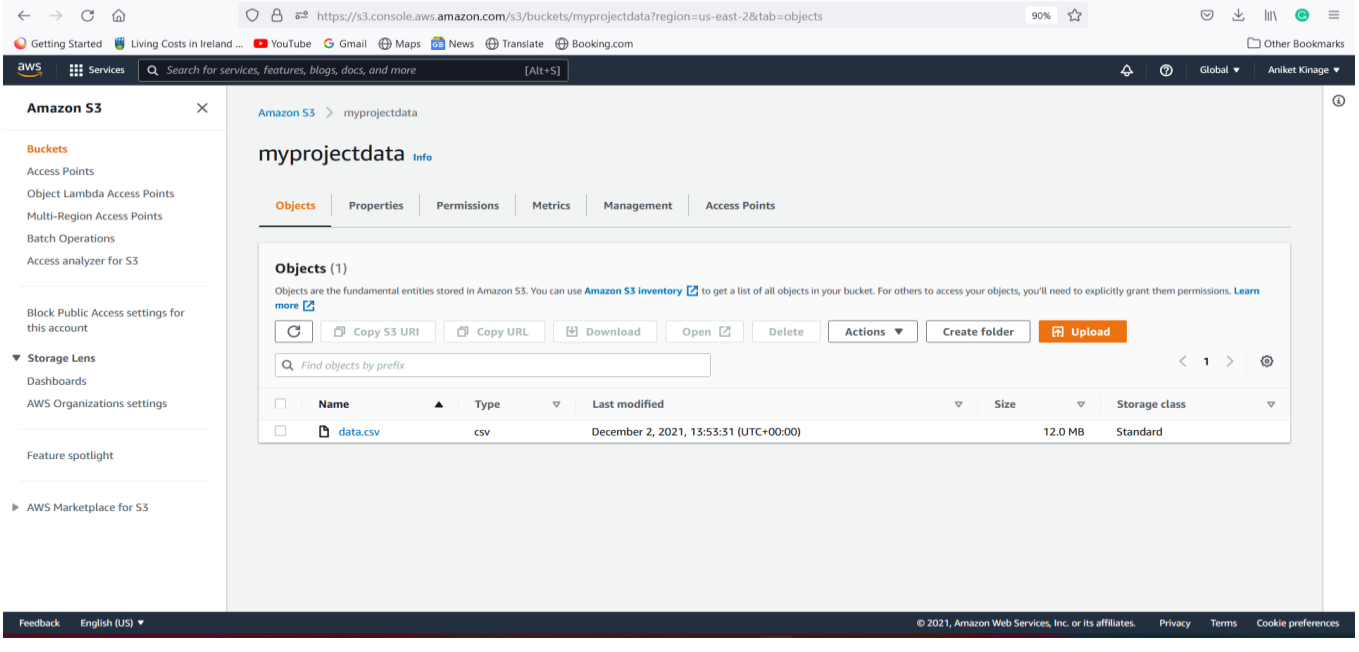


The diagram above shows the architecture design of our project. At first, we collect the news from various data sources. Then we clean the data in the pre-processing phase and store it into the Amazon S3 storage. Next is, we create a cluster and load the data into Redshift by using SQL query(with the help of Query editor) or data pipeline. After loading the entire data in the cluster, we connect the cluster with the visualisation tool, that is AWS Quicksight. Finally we create various visuals and display it on the dashboard for the purpose of analysis

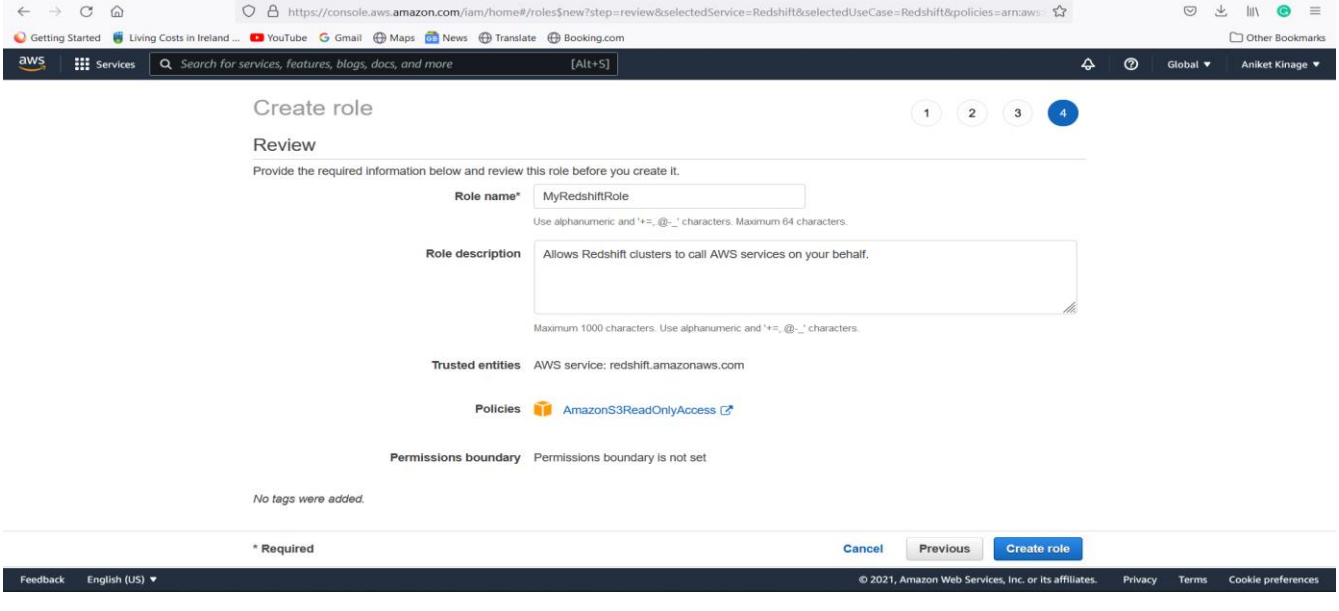
Chapter 4 - Implementation Details

Following are the steps for project execution :

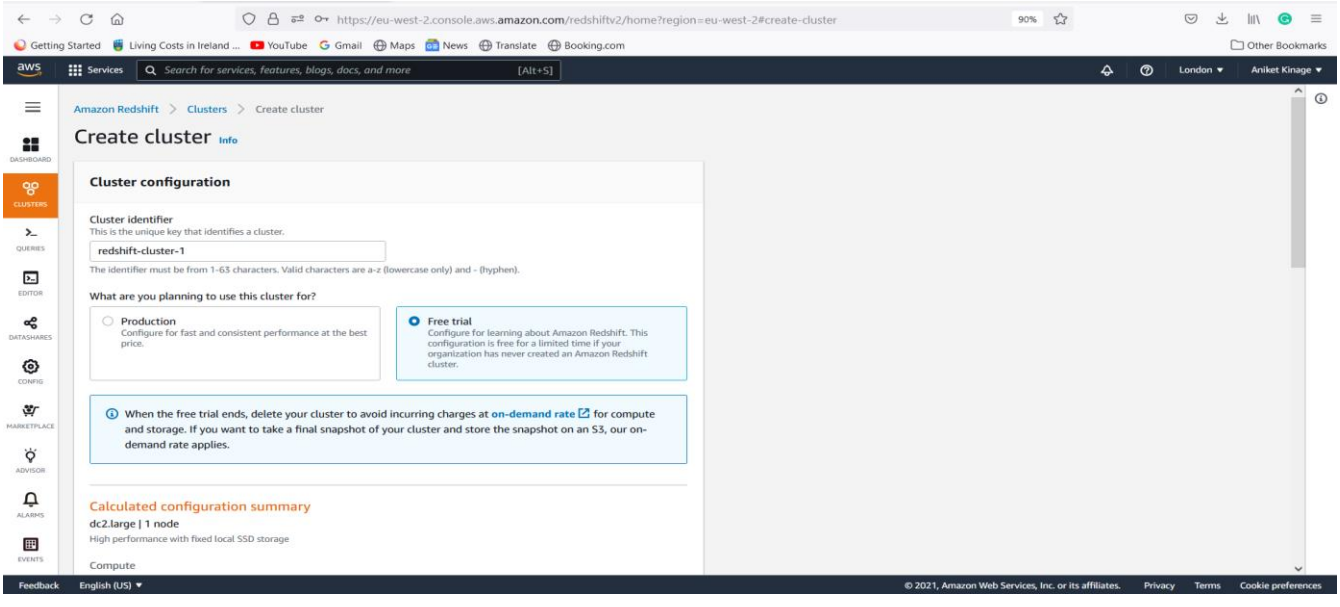
Step-1) Created an S3 bucket called as “myprojectdata” and stored the dataset(data.csv file) in it.



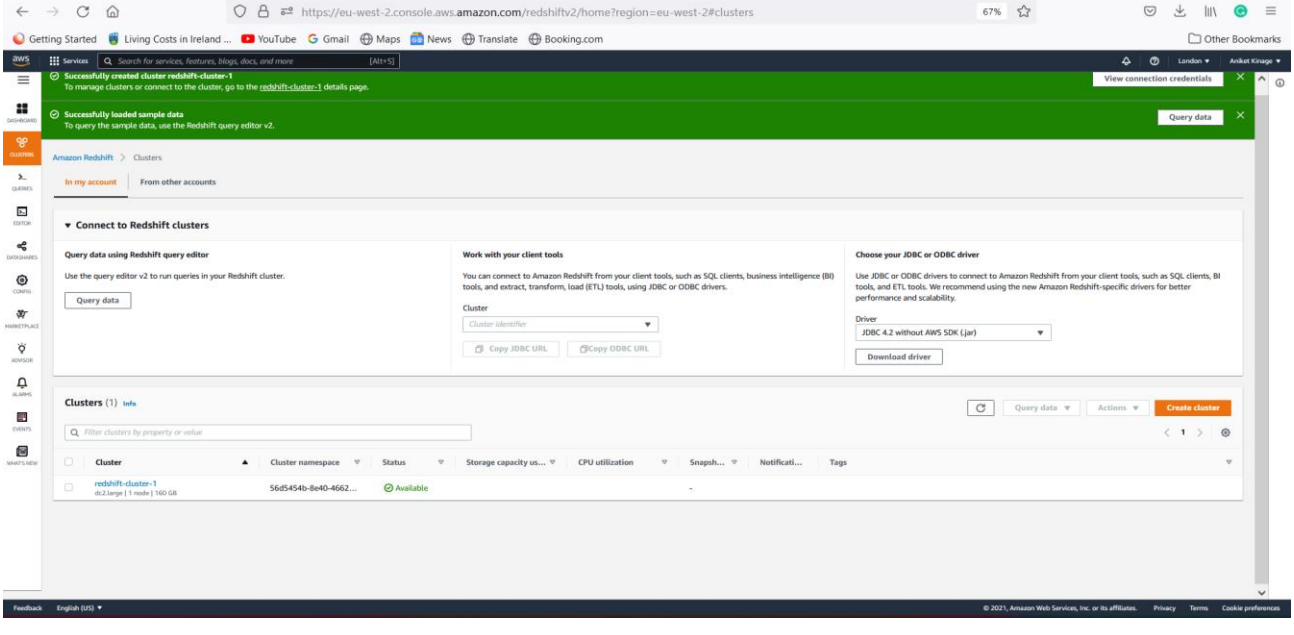
Step-2) Created a Identity and Access Management (IAM) Role to allow Redshift clusters to call AWS services on my behalf.



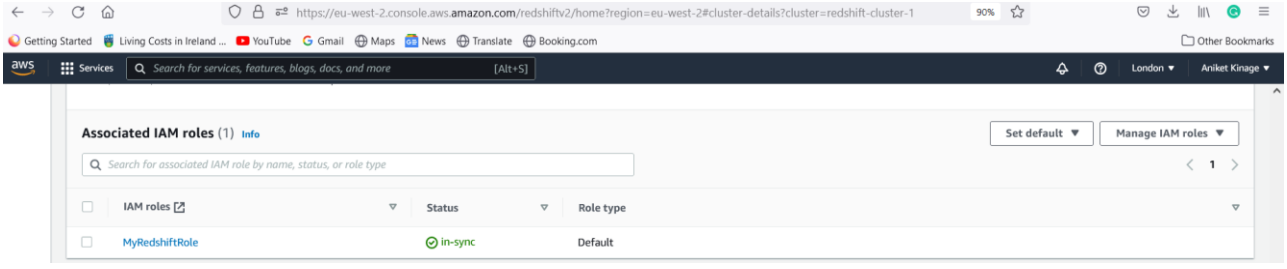
Step-3) Configure the Redshift Cluster. Enter the cluster name and enable the free tier version and create the cluster by choosing the admin username and password.



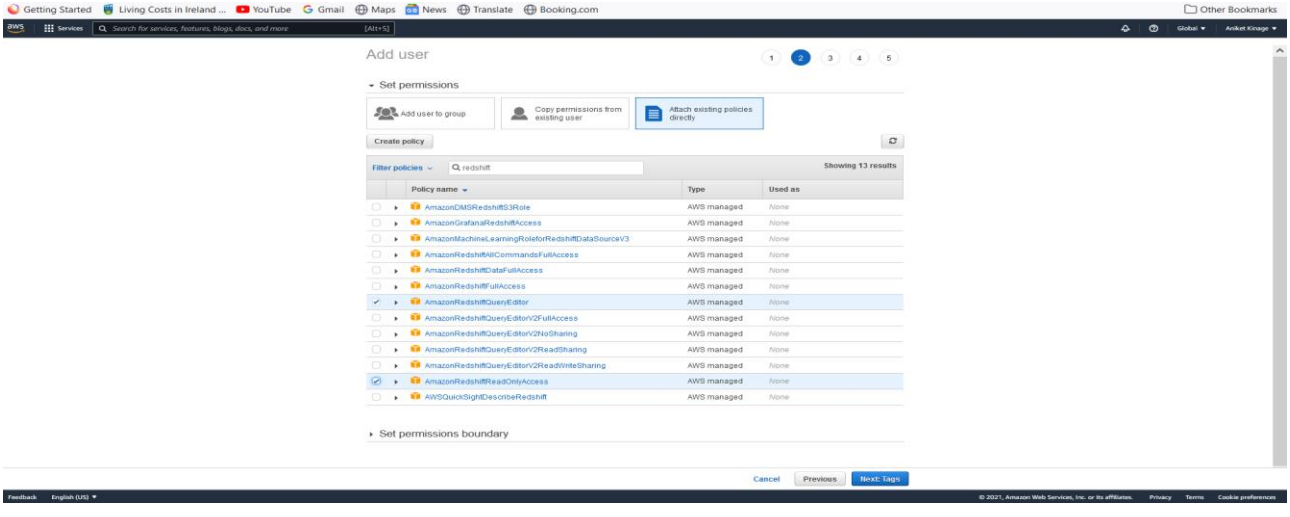
Step-4) Launch the redshift-cluster-1 and wait till it becomes available.



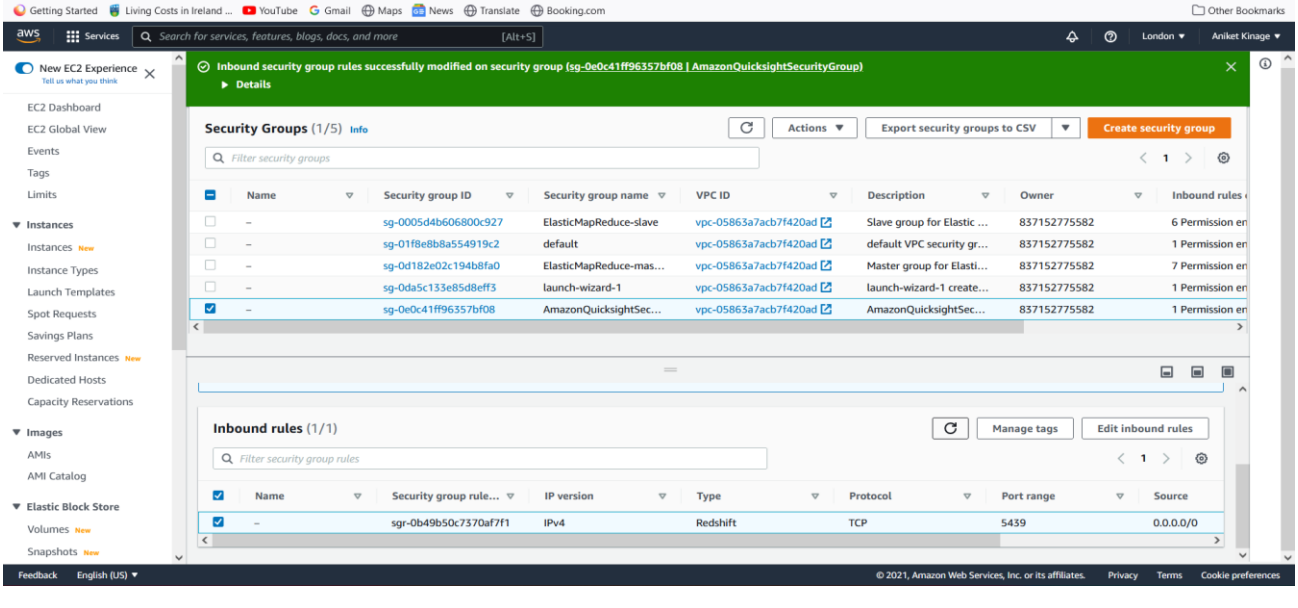
Step-5) Associate the IAM role that you created earlier with the redshift cluster and set it as default.



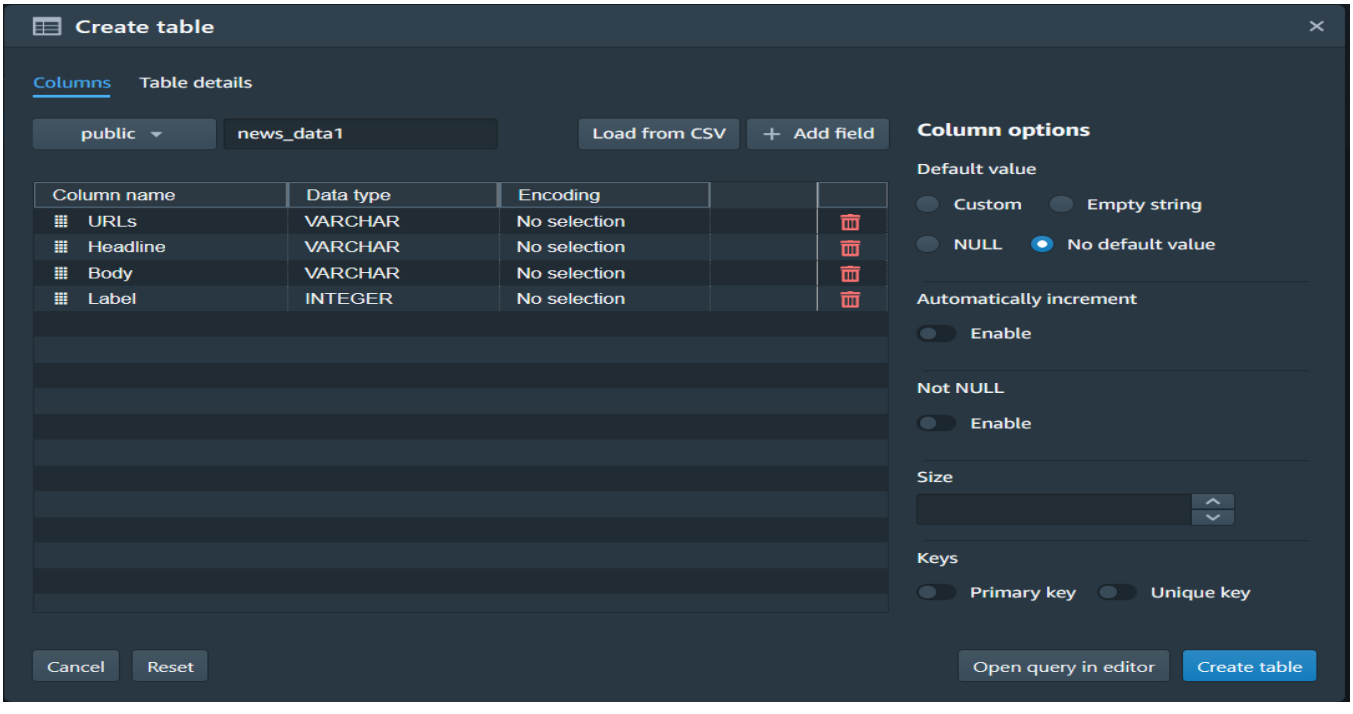
Step-6) Create a user called “Administrator” and grant him the permissions to access the Redshift cluster along with query editor.



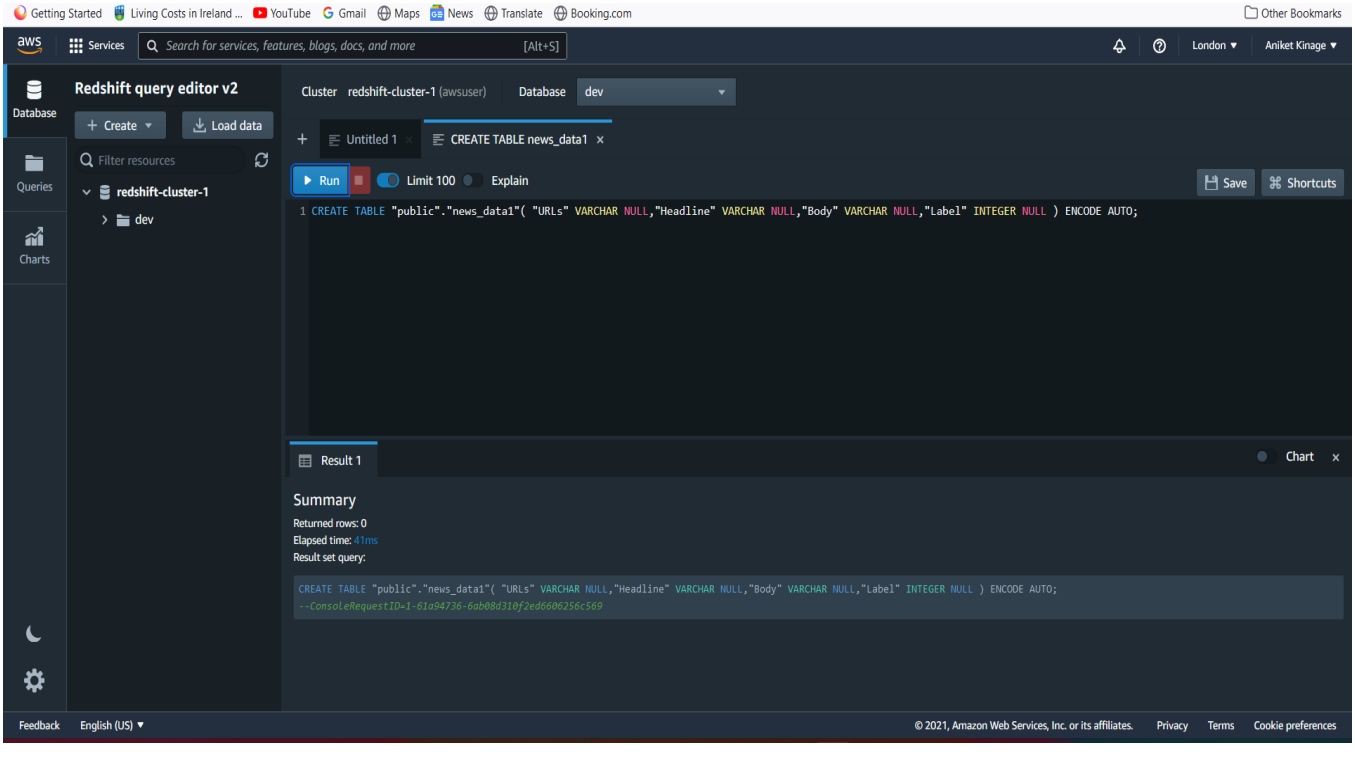
Step-7) Create a new security group of type Redshift with port 5439 for handling the traffic.



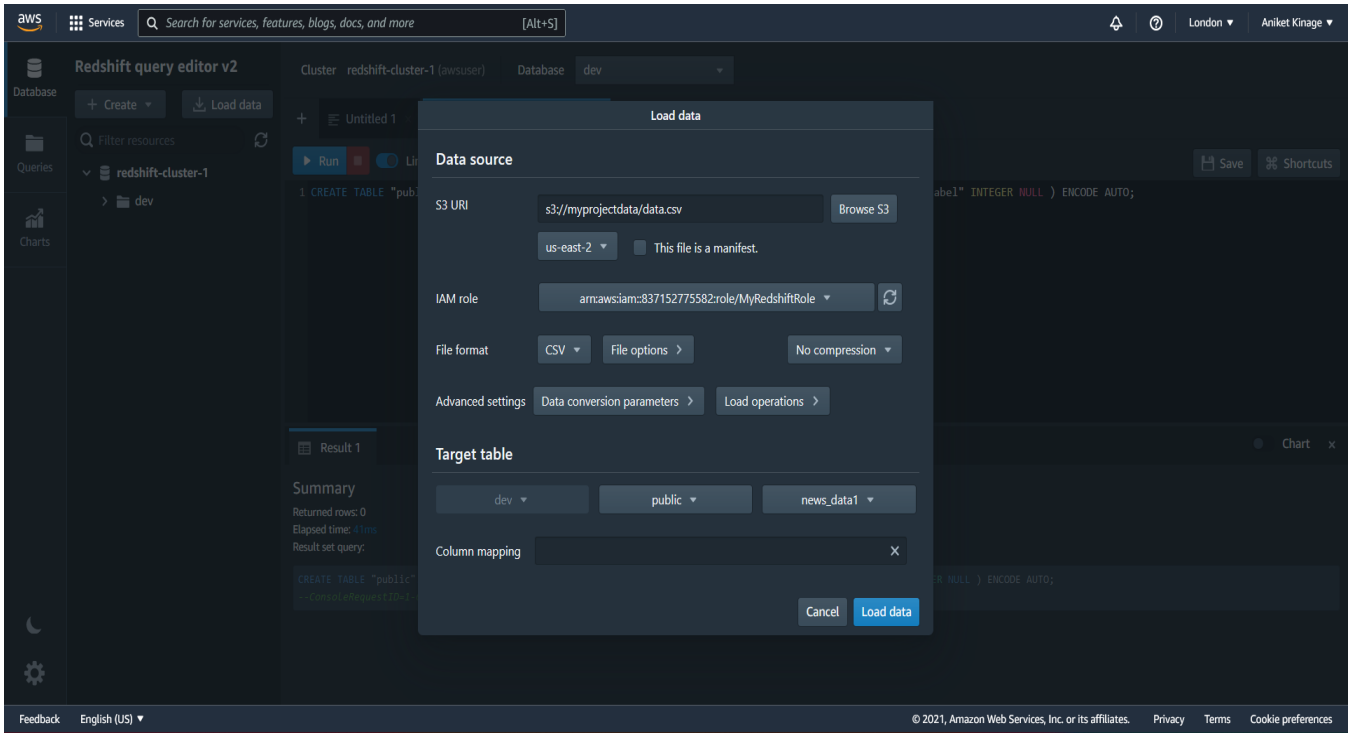
Step-8) Now, open the query editor v2 and create a table “news_data1” . Set the schema as public and create all the columns present in our dataset.



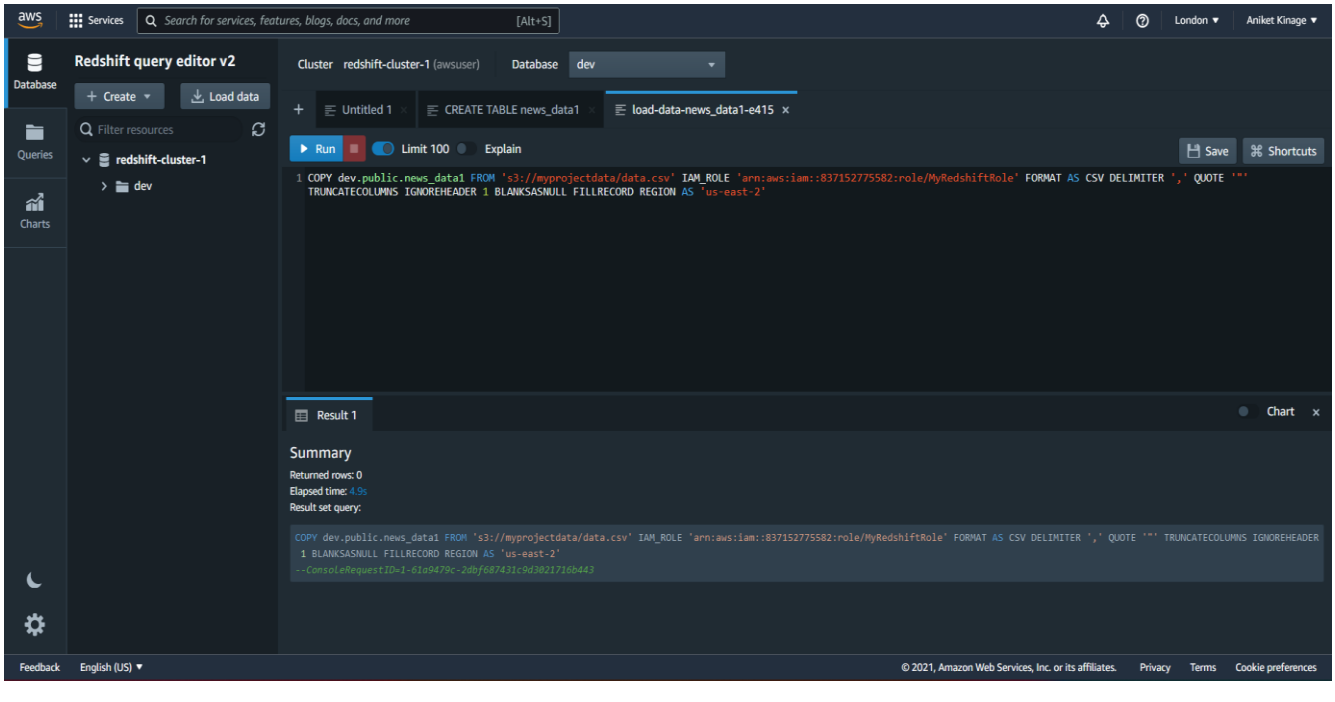
Step-9) Load the query and run the command. Table named “news_data1” is created.



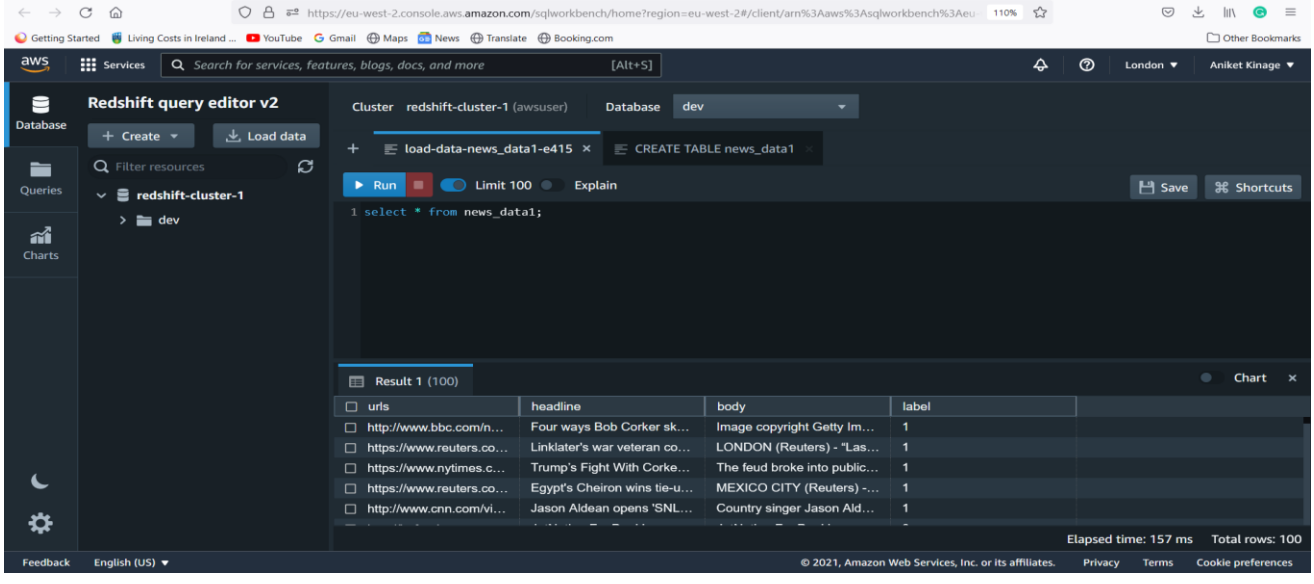
Step-10) Select the S3 data source and the associated IAM role. Select the schema as public and your target table and load the data.



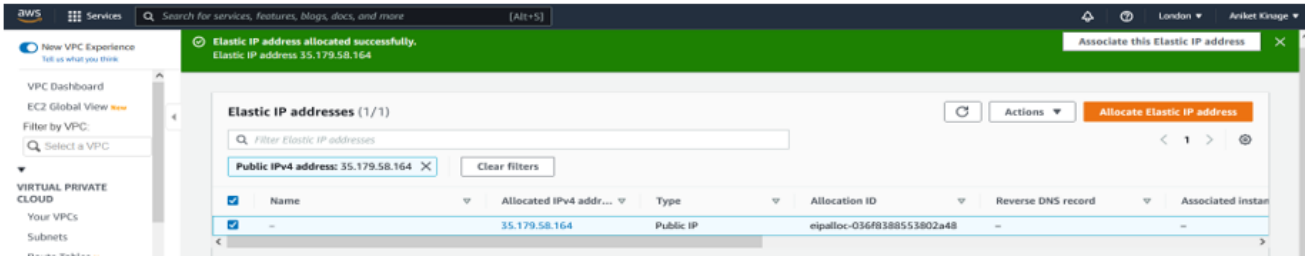
Step-11) Run the load query.



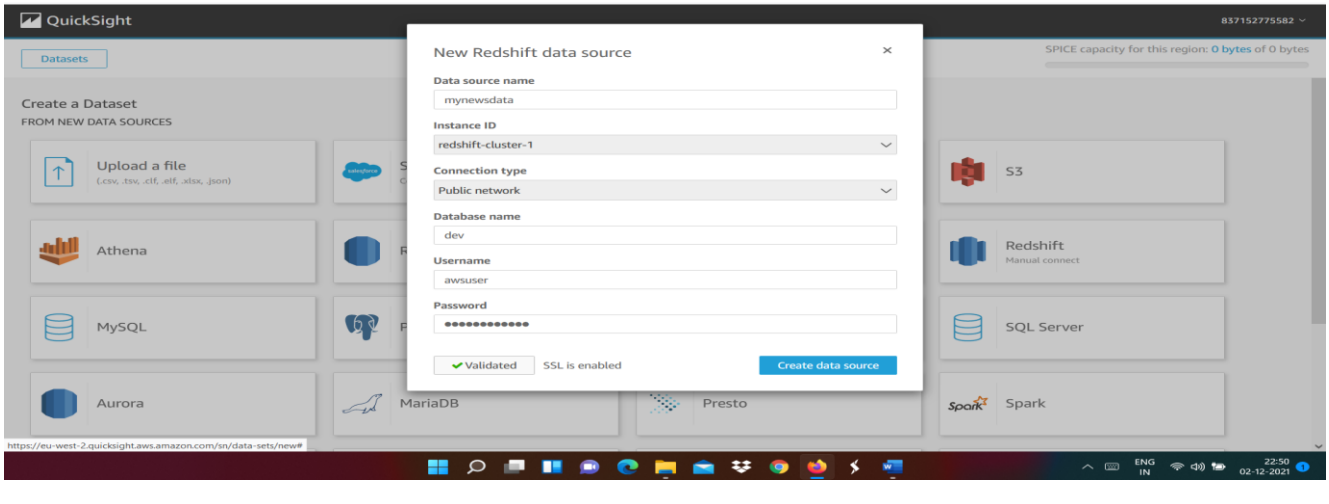
Step-12) Run some queries to ensure that your data is loaded properly in Redshift.



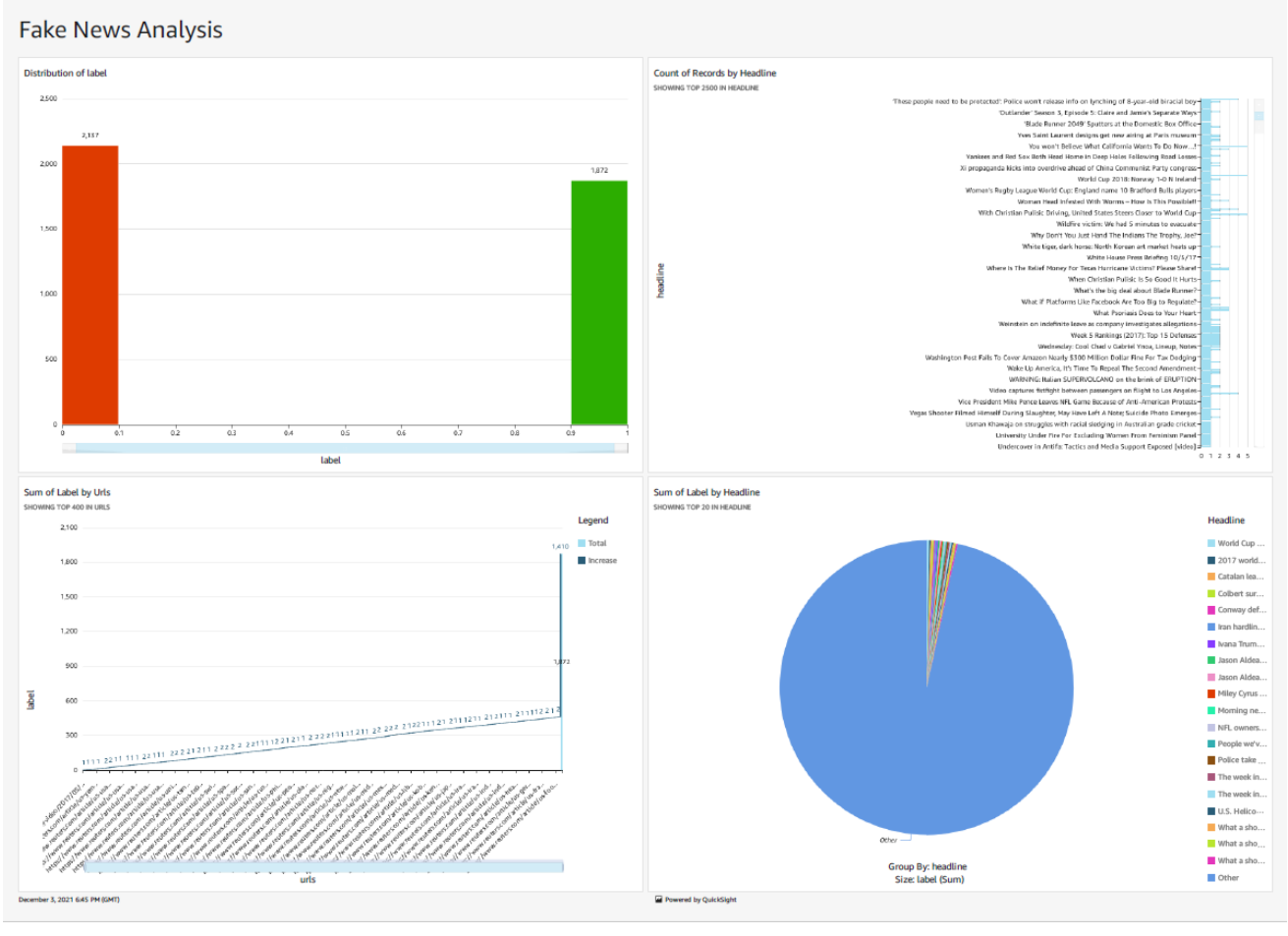
Step-13) Create the Elastic IP and associate it with the cluster.



Step-14) Connect the Amazon Quicksight with the redshift cluster. (Ensure that Quicksight and the Redshift Cluster are in the same region). Validate the connection and create the data source by choosing the appropriate table.



Step-15) Create the dashboard for visualising the fake news data using AWS Quicksight.



Chapter 5- Conclusion

.

Amazon Redshift's low cost, high performance, and ease of use expand data warehousing use cases beyond traditional business data warehousing and into big data and software-as-a-service applications with embedded analytics. Amazon Redshift clusters can be provisioned in minutes, allowing customers to get started with no commitments and scale up to a petabyte scale cluster. Using Redshift’s scaling capabilities we tried to analyse the fake news dataset effectively.

The main challenge in this project was to collect the data efficiently and processing it further before storing it in the data warehouse. The data coming from different locations can be in various forms, it can contain missing values or the data might not be relevant. Also we have to ensure that the data is properly loaded in the data warehouse. While creating the table in redshift, we have to ensure that we create the same table attributes as present in the dataset so that it gets aligned properly.

The important learning from this project was to build a complete data warehouse solution from scratch and analysing it for better insights. Also, learned how to store data in S3 and then integrating it to Redshift using SQL queries. Also used AWS Quicksight for making impactful visuals for analysis.