

# **Big Data Engineering and Applications**

In this document, we will explore the step-by-step process of uploading data to an AWS S3 bucket and subsequently analyzing it within an AWS Redshift cluster.

Accordingly, following topics are in discussion:

S3 bucket creation, Upload a CSV to S3 bucket, best practises in capacity utilisation and money saving, AWS Redshift cluster creation, Load data to Redshift cluster, SQL query processing and data analysis

## **Experiment II : Streamed data in Experiment I in AWS Redshift**

The twitter data that have been streamed in Experiment I has been transferred to AWS cloud platform for further analysis. This document illustrates the process of data storing steps in S3 bucket in AWS and analysis same on Amazon Redshift by SQL queries.

Before transferring to original data to S3, following steps have been performed to save the cost in AWS data analysis by reducing the volume of data transfer and fast processing with minimal errors.

1. Only the most specific columns that is necessary for the analysis of the questions in the Experiment III has been transferred to the S3 bucket. Thus, reduce the volume of data transferring to the S3 bucket and improve the efficiency of capacity utilization by saving money.
2. Data in the location field has been cleaned in python environment as to reduce the complexity in that field and its volume. As such the new column was created as country to store this cleaned data in location field. Thus, the null value rows have been dropped, emoji removed and mapped the location with country name for easy simple analysis. As such, the volume of the data file that is transferred to the AWS S3 bucket has been reduced.
3. Node type selected as dc2.large and number of nodes set as 1 in creation of Redshift cluster to minimize the AWS billing expenditure and gain value for money.
4. Both, the S3 bucket and the cluster region has been selected as one region to save money spent on the AWS platform.
5. The time duration spent on querying data has significantly reduced by uploading a well-structured csv file for analysis.

# Storing Data in Amazon S3

## Step 1: Starting the lab

Lab has been started by clicking the button 'Start Lab' to launch the lab and AWS button was selected to open the AWS Management Console in a new browser tab.

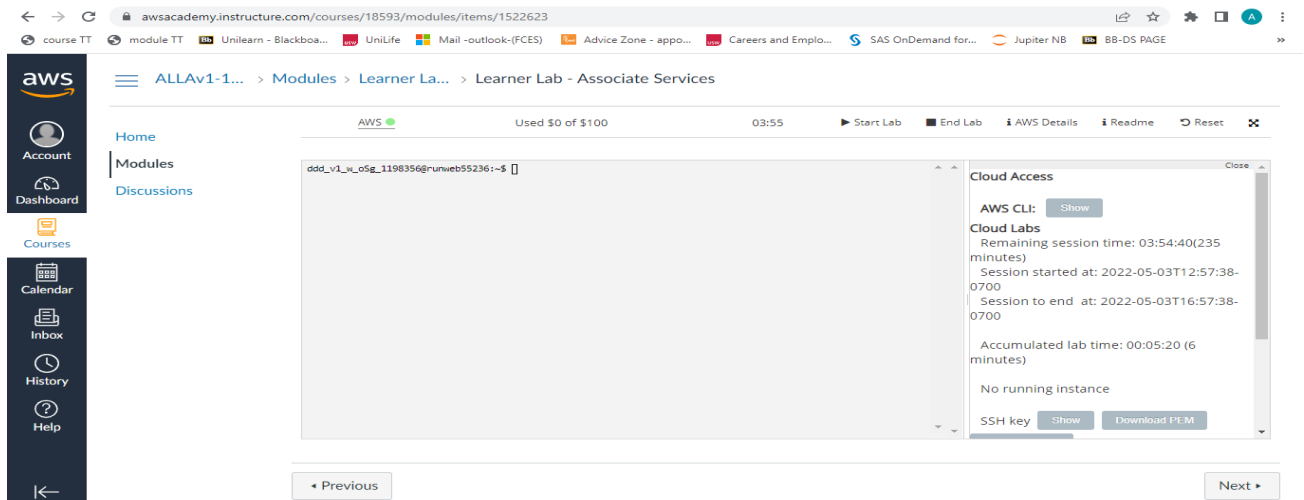


Fig. 1: Starting the lab

## Step 2: Creating the S3 bucket

Selected the 'Services' at the Services menu in the AWS Management Console and select 'S3'. Then, selected 'Create bucket'. The bucket name has assigned as 'adee30034311' as in Fig.2 below.

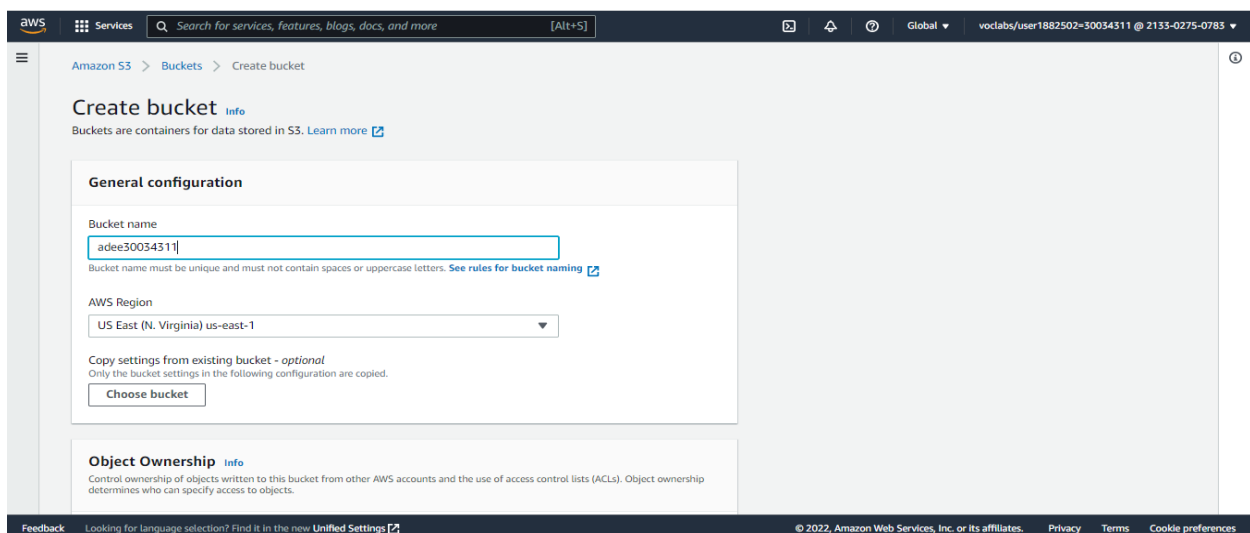


Fig 2: Enter a bucket name as 'adee30034311'

### Step 3: Storing data to the S3 bucket

Once the bucket has been successfully generated in the S3 console (Fig: 3), data has been stored for analysis. As such, the streaming data csv file named as 'tweets\_AWS\_Clean' was uploaded to the S3 bucket that has created in the previous task. Later, 'upload' was selected in the AWS S3 console (Fig: 3 and 4) to execute the process.

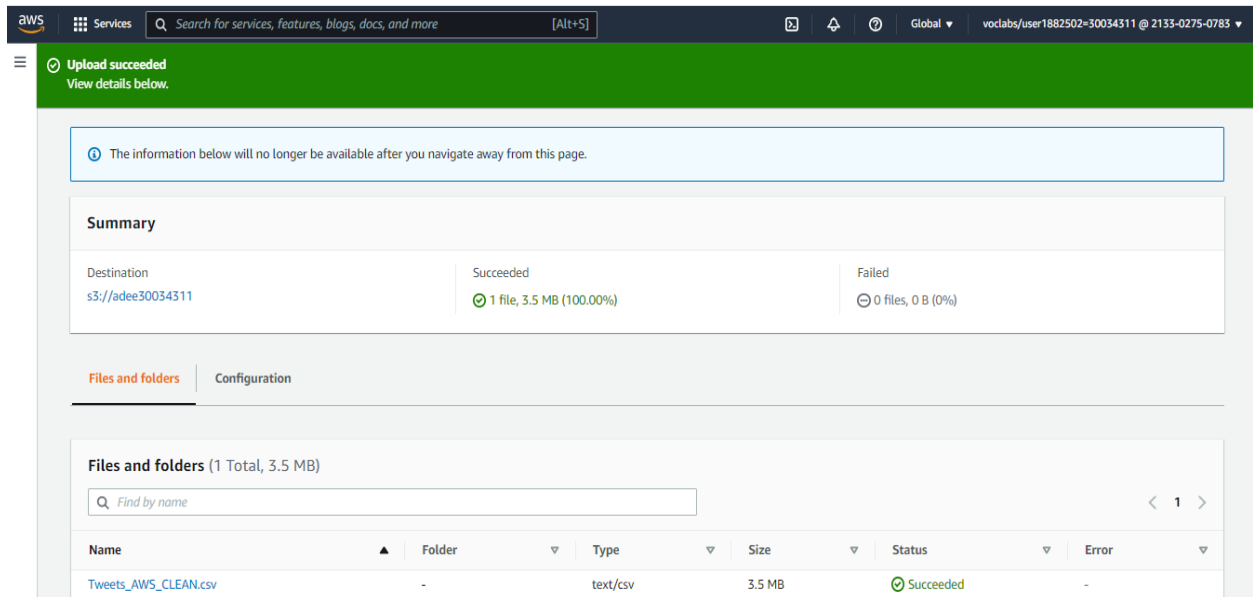


Fig. 3: Bucket creation

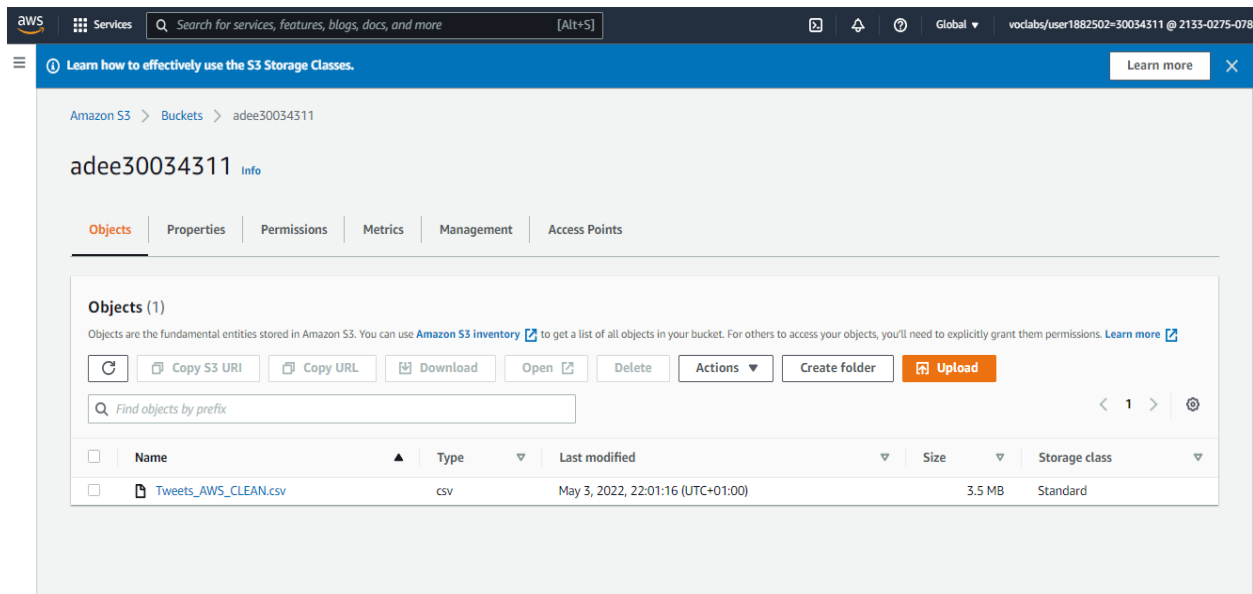
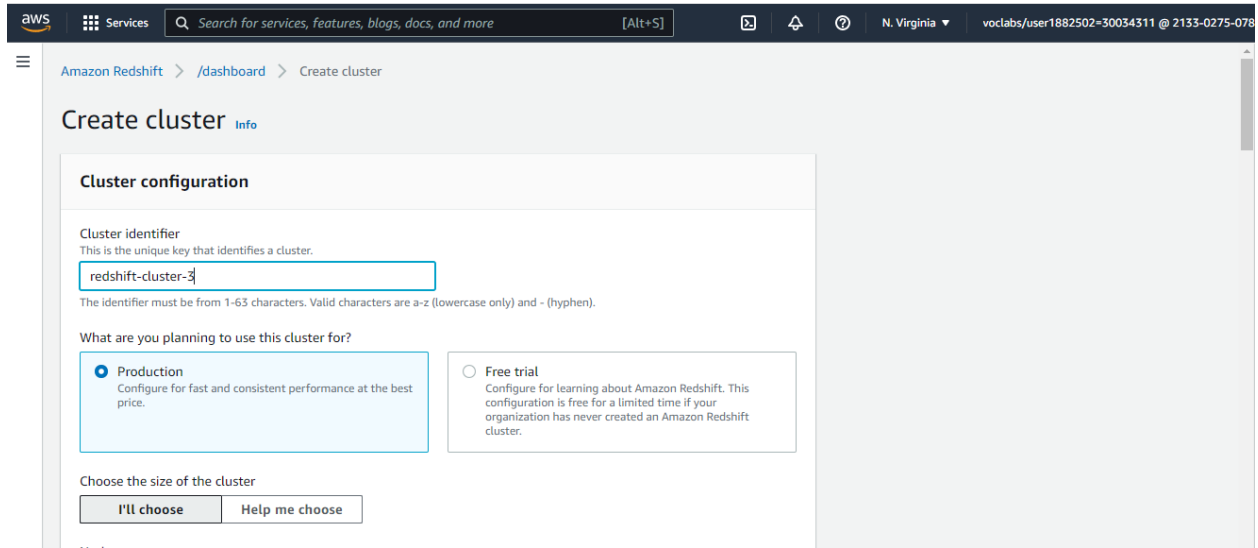


Fig: 4: CSV Data file uploaded to S3 bucket

## Creating a Redshift Cluster

In the AWS Management console, select 'Amazon Redshift'. Then, the cluster name has been updated as 'redshift-cluster-3' in the Cluster Identifier field in the management console (Fig. 5)



aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia voclabs/user1882502-30034311 @ 2133-0275-078

Amazon Redshift > /dashboard > Create cluster

### Create cluster [Info](#)

#### Cluster configuration

**Cluster identifier**  
This is the unique key that identifies a cluster.

redshift-cluster-3

The identifier must be from 1-63 characters. Valid characters are a-z (lowercase only) and - (hyphen).

**What are you planning to use this cluster for?**

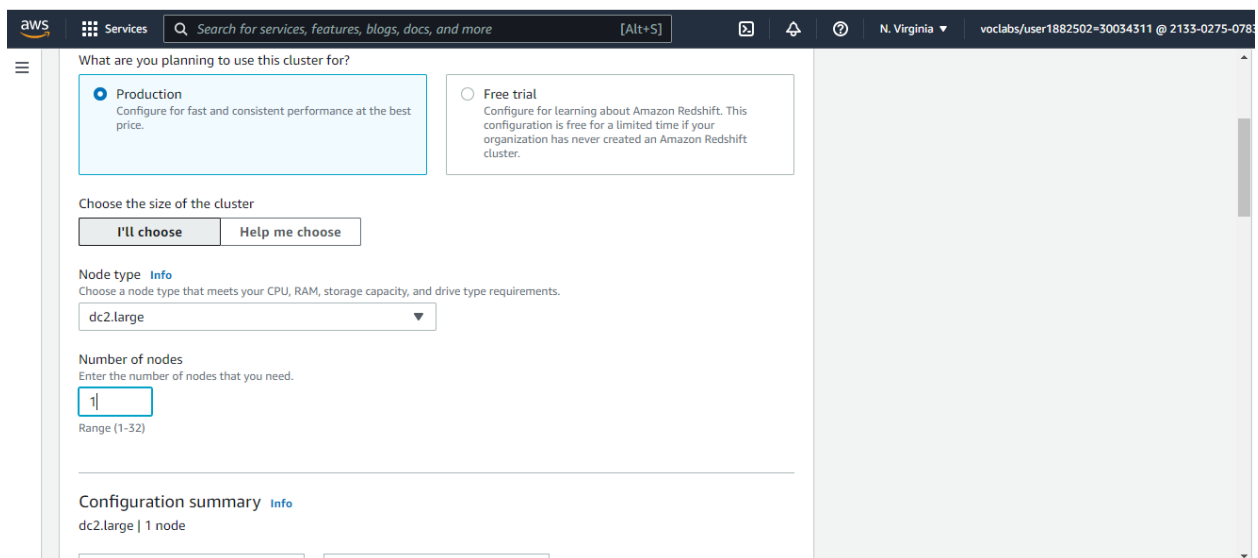
☒ **Production**  
Configure for fast and consistent performance at the best price.

☐ **Free trial**  
Configure for learning about Amazon Redshift. This configuration is free for a limited time if your organization has never created an Amazon Redshift cluster.

**Choose the size of the cluster**

Fig. 5: Cluster configuration

Later, the node type was selected as 'dc2.large' and the 'Number of nodes' set as 1 (Fig.6). This saves the billing cost of AWS platform.



aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia voclabs/user1882502-30034311 @ 2133-0275-078

What are you planning to use this cluster for?

☒ **Production**  
Configure for fast and consistent performance at the best price.

☐ **Free trial**  
Configure for learning about Amazon Redshift. This configuration is free for a limited time if your organization has never created an Amazon Redshift cluster.

**Choose the size of the cluster**

**Node type** [Info](#)  
Choose a node type that meets your CPU, RAM, storage capacity, and drive type requirements.

dc2.large

**Number of nodes**  
Enter the number of nodes that you need.

1

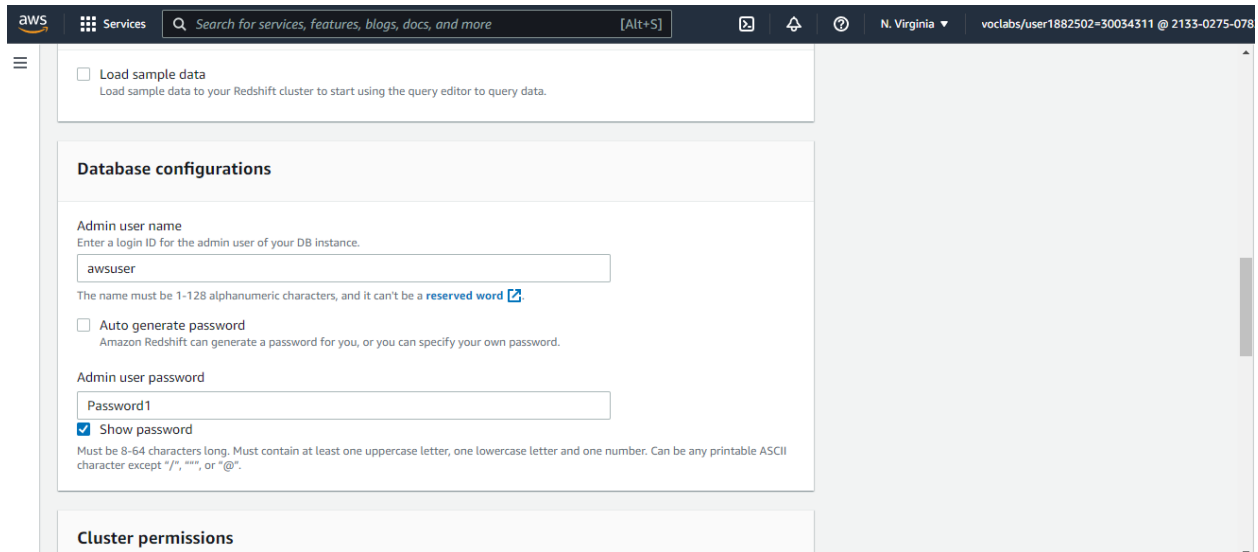
Range (1-32)

---

**Configuration summary** [Info](#)  
dc2.large | 1 node

Fig. 6: node type was selected as dc2.large and the Number of nodes set as 1

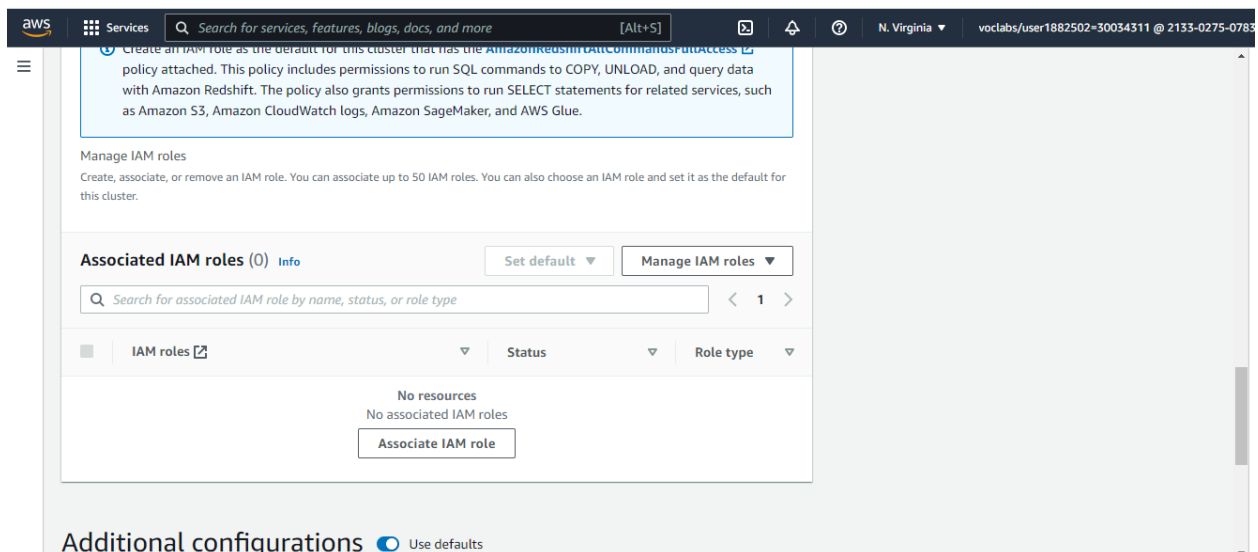
Admin user name was set as 'AWS user' and the password was assigned as 'Password1' in the database configuration section, as shown in the Fig. 7 below.



The screenshot shows the AWS Redshift console interface. At the top, there's a navigation bar with the AWS logo, 'Services' link, a search bar, and user information. The main content area is divided into sections. The 'Database configurations' section is active, showing fields for 'Admin user name' (set to 'awsuser') and 'Admin user password' (set to 'Password1'). The 'Show password' checkbox is checked. Below this is the 'Cluster permissions' section. The 'Load sample data' checkbox is unchecked.

Fig. 7: Data base configuration

Then, selected the 'Associate IAM role' as shown in the Fig.8. A new window as in Fig. 9 has appeared and 'MyRedShiftRole' has set as the Associate IAM role in that window. Then, selected 'Create cluster' as in Fig.10 below.



The screenshot shows the AWS Redshift console interface. The 'Associate IAM role' section is active, showing a list of associated IAM roles. The 'Associated IAM roles (0)' section is empty, and the 'Associate IAM role' button is visible. The 'Additional configurations' section is at the bottom, with the 'Use defaults' checkbox checked.

Fig. 8: Associate IAM role

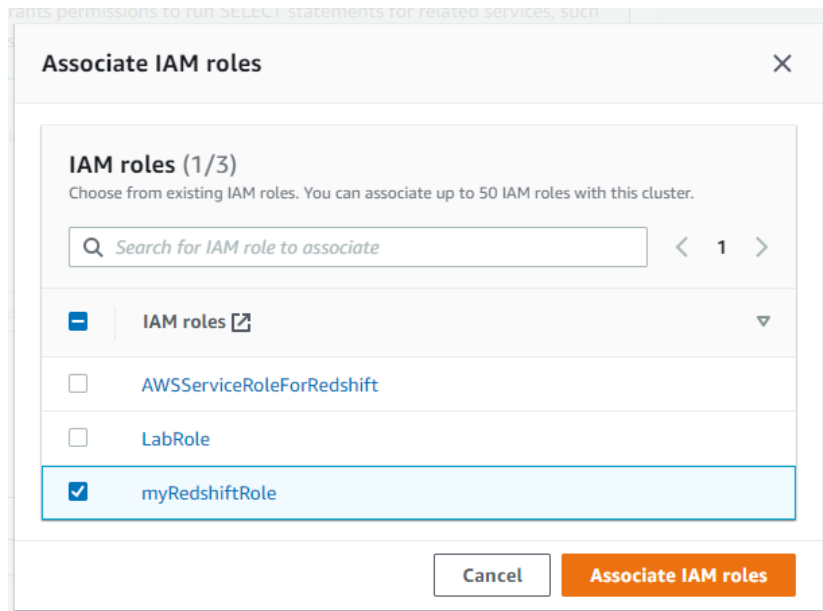


Fig. 9. MyRedShiftRole has set as the Associate IAM role

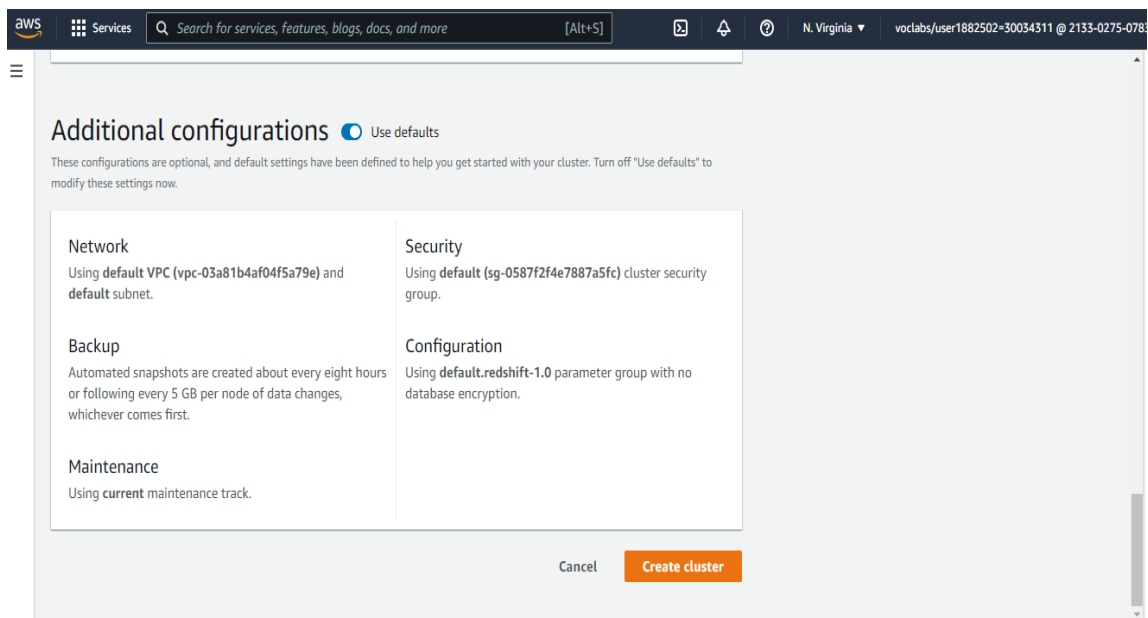


Fig.10. Create cluster

Once the cluster has been successfully created, the 'cluster status' was updated as 'available' in the Fig. 11 as illustrated below.

The screenshot displays the AWS Redshift console interface. At the top, there's a navigation bar with the AWS logo, 'Services' link, a search bar, and user information for 'N. Virginia' and 'voclabs/user1882502=30034311 @ 2133-0275-078'. Below the navigation bar, there's a 'Cluster' section with a 'Cluster identifier' dropdown, 'Copy JDBC URL' and 'Copy ODBC URL' buttons, and a 'Driver' dropdown set to 'JDBC 4.2 without AWS SDK (.jar)' with a 'Download driver' button. The main section is titled 'Clusters (3) Info' and includes a 'Create cluster' button. Below this is a table listing three clusters, all with a status of 'Available'.

Cluster	Cluster namespace	Status	Storage capacity us...	CPU utilization	Snapshots	Notificati...
redshift-cluster-1 dc2.large   1 node   160 GB	7287809d-d7d3-43f6-...	Available	< 1%	9%	-	
redshift-cluster-2 ra3.4xlarge   2 nodes   256 TB	e07bcd9b-0701-4e07-...	Available	< 1%	2%	-	
redshift-cluster-3 dc2.large   1 node   160 GB	c4e49972-d64a-43f1-...	Available			-	

Fig.11. Cluster Status as available



## Query data in the Redshift Cluster

‘Query data’ in the management console has been selected to start the data query as in Fig. 12.

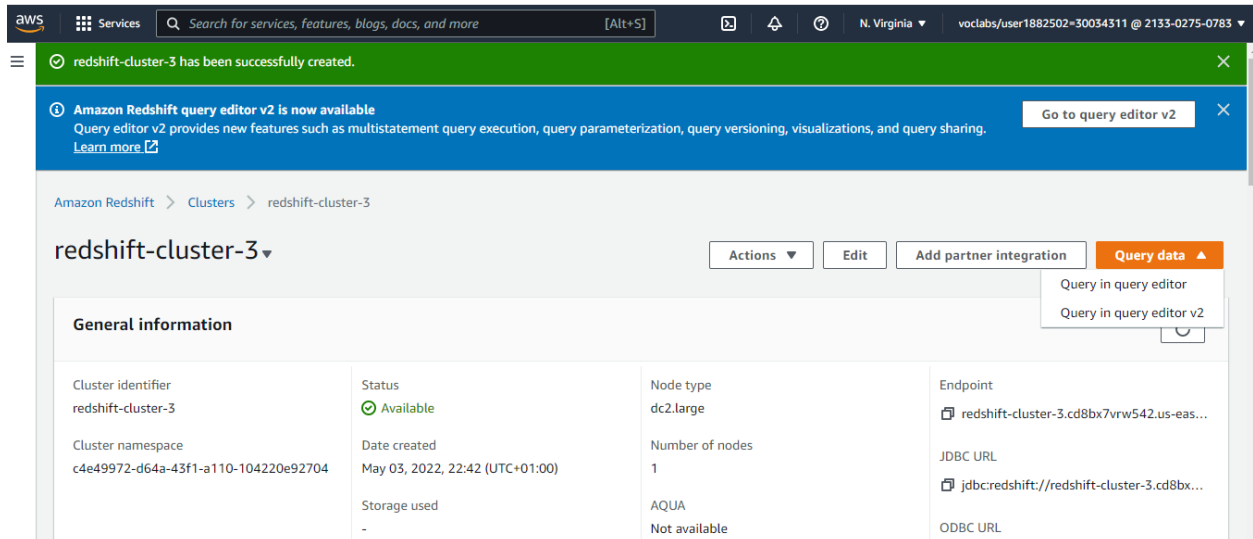


Fig.12. Start query data

Then, a new query window has been appeared as in Fig. 13. Then, select ‘create’ and choose ‘Table’ in the popup window to create a new table in the dev database.

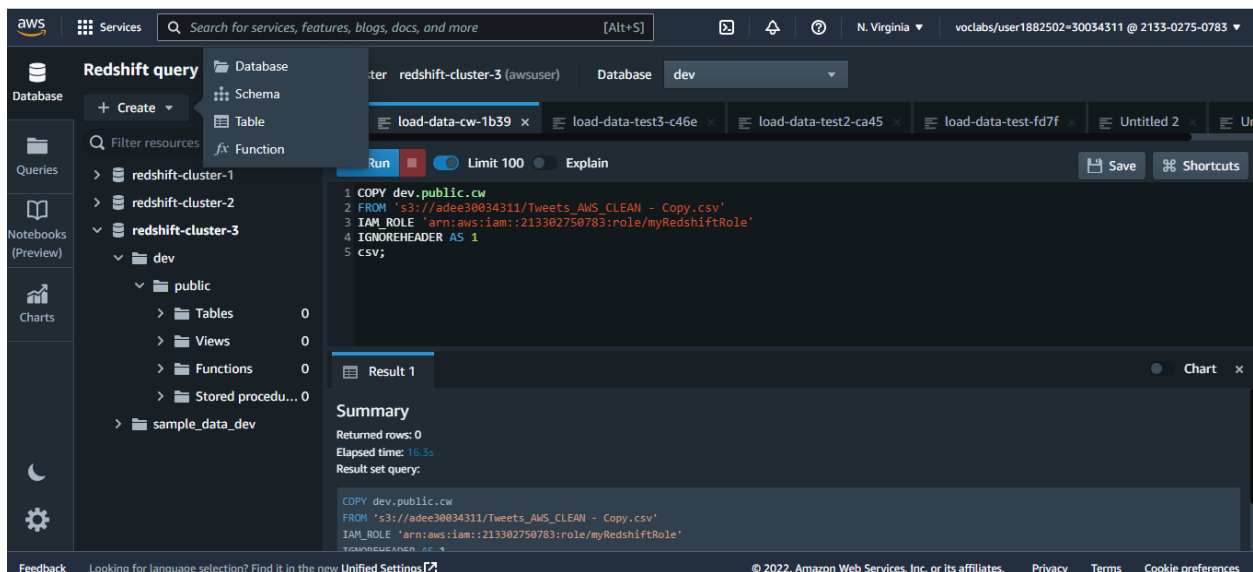


Fig.13. Create a new table

From the 'Schema' dropdown list, choose the 'public schema', assigned a table name and browse the required file.

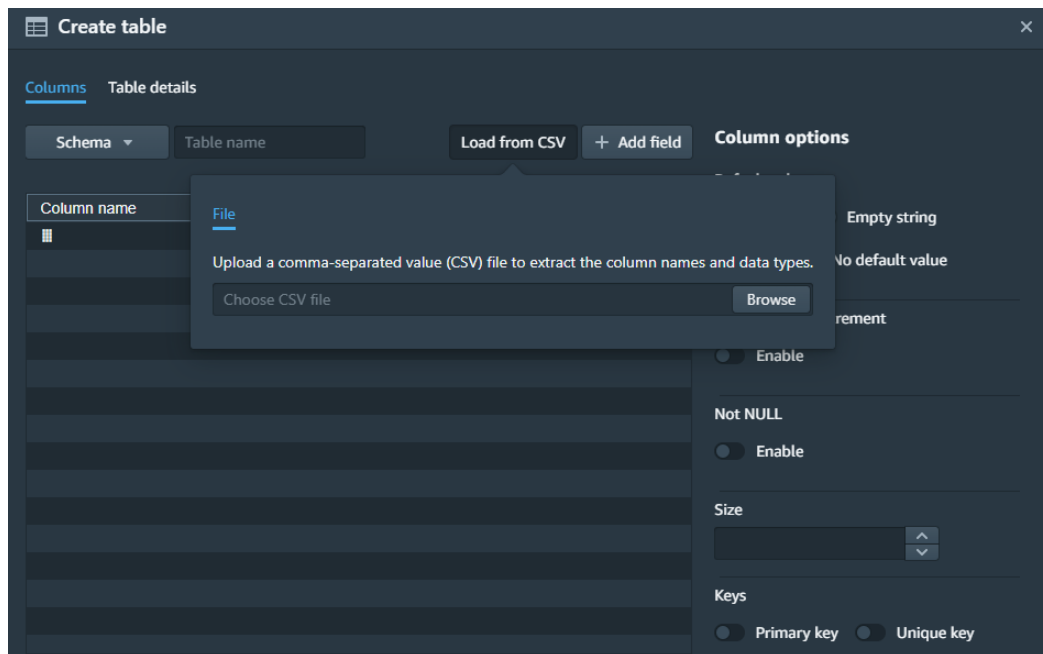


Fig.13. Create a new table

Once the data being uploaded, the window will appear as in the Fig. 14 and modify the data types as required. Later, the table name was assigned as 'tweet\_table1' and then, select 'Create table'.

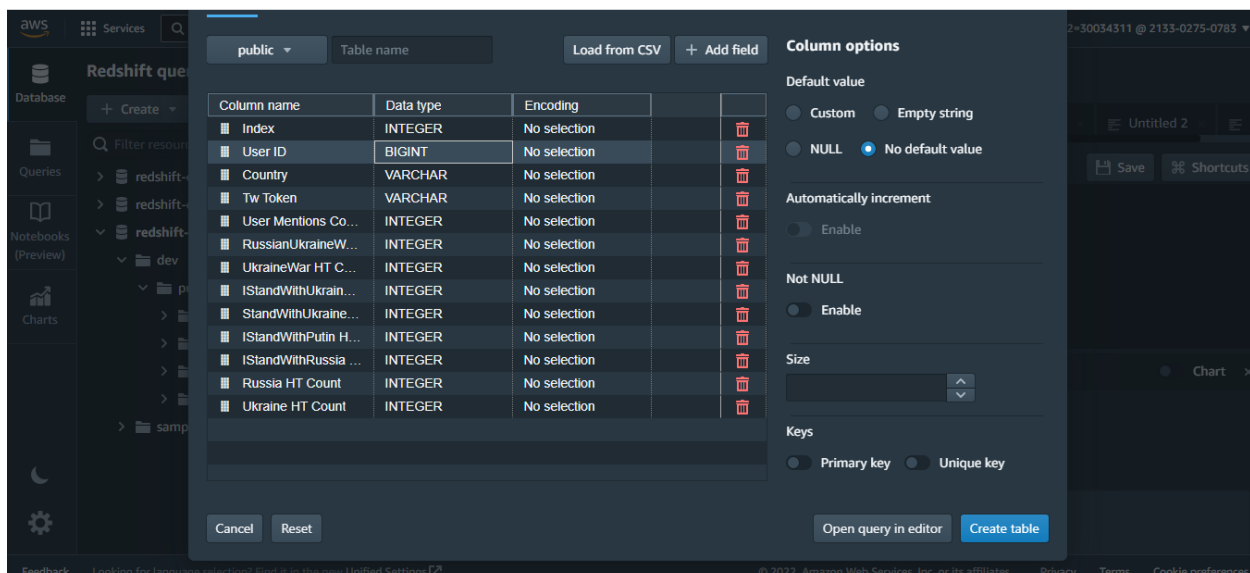


Fig. 14. Data table fields

Once the table has been created, choose the 'upload data' in the redshift management console. Select, the 'IAM role', set the schema as 'public' and select the table name as 'tweet\_table1'.

Then, select for 'Browse S3' (Fig.15) which route to a new window as illustrated in the Fig. 16 and select the csv file as in the Fig.17.

**Load data**

**Data source**

S3 URI: s3://adee30034311/Tweets\_AWS\_CLEAN.csv Browse S3

us-east-1 This file is a manifest.

IAM role: arn:aws:siam::213302750783:role/myRedshiftRole Refresh

File format: CSV File options No compression

Advanced settings: Data conversion parameters Load operations

**Target table**

dev public tweet\_table1

Column mapping ×

Cancel Load data

Fig. 15. To load the data from Amazon S3 to Redshift

**S3 buckets**

**Buckets**

Name	Creation date
adee30034311	May 03, 2022, 21:56:31 (UTC+01:00)

1 to 1 of 1 « < Page 1 of 1 > »

Cancel Choose

Fig. 16. Select S3 bucket

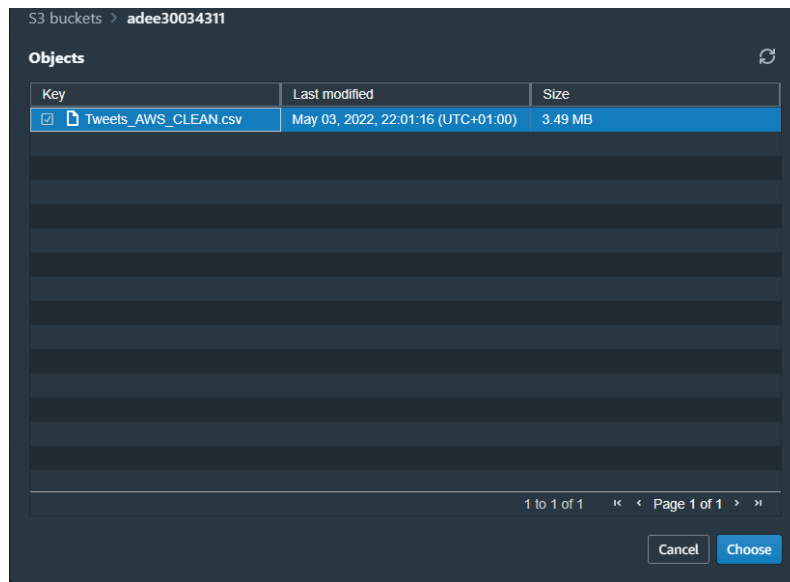


Fig. 17. Select csv file

Finally, the query window will appear and the code mentioned below has been executed to load the data to the query window. The data has been loaded into Amazon Redshift cluster, for writing SQL queries (Fig.18)

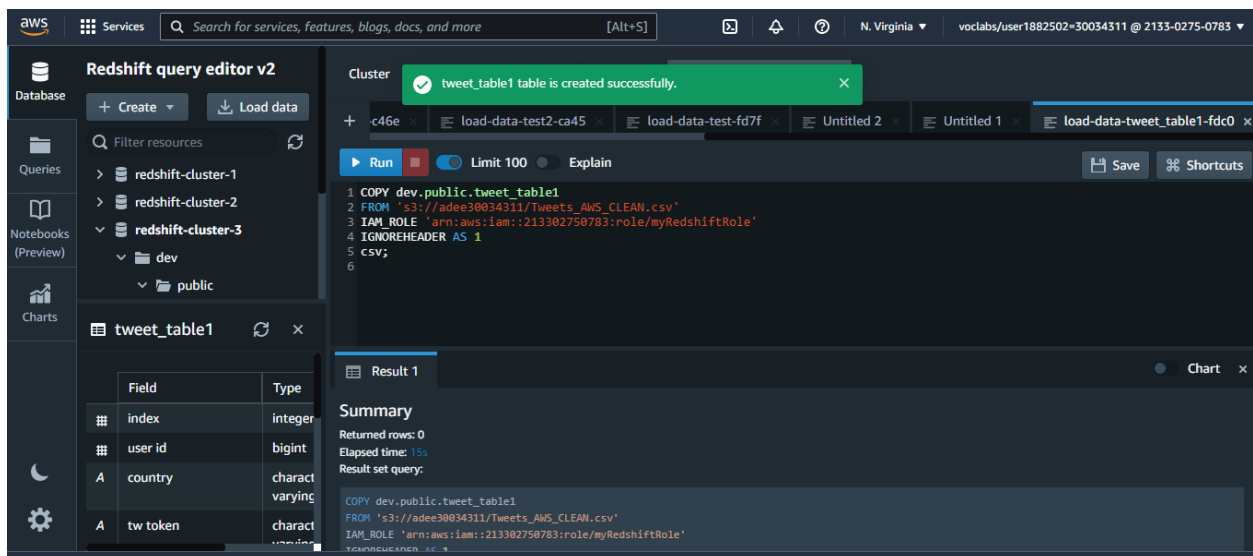
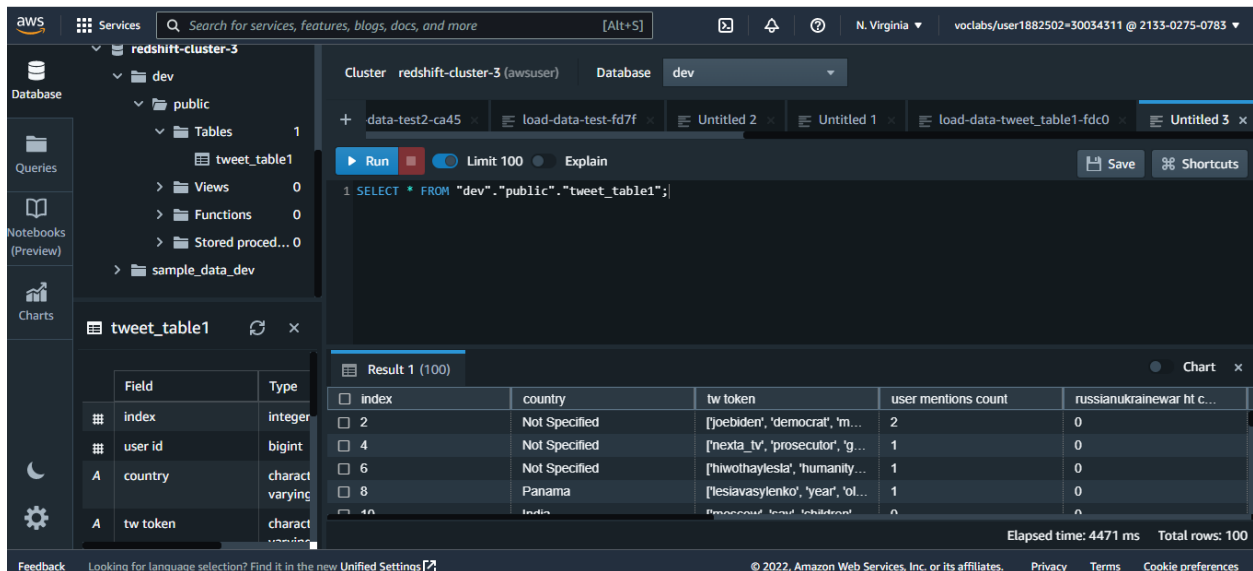


Fig. 18. data loaded in to Amazon Redshift cluster, for SQL

## Sample code;

```
COPY dev.public.tweet_table1
FROM 's3://adee30034311/Tweets_AWS_CLEAN.csv'
IAM_ROLE 'arn:aws:iam::213302750783:role/myRedshiftRole'
IGNOREHEADER AS 1
CSV;
```

The below output illustrate (Fig.19), that the stream data file (tweet\_table1) has been successfully launched in the RedShift management console for successful query execution.



The screenshot shows the Amazon Redshift console interface. On the left, the navigation pane shows the database structure: **redshift-cluster-3** > **dev** > **public** > **Tables** > **tweet\_table1**. The main panel displays the SQL query editor with the query: `1 SELECT * FROM "dev"."public"."tweet_table1";`. Below the query editor, the results are shown in a table format. The table has 5 columns: **index**, **country**, **tw token**, **user mentions count**, and **russianukrainewar ht c...**. The first row shows an index of 2, country 'Not Specified', tw token '[joebiden', 'democrat', 'm...', user mentions count 2, and russianukrainewar ht c... 0. The second row shows an index of 4, country 'Not Specified', tw token '[nexta\_tv', 'prosecutor', 'g...', user mentions count 1, and russianukrainewar ht c... 0. The third row shows an index of 6, country 'Not Specified', tw token '[hiwothaylesla', 'humanity...', user mentions count 1, and russianukrainewar ht c... 0. The fourth row shows an index of 8, country 'Panama', tw token '[lesiavasylenko', 'year', 'ol...', user mentions count 1, and russianukrainewar ht c... 0. The fifth row shows an index of 10, country 'Not Specified', tw token '[lesiavasylenko', 'year', 'ol...', user mentions count 1, and russianukrainewar ht c... 0. The bottom status bar indicates 'Elapsed time: 4471 ms' and 'Total rows: 100'.

index	country	tw token	user mentions count	russianukrainewar ht c...
2	Not Specified	[joebiden', 'democrat', 'm...	2	0
4	Not Specified	[nexta_tv', 'prosecutor', 'g...	1	0
6	Not Specified	[hiwothaylesla', 'humanity...	1	0
8	Panama	[lesiavasylenko', 'year', 'ol...	1	0
10	Not Specified	[lesiavasylenko', 'year', 'ol...	1	0

Fig. 19. data loaded in to Amazon Redshift cluster, for SQL

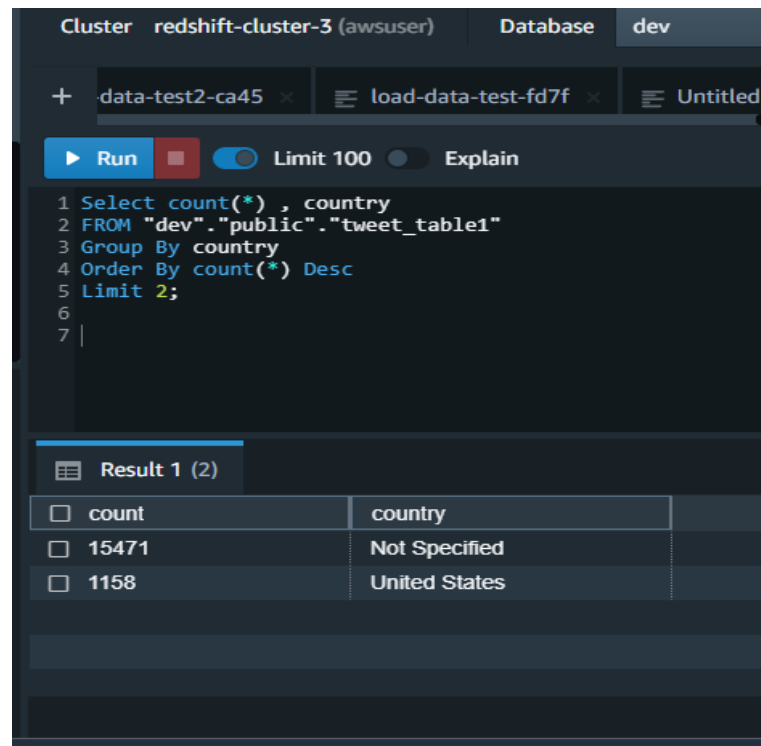
As mentioned above, the process of data loading to the AWS Redshift has been completed and management console has been set up for the SQL query execution following commands has been performed.

## Query the data that have streamed in Experiment II in AWS Redshift to find the following:

The twitter data that have been streamed in Experiment II has been transferred to AWS cloud platform for further analysis. In the above of this document illustrates the process of data storing S3 bucket in AWS and analysis same on Amazon Redshift by SQL queries. Thus, following queries have been processed.

### 1. The country which has authored the most tweets:

Below output in Fi. 20 illustrate the SQL query and its output for the country which has authored the most tweets. The output illustrate, majority tweets user locations are different to the country and except that row, country which has authored the most tweets is United States.



The screenshot shows the AWS Redshift console interface. At the top, the cluster is identified as 'redshift-cluster-3 (awsuser)' and the database is 'dev'. Below this, there are tabs for 'data-test2-ca45', 'load-data-test-fd7f', and 'Untitled 2'. A 'Run' button is visible, along with a 'Limit 100' toggle and an 'Explain' button. The SQL query is displayed in a text area:

```
1 Select count(*) , country
2 FROM "dev"."public"."tweet_table1"
3 Group By country
4 Order By count(*) Desc
5 Limit 2;
6
7 |
```

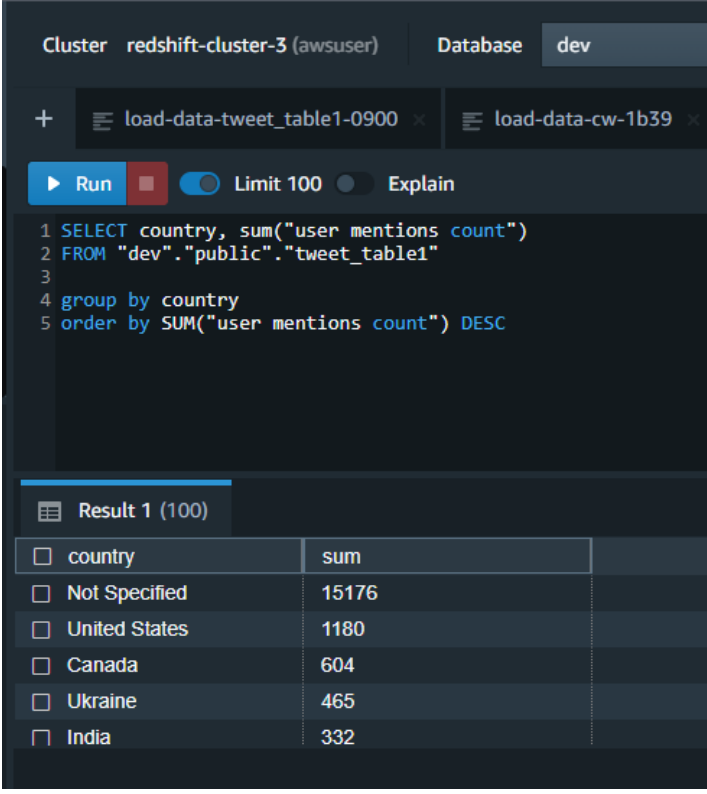
Below the query, the results are shown under the heading 'Result 1 (2)'. The results are displayed in a table with two columns: 'count' and 'country'.

count	country
15471	Not Specified
1158	United States

Fig. 20: Query for country which has authored the most tweets

#### 4 Total number of user mentions in tweets from each country respectively

Fig 21 illustrates the SQL query and its output for the total number of user mentions in tweets from each country respectively. The output illustrates, Total number of user mentions in tweets are from the column of the country has not accurately defined. And rest of the countries are listed below in the descending order.



The screenshot shows the AWS Redshift console interface. At the top, the cluster is identified as 'redshift-cluster-3 (awsuser)' and the database is 'dev'. Below this, there are tabs for 'load-data-tweet\_table1-0900' and 'load-data-cw-1b39'. A 'Run' button is visible, along with a 'Limit 100' toggle and an 'Explain' button. The SQL query is displayed in a text area:

```
1 SELECT country, sum("user mentions count")
2 FROM "dev"."public"."tweet_table1"
3
4 group by country
5 order by SUM("user mentions count") DESC
```

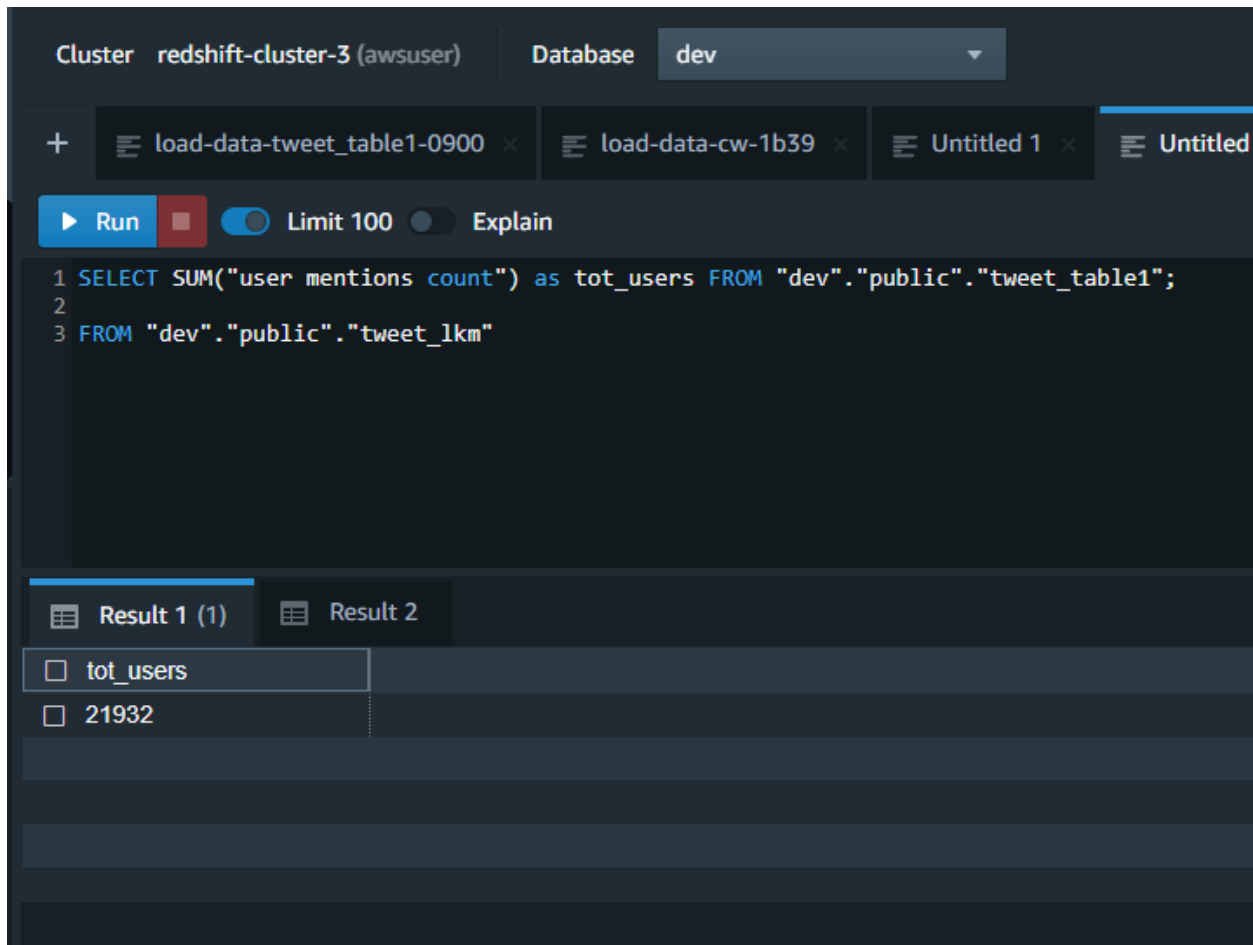
Below the query, the results are shown under the heading 'Result 1 (100)'. The results are displayed in a table with two columns: 'country' and 'sum'. The data is sorted in descending order of the sum of user mentions.

country	sum
Not Specified	15176
United States	1180
Canada	604
Ukraine	465
India	332

Fig. 21: Query for Total number of user mentions in tweets from each country respectively

## 5. Total number of user mentions in all tweets.

Fig. 22 illustrates the SQL query and its output Total number of user mentions in all tweets. The output illustrate, Total number of user mentions in all tweets as 21,932.



The screenshot shows the AWS Redshift console interface. At the top, the cluster is identified as 'redshift-cluster-3 (awsuser)' and the database is 'dev'. Below this, there are tabs for 'load-data-tweet\_table1-0900', 'load-data-cw-1b39', 'Untitled 1', and 'Untitled'. A 'Run' button is visible, along with a 'Limit 100' toggle and an 'Explain' button. The SQL query is displayed in a text area:

```
1 SELECT SUM("user mentions count") as tot_users FROM "dev"."public"."tweet_table1";  
2  
3 FROM "dev"."public"."tweet_lkm"
```

Below the query, the results are shown in a table. The first result is labeled 'Result 1 (1)' and contains two columns: 'tot\_users' and '21932'.

tot_users
21932

Fig. 22: Total number of user mentions in all tweets.



### 3. The most frequent hashtag/word mentioned in Experiment II found in all tweets

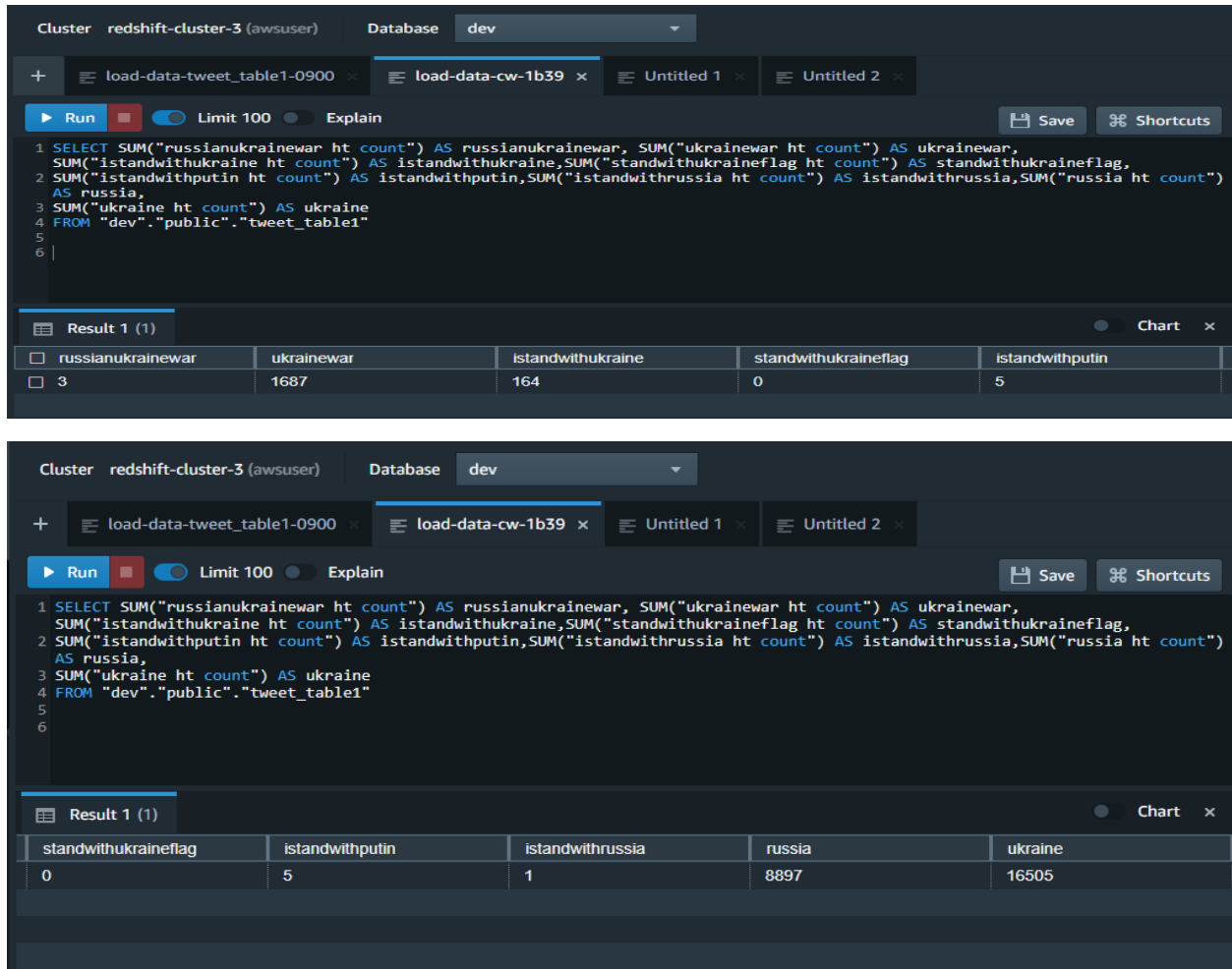


Fig. 23: Total number of user mentions in all tweets.

The above output illustrates the most frequent hashtag/word mentioned in Experiment II found in all tweets as Ukraine.