**Load and query data in the Amazon Redshift cluster**

**Aim**

The purpose of this experiment is to load data into the Amazon Redshift cluster and perform the following queries:

* the country which has authored the most tweets
* the most frequent hashtag/word mentioned in Experiment II found in tweets from each country
* the most frequent hashtag/word mentioned in Experiment II found in all tweets
* total number of user mentions in tweets from each country respectively, and
* total number of user mentions in all tweets.

**Introduction**

This experiment uses Amazon Web Services to load and query bigdata. Amazon Web Services offers a comprehensive and fully integrated portfolio of cloud computing services that assist us in developing, securing, and deploying big data applications.

Here we are uploading the output from experiment 2 (a comma separated file(CSV) ) to the AWS. Since our aim is to query data, we must use Amazon Redshift to load data. Amazon Redshift uses SQL to query data across different storage. For that first, we need to store data in the Amazon S3. Amazon Simple Storage Service (Amazon S3) is one of the most widely used data storage services.

**The steps to load CSV to amazon s3 are given below**.

1. Create a new s3 Bucket with the name ‘bdcw2nisha’ and region us-east1

On the AWS Management Console, on the Services menu, choose Services then choose S3.

Here we are using the user group ‘awsusers’ which by default has the ‘*AmazonS3FullAccess’* policy attached to it. This policy states that users in that group are allowed to take all actions for Amazon S3 on all resources.

Graphical user interface, text, application

Description automatically generated

Figure 1: S3 bucket creation

Graphical user interface, text, application, email

Description automatically generated

Figure 2: S3 bucket: bdcw2nisha

1. Upload file

Click on the newly created bucket and select upload. Upload the file ‘cleaned\_tweets’ from the local storage.

Graphical user interface, application, Word

Description automatically generated

Figure 3: Upload file from local drive

Graphical user interface, text, application

Description automatically generated

Figure 4: Uploaded cleaned\_tweets.csv file in the AWS s3 bucket

Now the file is saved to s3 bucket from your local drive.

**The steps to load data into Redshift are given below**.

1. Review the security group for accessing the Amazon Redshift console.

## You must be authenticated and authorised for Amazon Redshift access before you can access the Amazon Redshift console. So make sure the policy AmazonS3ReadOnlyAccess is attached to the ‘myRedshiftRole’. This policy authorises Amazon Redshift to obtain a list of items from Amazon S3 and retrieve those items.

## Graphical user interface, text, application, email Description automatically generated

Figure 5: myRedshiftRole policies attached

1. Create and configure an Amazon Redshift cluster

On the AWS Management Console, on the Services, choose Amazon Redshift, and choose ‘create cluster’.

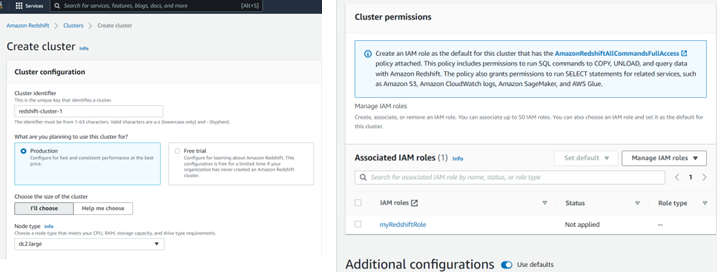


Figure 6 Cluster(redshift-cluster-1) Creation

1. Check the status of newly created cluster.

The cluster is ready when the status becomes green and shows ‘Available’,

Graphical user interface, text, application, email

Description automatically generated

Figure 7 Created cluster

1. Create a security group for the cluster

The virtual private cloud (VPC) that hosts your Amazon Redshift cluster must be configured to allow traffic through port 5439. For that On the AWS Management Console, go to the **Services** and choose **EC2 and create a new security group** ‘Redshift Security Group’. This security group must have an inbound rule which has the port range 5439.

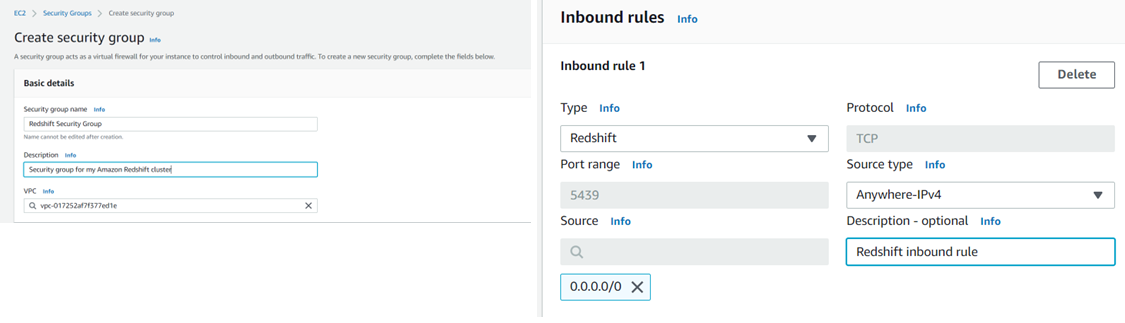


Figure 8 Security group creation

1. Configure redshift-cluster-1

Add the security group created in the previous step to the cluster.

Graphical user interface, application

Description automatically generated

Figure 9 Add security group to the cluster

Now the cluster creation is completed and we can load the data stored in the s3 bucket to redshift for querying.

**The steps to load data from s3 bucket to redshift are given below**.

1. Create the table.

Create a table ‘tweets’ with attributes exactly the same as the column names of the CSV file.

The SQL command for creating the table is given below.

create table tweets(

index\_col integer not null distkey sortkey ,

tweet\_text varchar(5000),

country varchar(100),

no\_user\_mentions integer,

ukraine integer,

russia integer,

ukrainewar integer,

istandwithukraine integer,

standwithukraineflag integer,

istandwithputin integer,

istandwithrussia integer,

russianukrainewar integer

);

**Output:**

Graphical user interface, text, application

Description automatically generated

Figure 10: Result of Create table tweets

### Load data from Amazon S3

Graphical user interface, text, application

Description automatically generated

Figure 11: Load data from amazon s3

The data from ‘cleaned\_tweets.csv’ is now stored in the redshift table ‘tweets’. Now we can perform SQL queries to get required information from this table. The answers are attached as CSV files with the report.

**Query data**

The following queries were executed as per the course work documentation.

**Q1:**

**Aim:**

To retrieve the country which has authored the most tweets

SQL Querry:

SELECT country, No\_of\_Tweets

FROM (

SELECT country, count(country) No\_of\_Tweets

from tweets

GROUP BY country

order by No\_of\_Tweets desc limit 1 );

**Output:**

Graphical user interface, text, application

Description automatically generated

Figure 12: Q1 output

**Q2.**

**Aim:**

To retrieve the most frequent hashtag/word mentioned in Experiment II found in tweets from each country

select country, frequency, case frequency when ukraine then 'Ukraine'

when russia then 'Russia'

when ukrainewar then 'Ukrainewar'

when istandwithukraine then 'Istandwithukraine'

when standwithukraineflag then 'Standwithukraineflag'

when istandwithputin then 'Istandwithputin'

when istandwithrussia then 'Istandwithrussia'

when russianukrainewar then ' russianukrainewar '

end As Frequent\_Hashtag

from (

select country, greatest(ukraine, russia, ukrainewar, istandwithukraine, standwithukraineflag, istandwithputin, istandwithrussia) as frequency

, ukraine, russia, ukrainewar, istandwithukraine, standwithukraineflag, istandwithputin, istandwithrussia, russianukrainewar

from (

select country

, sum(ukraine) as ukraine

, sum(russia) as russia

, sum(ukrainewar) as ukrainewar

, sum(istandwithukraine) as istandwithukraine

, sum(standwithukraineflag) as standwithukraineflag

, sum(istandwithputin) as istandwithputin

, sum(istandwithrussia) as istandwithrussia

, sum(russianukrainewar) as russianukrainewar

from tweets

group by country

)

)

**Output:**

Table

Description automatically generated with medium confidence

Figure 13: Q2 Output

**Q3.**

**Aim:**

To retrieve the most frequent hashtag/word mentioned in Experiment II found in all tweets

select frequency, case frequency when ukraine then 'Ukraine'

when russia then 'Russia'

when ukrainewar then 'Ukrainewar'

when istandwithukraine then 'Istandwithukraine'

when standwithukraineflag then 'Standwithukraineflag'

when istandwithputin then 'Istandwithputin'

when istandwithrussia then 'Istandwithrussia'

when russianukrainewar then ' russianukrainewar '

end As Frequent\_Hashtag

from (

select greatest(ukraine, russia, ukrainewar, istandwithukraine, standwithukraineflag, istandwithputin, istandwithrussia) as frequency

, ukraine, russia, ukrainewar, istandwithukraine, standwithukraineflag, istandwithputin, istandwithrussia, russianukrainewar

from (

select sum(ukraine) as ukraine

, sum(russia) as russia

, sum(ukrainewar) as ukrainewar

, sum(istandwithukraine) as istandwithukraine

, sum(standwithukraineflag) as standwithukraineflag

, sum(istandwithputin) as istandwithputin

, sum(istandwithrussia) as istandwithrussia

, sum(russianukrainewar) as russianukrainewar

from tweets

)

)

Output:

**Output:**

Graphical user interface, application, Teams

Description automatically generated

Figure 14: Q3 Output

**Q4.**

**Aim:**

To retrieve the total number of user mentions in tweets from each country respectively

SELECT country, sum(no\_user\_mentions) No\_user\_mentions

FROM tweets

group by country;

**Output:**

Graphical user interface, application

Description automatically generated

Figure 15: Q4 Output

**Q5.**

**Aim:**

To retrieve the total number of user mentions in all tweets

SELECT sum(no\_user\_mentions) Total\_User\_mentions

FROM tweets;

**Output:**

Graphical user interface, application, Teams

Description automatically generated

Figure 16: Q5 Output

**The amount used to perform this complete operation is given in fig 17.**

Graphical user interface, text, application

Description automatically generated

Figure 17: AWS cost

**Conclusion**

The output of experiment 2 was successfully uploaded to amazon s3 bucket and copied the contents of CSV file to the amazon redshift table. All the five tasks given in the CW documentation were performed using SQL commands and the outputs were collected. The results shows that the maximum number of tweets are from Unites States, Most of the countries (out of 161) frequently used the hashtag Ukraine while tweeting, and a total of 13130 user mentions were made during this period.