RedShift Lan Guide

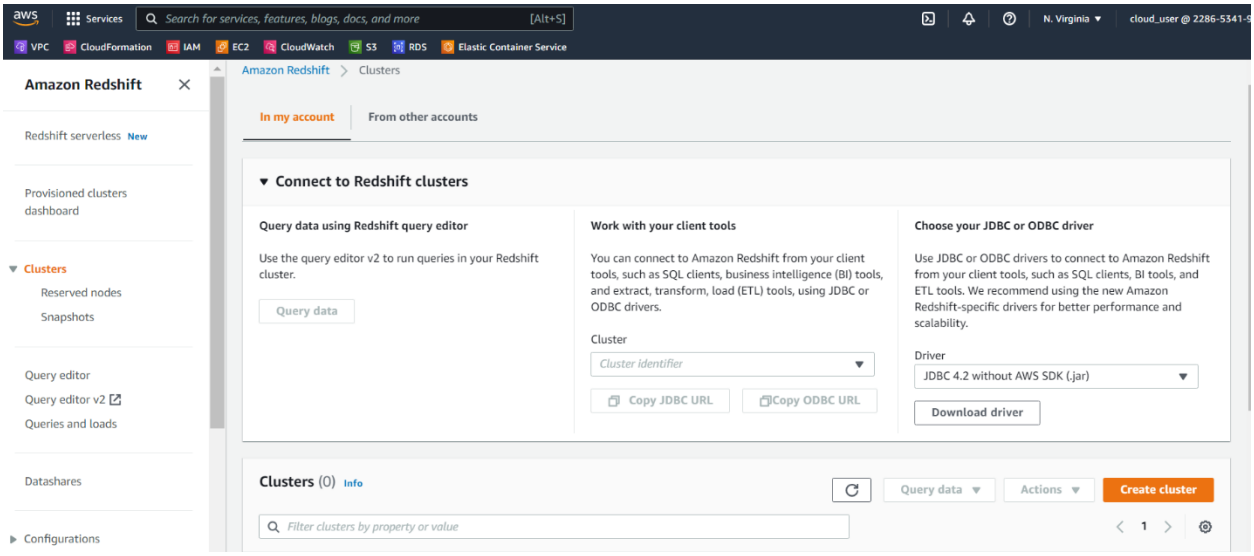
Steps Overview :

1. Create IAM Role (KarthikSpectrumRole) with permissions – S3ReadAccess and GlueConsoleFullAccess. Copy the role arn into a notepad for later reference.
    a. Go tot IAM console and Click on Create Role
    b. Choose AWS-Service > Redshift > and usecase as “Redshift - Customizable”
    c. Then to attache Permission -> search for
        i. AmazonS3ReadOnlyAccess
        ii. AWSGlueConsoleFullAccess
    d. Click on create Role
2. Create Redshift Cluster. Choose free trial. Attach IAM Role created in step1.
    a. In case of Production cluster -Role can be attached during the create process or after creating the cluster as well by going to properties section.
    b. In case of Trial Cluster - Role can be attached only after creating the cluster as well by going to properties section.
3. By Default DB name is **dev**, username is **awsuser**, setup your own password “DnAredshift1234”
4. Create a bucket – **“karthiksamplespectrum”**. Upload the provided files.

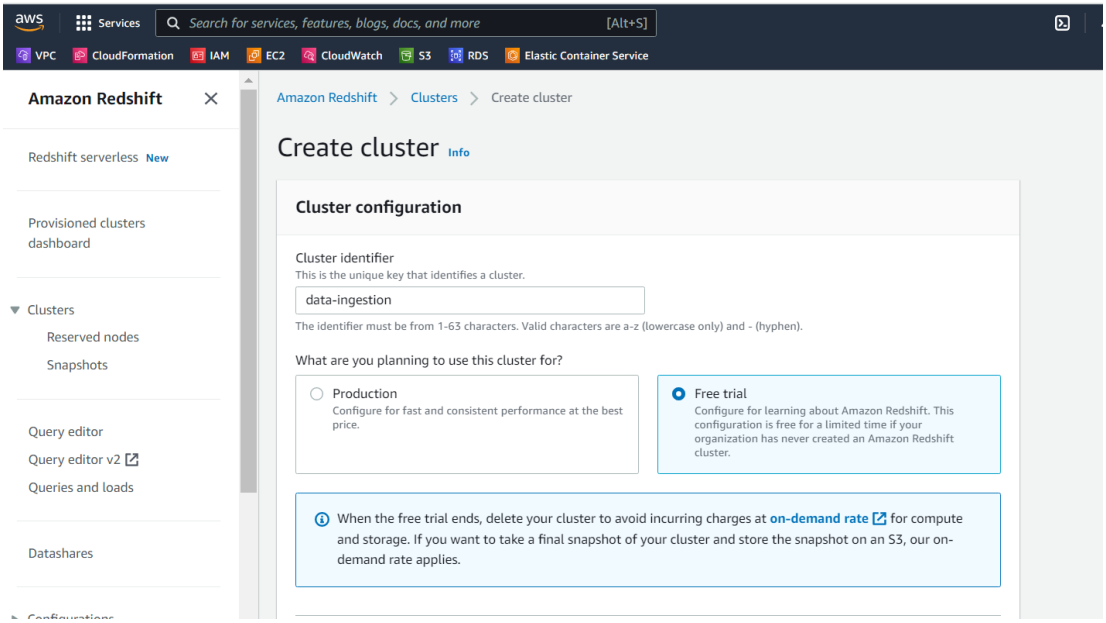
Follow the below snapshots to create the Redshift Cluster :

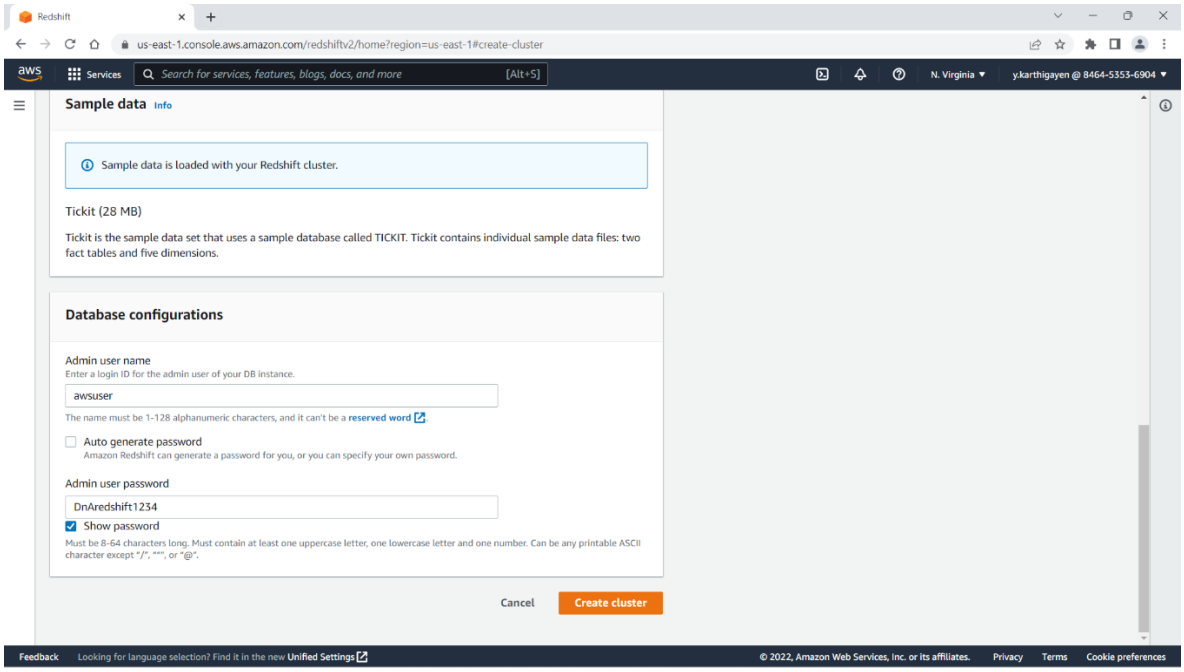
- Search for redshift service and open it

- Open clusters option on the left side panel and click on create cluster as shown below

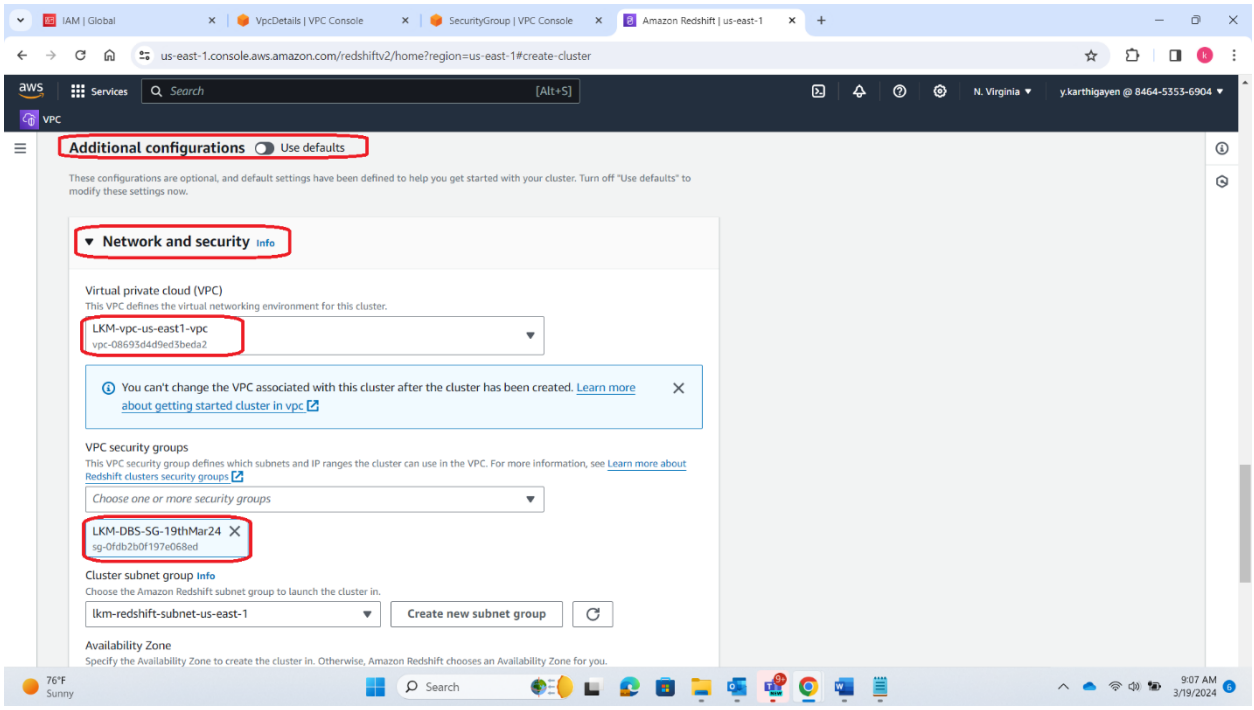


- Give some valid name to the cluster and use free trail as shown below



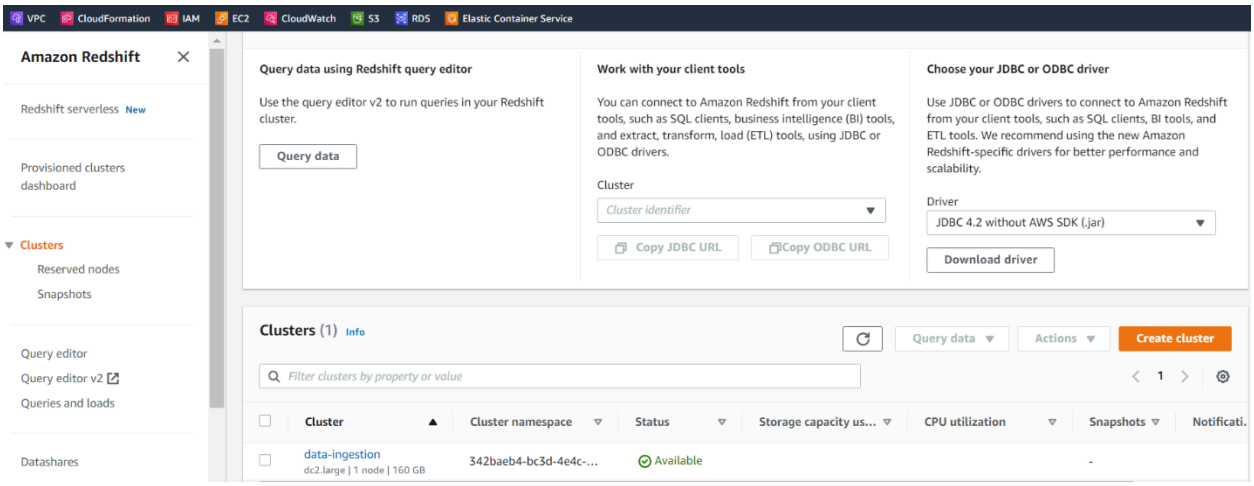


Under Additional Configuration, please change the network setting as shown in the below snapshot and choose LKM VPC and click on create cluster button



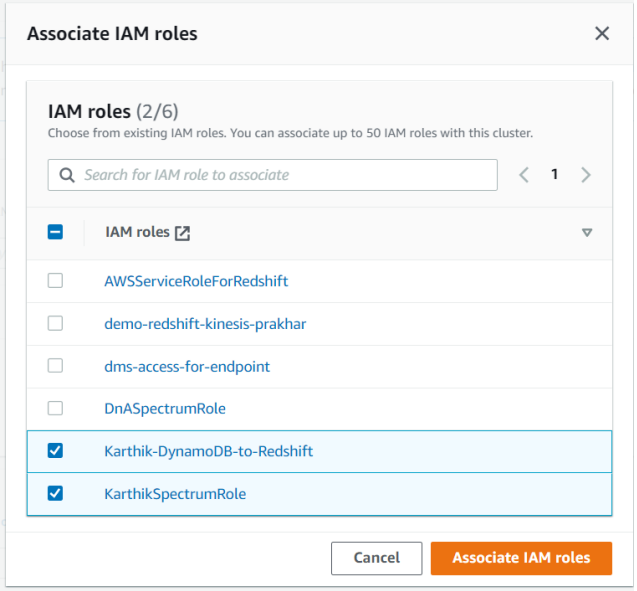
NOTE : It will take 8 to 10 minutes for cluster to comeup

Open your cluster now

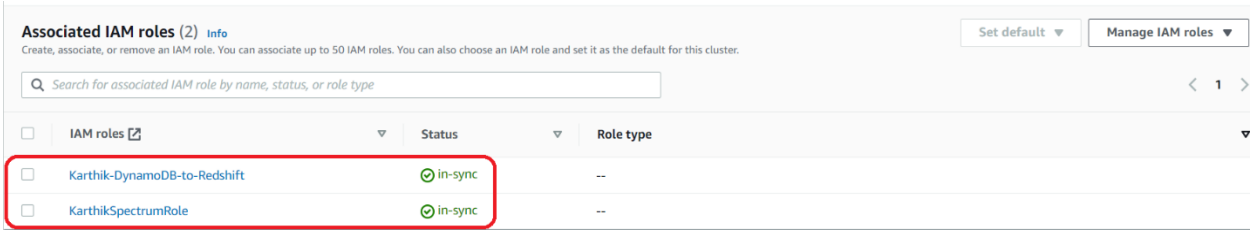


- Go to **Properties** Tab

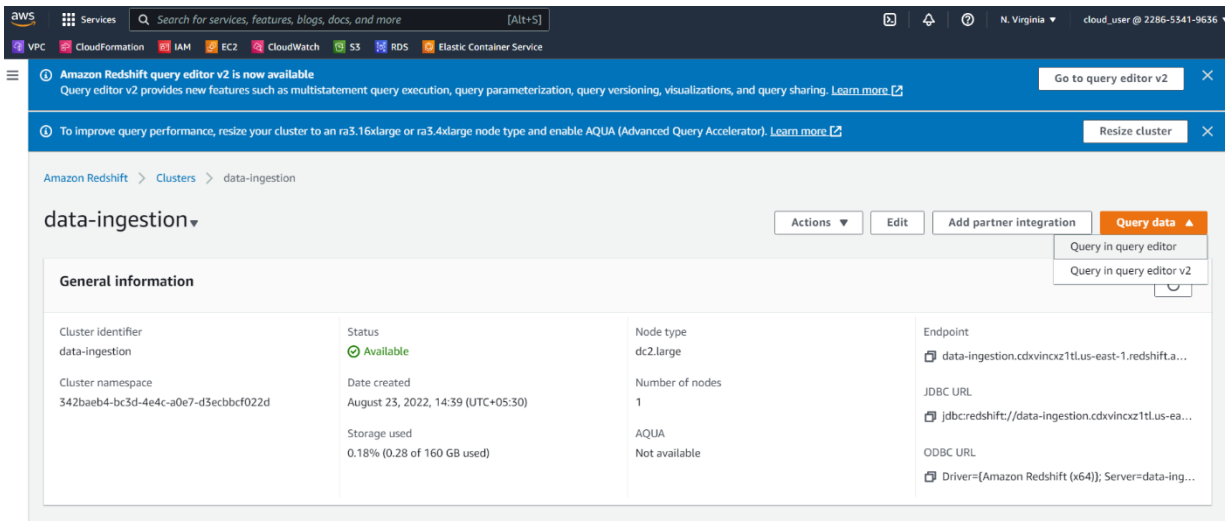
- Click on Associate IAM role
  - Add the role which we created in the beginning as shown below.



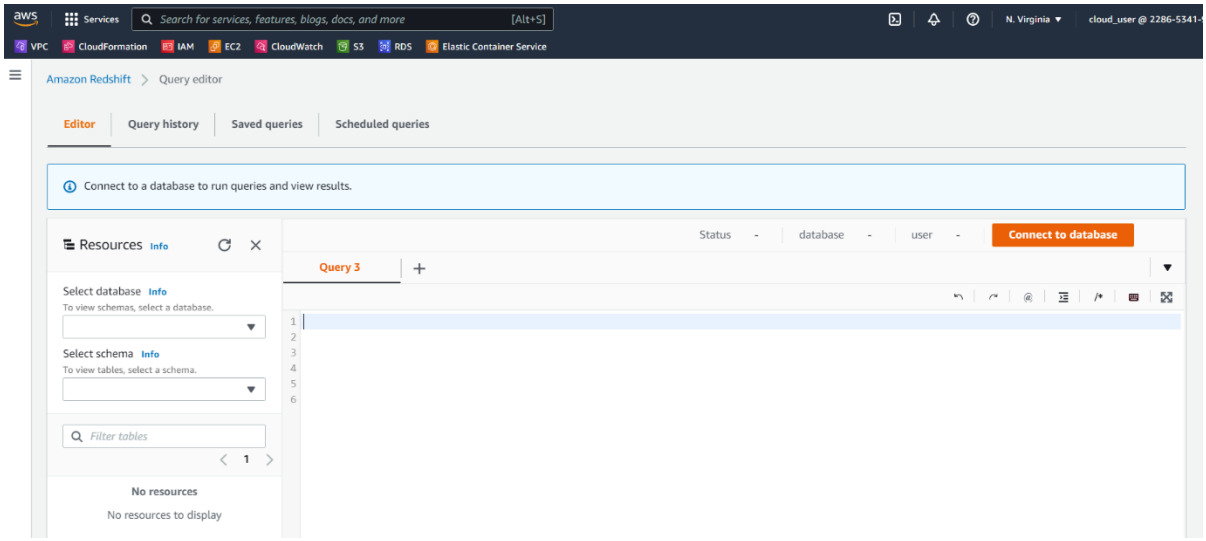
Now refresh the page and you will see as shown below IAM role **in-sync**



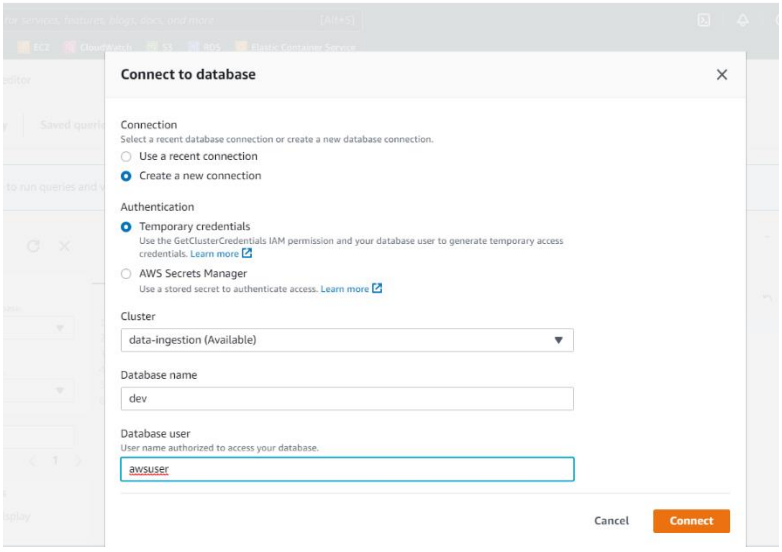
- Now open the query editor as shown below in right top corner



- Click on **connect to Database**



- Give the credentials that you used while creating the cluster as shown below and click on **connect**



- **Now lets run the SQL commands on the Redshift Query Editor**

In Redshift go to Editor->Query Editor and create a table Event as follows:

```
create table event1(
        eventid integer not null distkey,
        venueid smallint not null,
        catid smallint not null,
        dateid smallint not null sortkey,
        eventname varchar(200),
        starttime timestamp
);
```

Have a look at QueryResults tab at the bottom.

Copy the data to the above event table by using below mentioned COPY command:
<*bucketname, role and region needs to be provided* by you>

```
copy event1 from 's3://dw-rs-bucket-karthik/allevents_pipe.txt'
iam_role  'arn:aws:iam::846453536904:role/KarthikSpectrumRole'
delimiter '|' timeformat 'YYYY-MM-DD HH:MI:SS' region 'us-east-1';
```

Check how much data is in event table by running below query:

```
select count(*) from event1;
```

output will be 8798 rows

```
unload ('select * from event1 where catid = 7') to 's3://dw-rs-bucket-karthik/offloaded/' iam_role
'arn:aws:iam::846453536904:role/KarthikSpectrumRole'  parallel off ;

delete from event1 where catid = 7;

select COUNT(*) from crimedata;
```

Create external schema and tables: <data catalog is reference to Glue>

```
create external schema spectrum
from data catalog
```

```
database 'spectrumdb'
iam_role 'arn:aws:iam::846453536904:role/KarthikSpectrumRole'
create external database if not exists;
```

Create a table as shown below:

Sales is the name of table and spectrum schema is applied. Specify approximate number of rows as 172000. Examine number of rows in sales_tab.txt

Ensure that sales_tab.txt is available at s3://dw-rs-bucket-karthik/spectrum/sales/

```
create external table spectrum.sales1(
    salesid integer,
    listid integer,
    sellerid integer,
    buyerid integer,
    eventid integer,
    dateid smallint,
    qtysold smallint,
    pricepaid decimal(8,2),
    commission decimal(8,2),
    saletime timestamp)
row format delimited
fields terminated by '\t'
stored as textfile
location 's3://dw-rs-bucket-karthik/spectrum/sales/'
table properties ('numRows'='172000');
```

Go to AWS Glue -> Data Catalog-> Databases-> spectrumDB->tables->sales and examine the columns that was mentioned while creating the table.

This is how spectrum works, it keeps data catalog and metadata of the table in Glue

Go to Athena. You will now find DataSource as AWSDataCatalog, Database as spectrumdb and table as sales.
Go to Redshift and execute below queries:

```
select count(*) from spectrum.sales1;
```

Output will be 172462

```
select * from spectrum.sales1 limit 3;


select top 15 event1.eventname as event_name, sum(spectrum.sales1.pricepaid) as
gross_ticket_sales from spectrum.sales1,event1

where spectrum.sales1.eventid=event1.eventid

and spectrum.sales1.pricepaid > 30

group by event1.eventname

order by 2 desc;
```