REDSHIFT DATA PIPELINE: FROM GENERATION TO DATABASE

INTRODUCTION

The "Redshift Data Pipeline: From Generation to Database" project showcases the end-to-end process of creating and populating a database on Amazon Redshift.

This documentation provides insights into the setup of a cloud environment, including the creation of a VPC, subnets, S3 bucket, IAM roles, security groups, and Redshift cluster. It covers the generation of sample data, formatting it, and loading it into the Redshift database. Additionally, for those interested in replicating or examining the infrastructure setup, I offer an Infrastructure as Code (IaC) option click here. This project demonstrates the implementation of a robust data pipeline leveraging AWS services to efficiently manage and analyze datasets.

This demonstration walks through getting started with Amazon Redshift cluster and shows how to both load and query your data.

PREPARING THE ENVIRONMENT

A). Create a VPC.

The first step in configuring your environment is to create a virtual private cloud (VPC) to hold the resources for both your Amazon Elastic Compute Cloud (Amazon EC2) instance and Amazon Redshift database. (*The N.Virginia us-east-1 Region was used in this demonstration*).

In this project, the following settings were used:

- *Name:* Amazon Redshift Project
- *Location:* us-east-1
- *IPv4 CIDR block:* 10.60.0.0/16
- *Internet gateway:* Redshift IGW
- *Public route table:* Redshift Project Public
- *Private route table:* Redshift Project Private

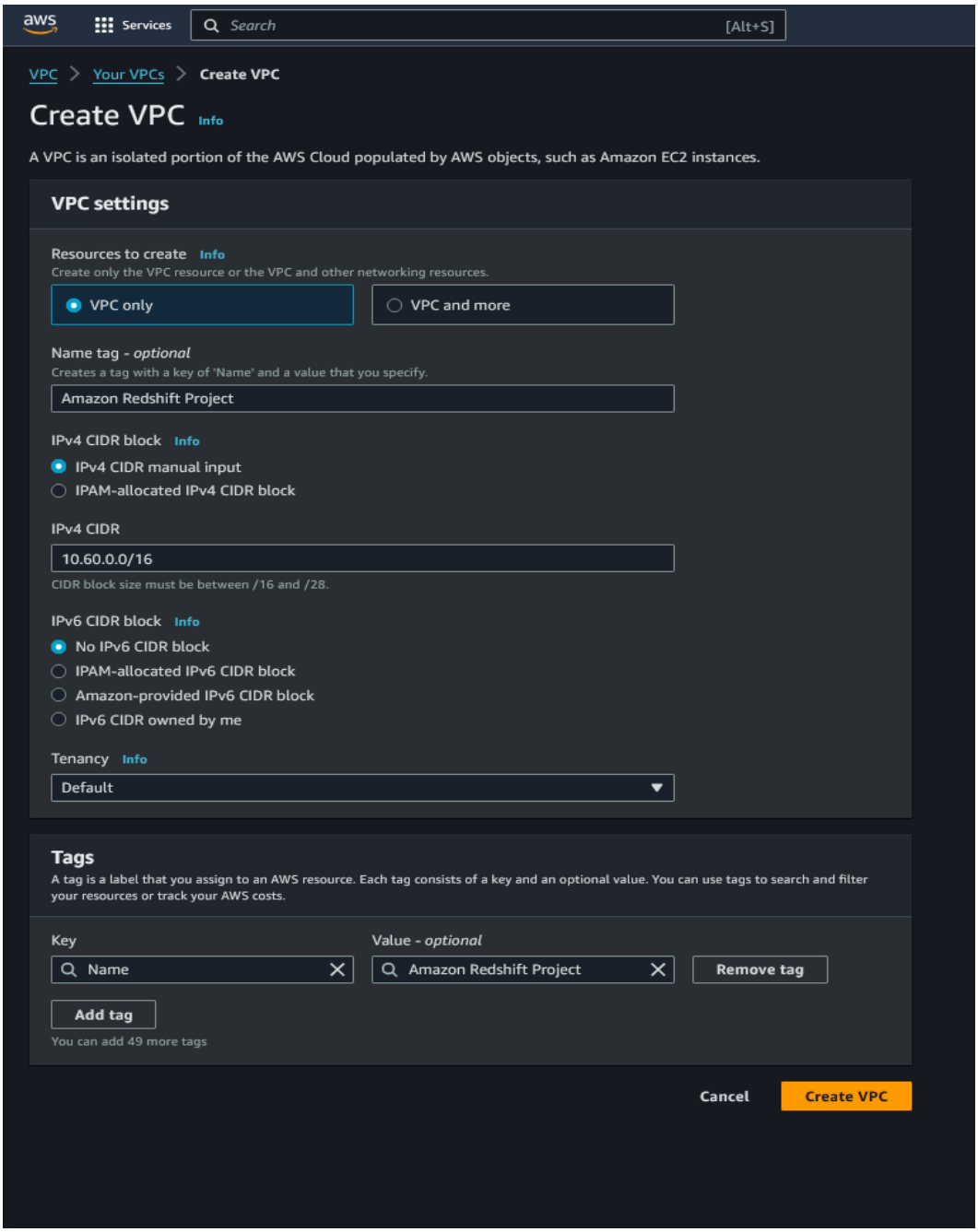

Figure 1. VPC Creation - After review click the 'create VPC' button.

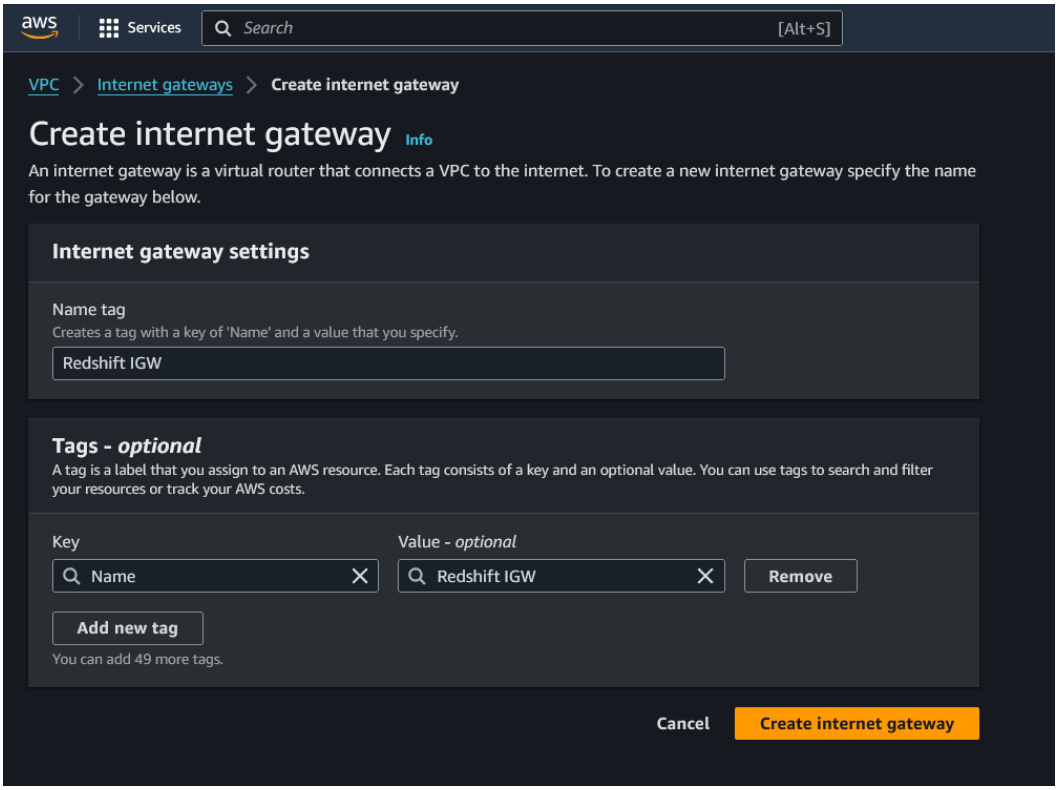

Figure 2. Internet Gateway creation.

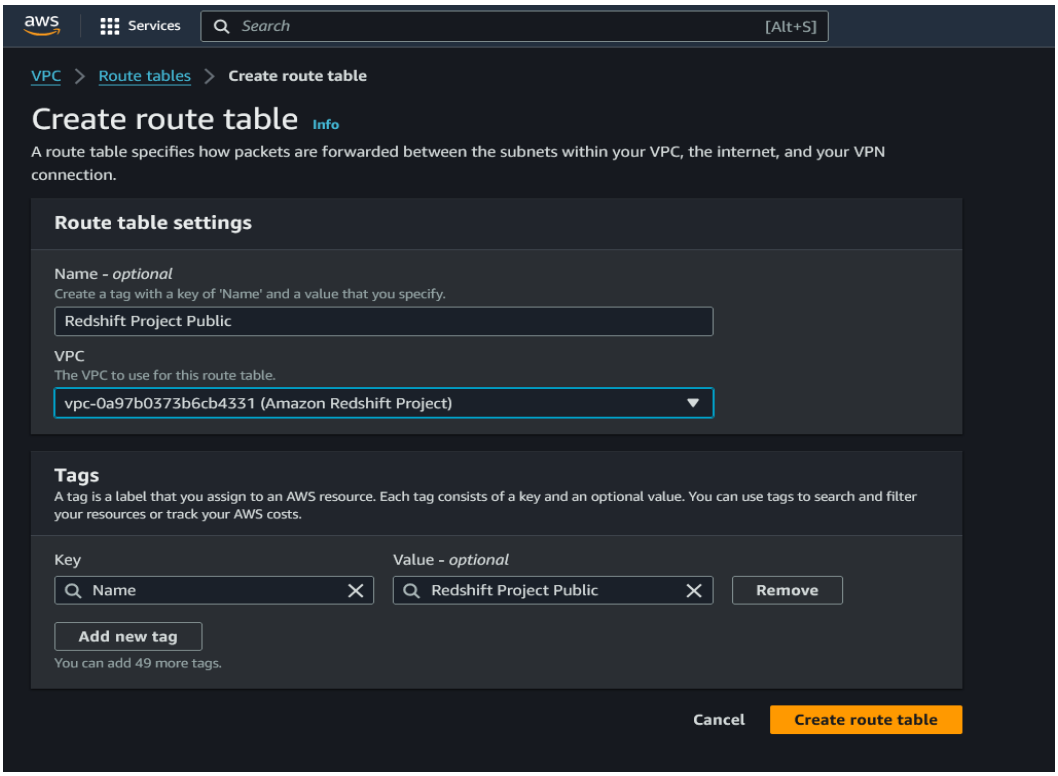

Figure 3. Public Route Table, click on the 'create route' table button.

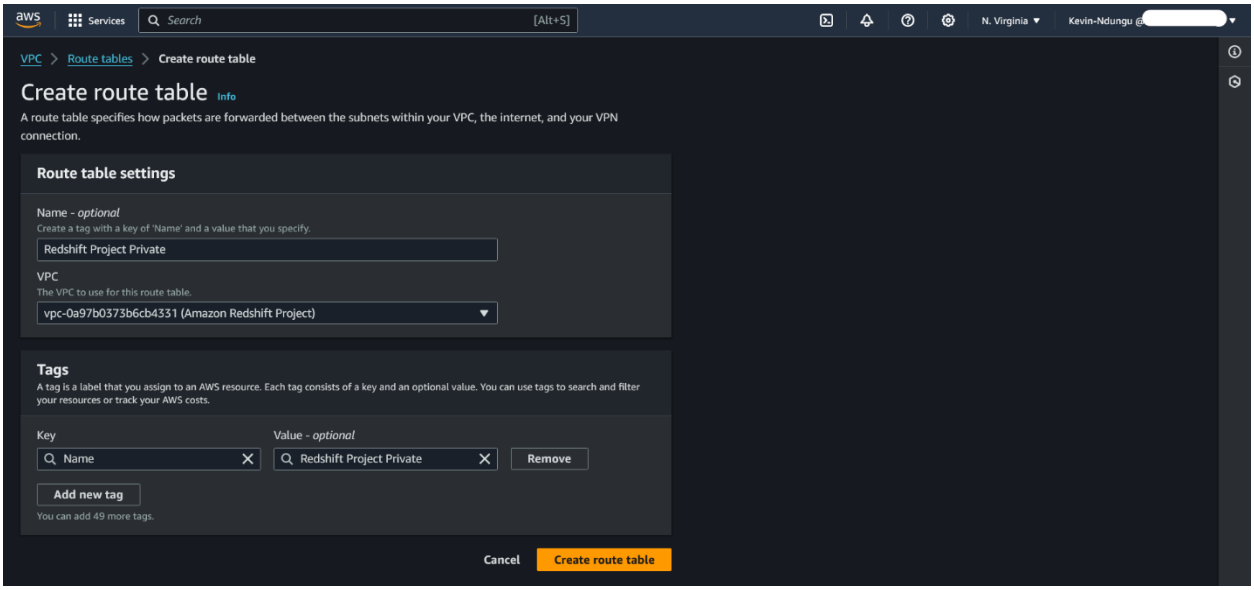

Figure 4. Private Route Table, click on the 'create route table' button.

B). Create the subnets.

Amazon Redshift is a managed service that connects to subnets in your VPC.

In this project, the following settings were used:

- **Private subnet 1**
    - *Name:* Redshift Private 01
    - *CIDR:* 10.60.101.0/24

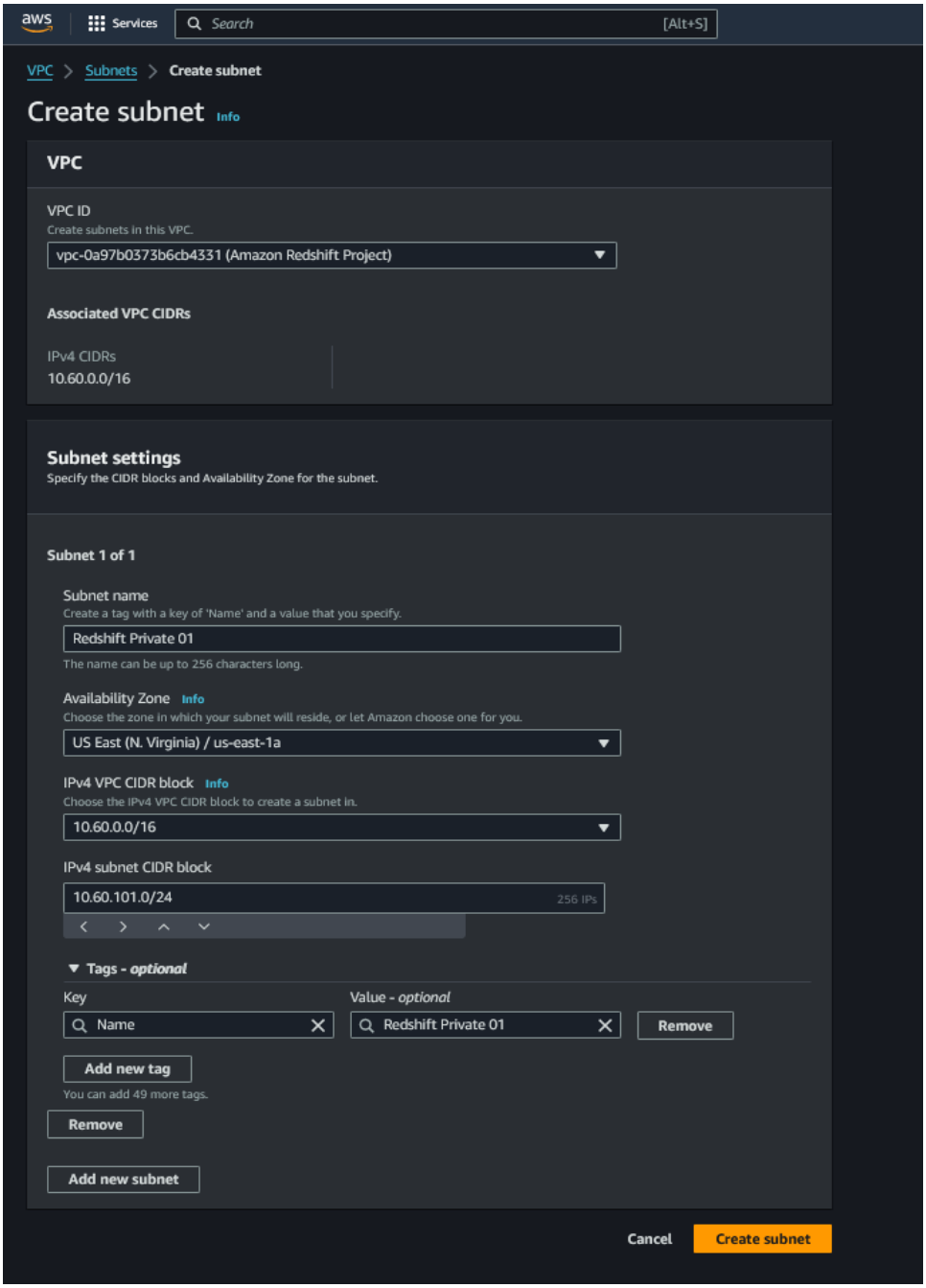

Figure 5. Click on the 'create subnet' button.

- **Private subnet 2**
    - *Name:* Redshift Private 02
    - *CIDR:* 10.60.102.0/24

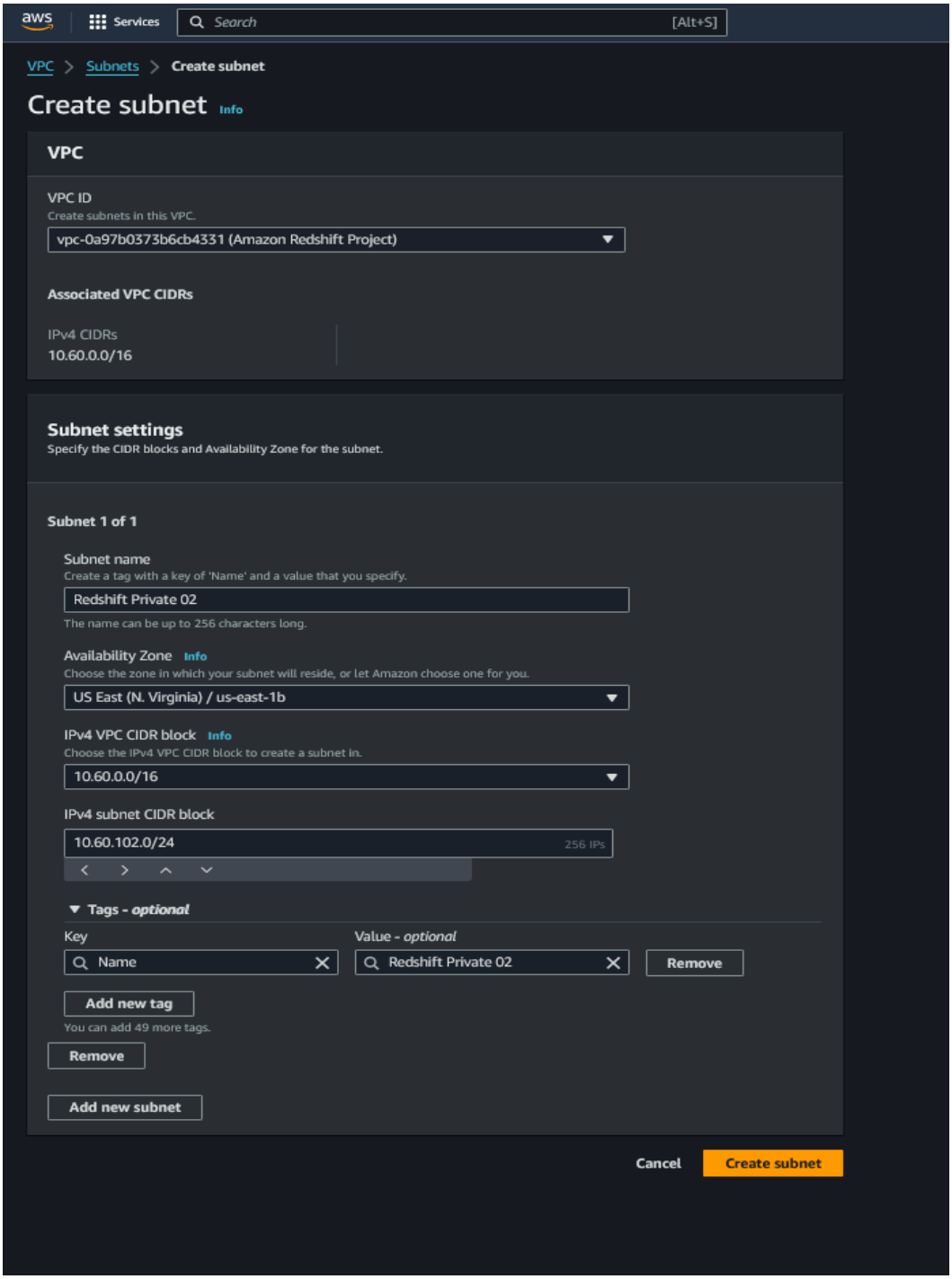

Figure 6. Click on the 'create subnet' button.

C). Create an Amazon S3 bucket.

The data used for this demo is randomly generated and uploaded to an Amazon Simple Storage Service (Amazon S3) bucket.

In this project, the following settings were used:

- *Region:* us-east-1
- *Bucket:* redshift-database-demo *(Make sure to use a unique bucket name)*

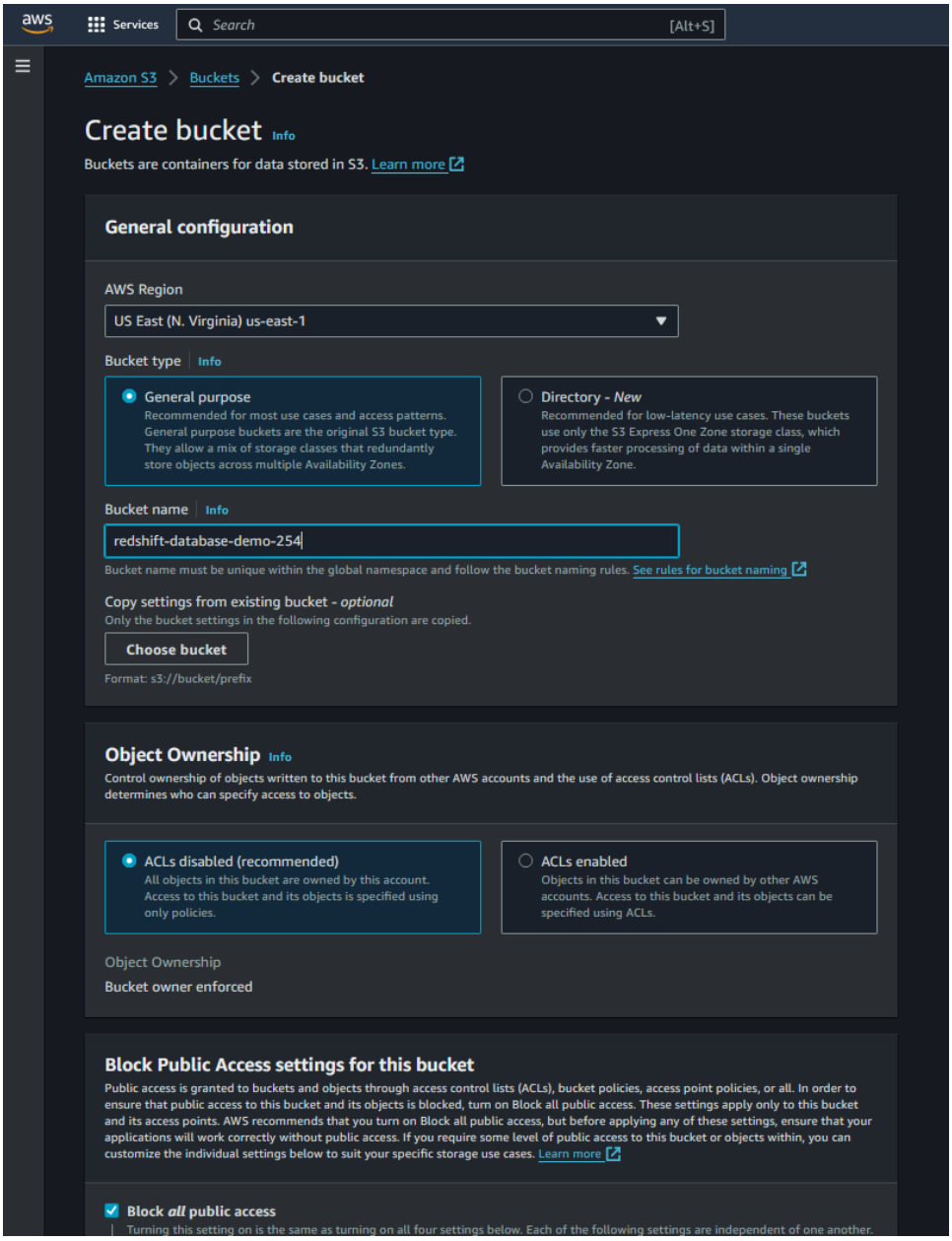

Figure 7.

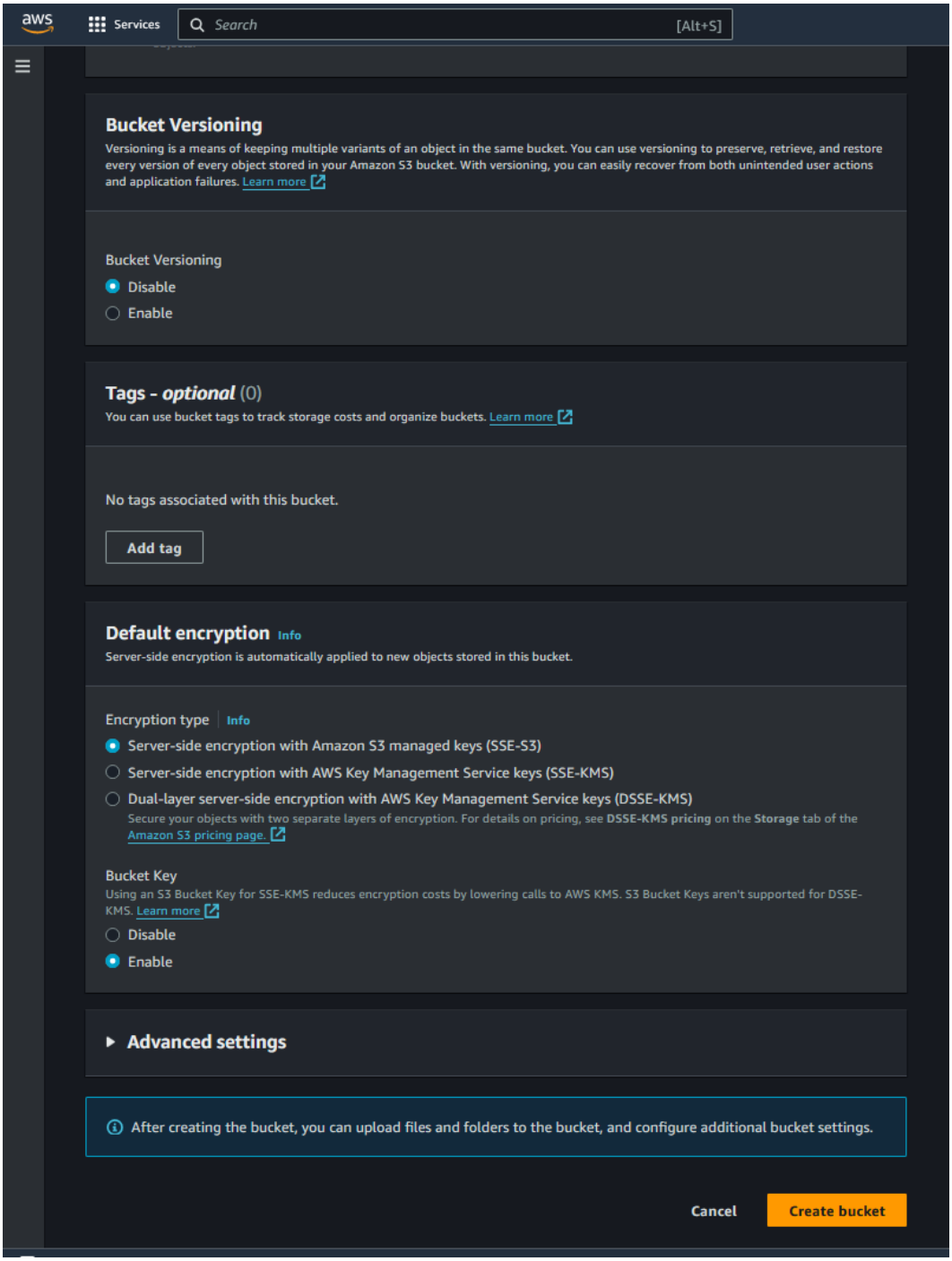

Figure 8.

D). Create an Amazon S3 endpoint.

By default, traffic between an Amazon Redshift cluster and Amazon S3 traverses the public AWS network. A VPC endpoint forces all COPY and UNLOAD traffic between your cluster and your data on Amazon S3 to stay in your VPC.

In this project, the following settings were used:

- *Service name:* com.amazonaws.us-east-1.s3
- *Endpoint type:* Gateway
- *Route table:* Redshift Project Private

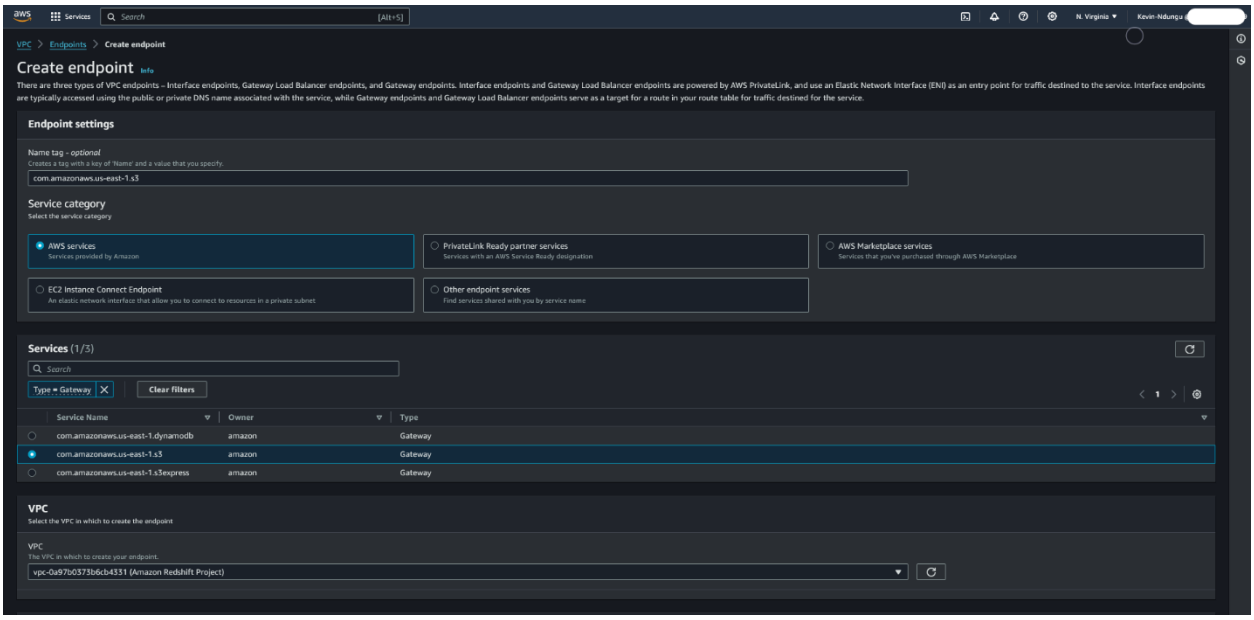

Figure 9. Part A.

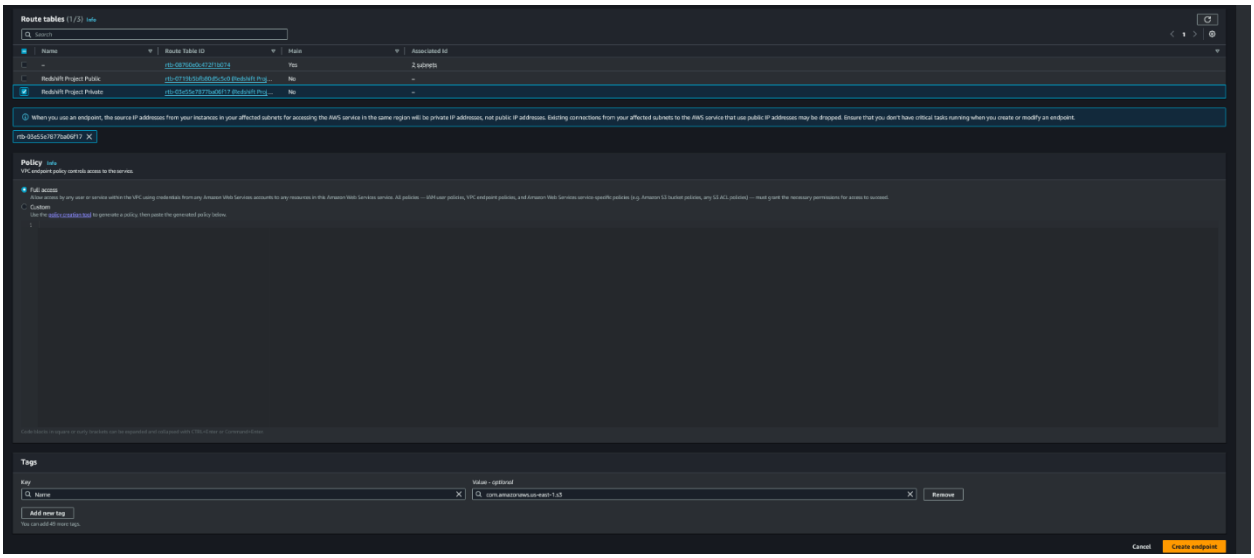

Figure 10. Click the Create endpoint button.

NB: The Amazon Redshift cluster must be in the same Region as the S3 bucket.

E). Create an IAM role.

Amazon Redshift needs permission to copy data to and from S3 buckets. This access is granted using AWS Identity and Access Management (IAM) roles. A role defines a set of permissions for making AWS service requests.

In this project, the following settings were used:

- *Role name:* RedshiftDemoProject
- *Policy name:* RedshiftDemoProject

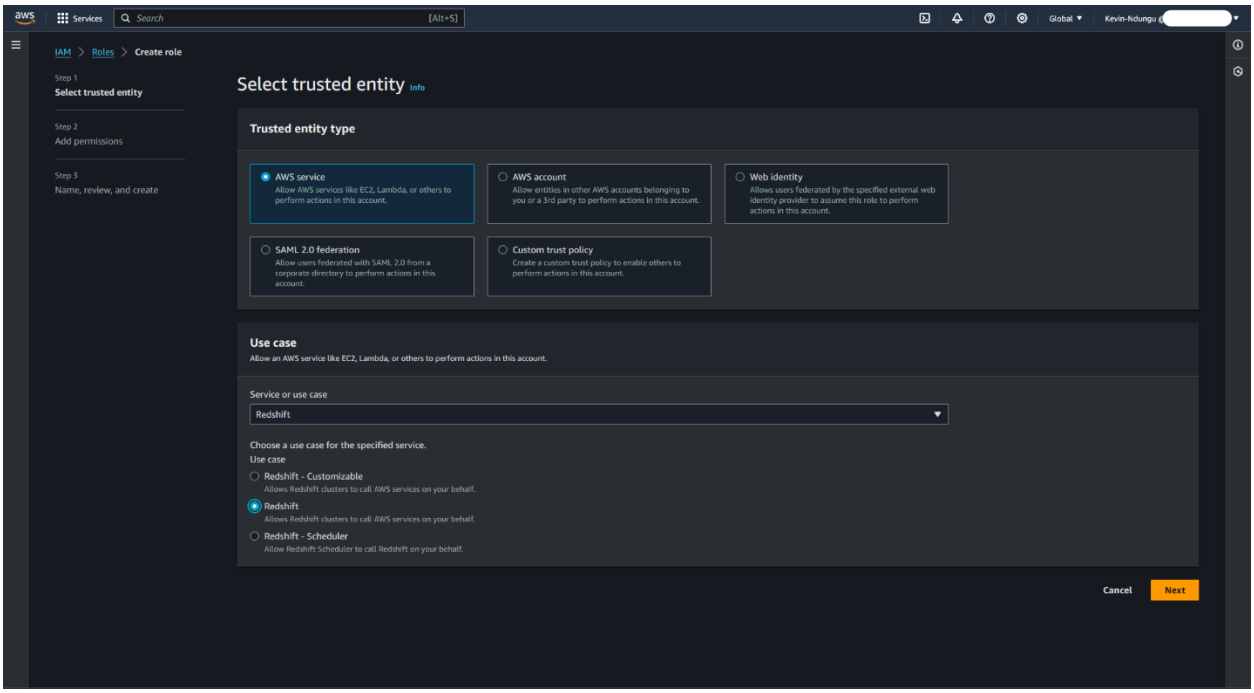

Figure 11.

In this example, the role and policy names are the same. This was for ease of use. They can be different.

The policy gives Amazon Redshift full access to a specific S3 bucket, redshift-database-demo, and the objects inside it.

It also has a trust relationship with redshift.amazonaws.com. The trust relationship defines who/what can assume a role. Because of this trust relationship, only Amazon Redshift can assume this role, and it has to be explicitly assigned to the cluster.

F). Create the security groups.

Security groups control access to EC2 instances and Amazon Redshift clusters. In this project, two security groups were created: one is for an EC2 instance and the other for the Amazon Redshift cluster.

In this project, the following settings were used:

- **Group 1**
    - *Name:* Redshift EC2
    - *Description:* EC2 access for Redshift Project
    - *Rule:* Allow SSH (port 22) from 0.0.0.0/0 (the internet)

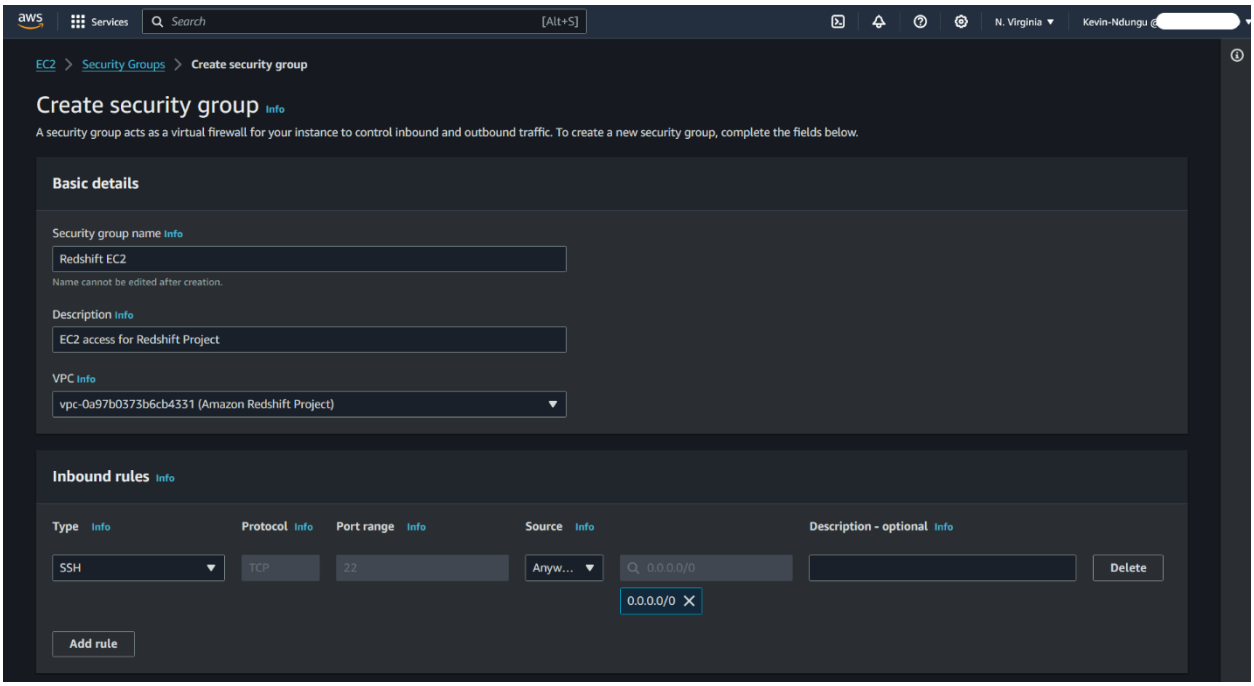

Figure 12.

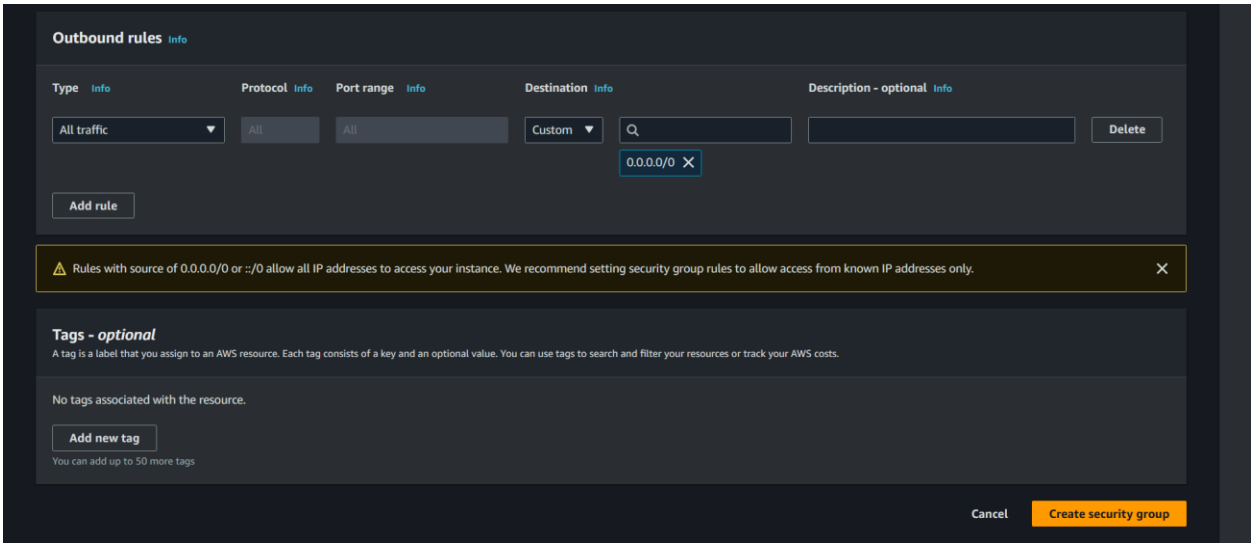

Figure 13. Click the Create security group button.

- **Group 2**
    - *Name:* Redshift Access
    - *Description:* Access to Redshift
    - *Rule:* Allow TCP port 5439 from the group "Redshift EC2"

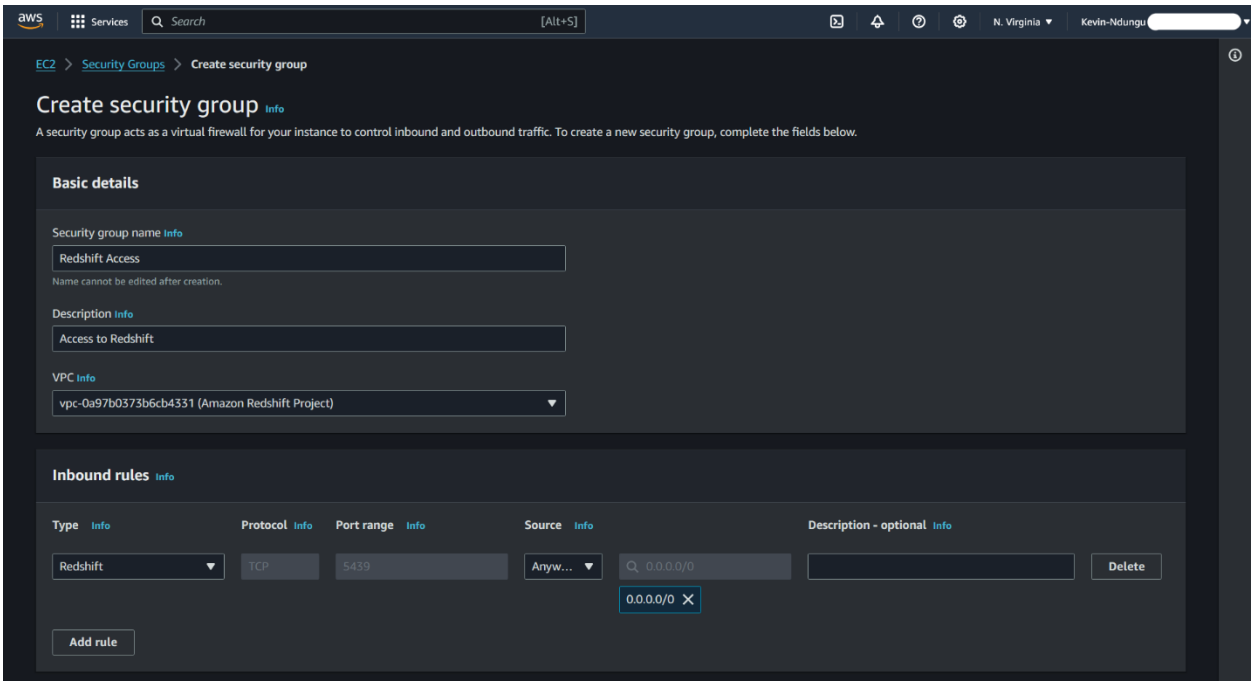

Figure 14.

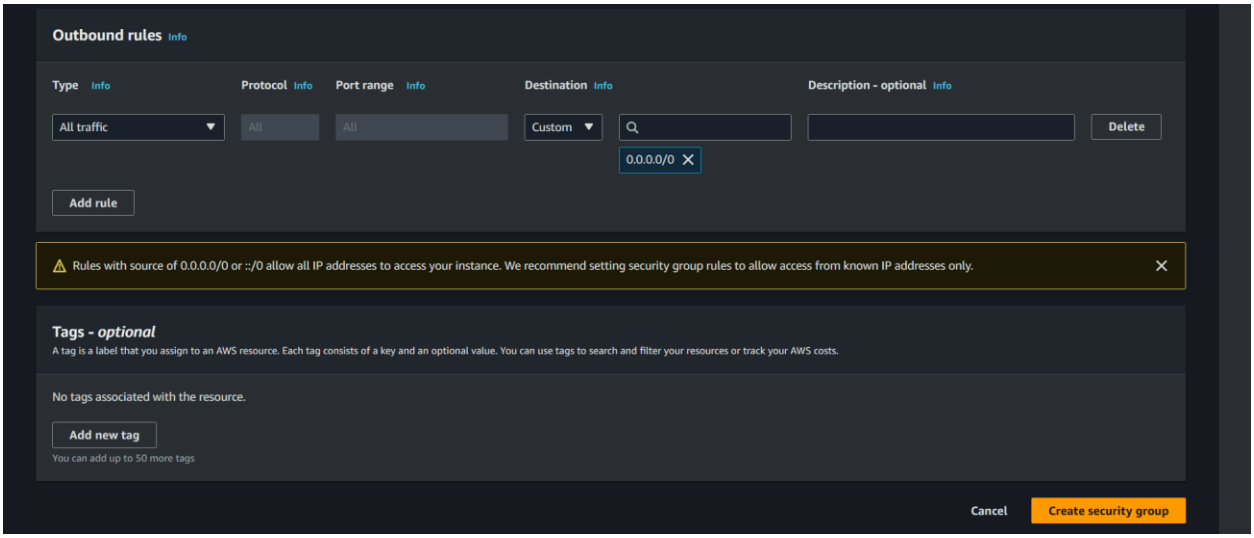

Figure 15. Click the Create security group button.

You could use a single group for both EC2 and Amazon Redshift access.

G). Create the subnet group.

It is recommended that you place your Amazon Redshift cluster in a private subnet.

In this project, the following settings were used:

- *Name:* Redshift-subnet-group
- *Description:* Private subnet access for Redshift
- *VPC:* Amazon Redshift Project
- Add all private subnets.

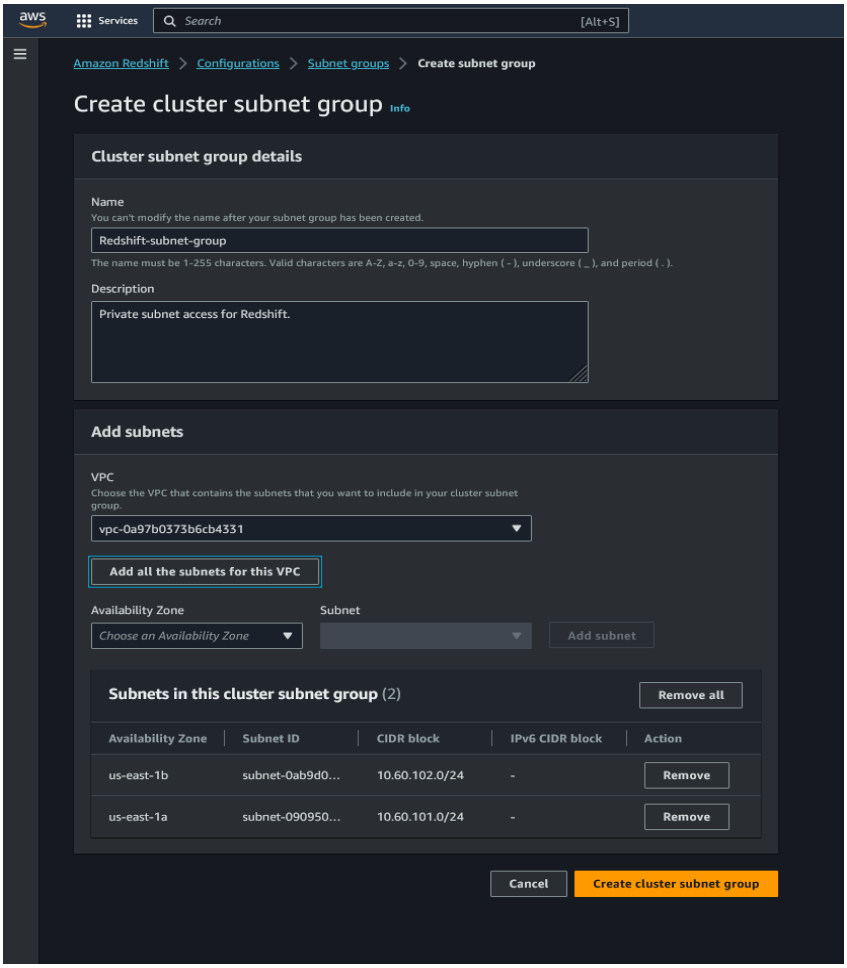

Figure 16. Click the Create cluster subnet group button.

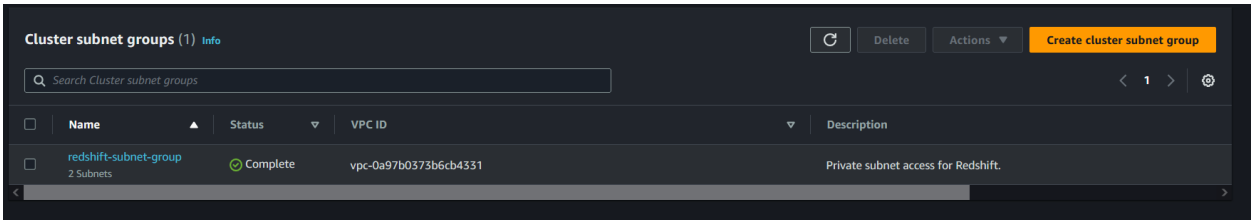

Figure 17. Successful creation of the cluster subnet group.

Now at this stage we begin creation of the Amazon Redshift cluster.

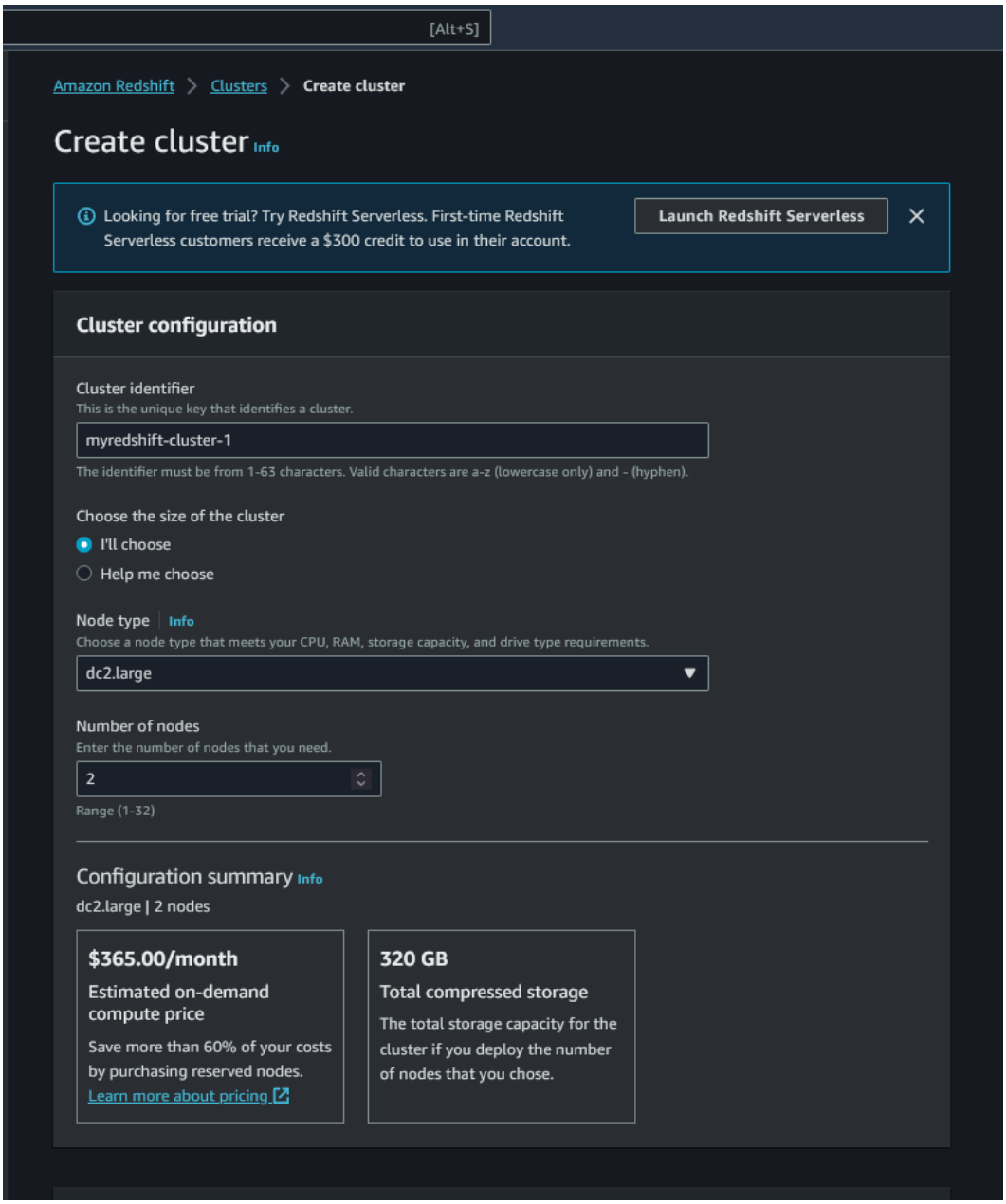

Figure 18.

Choose the dc2.large node type and the number of nodes two.

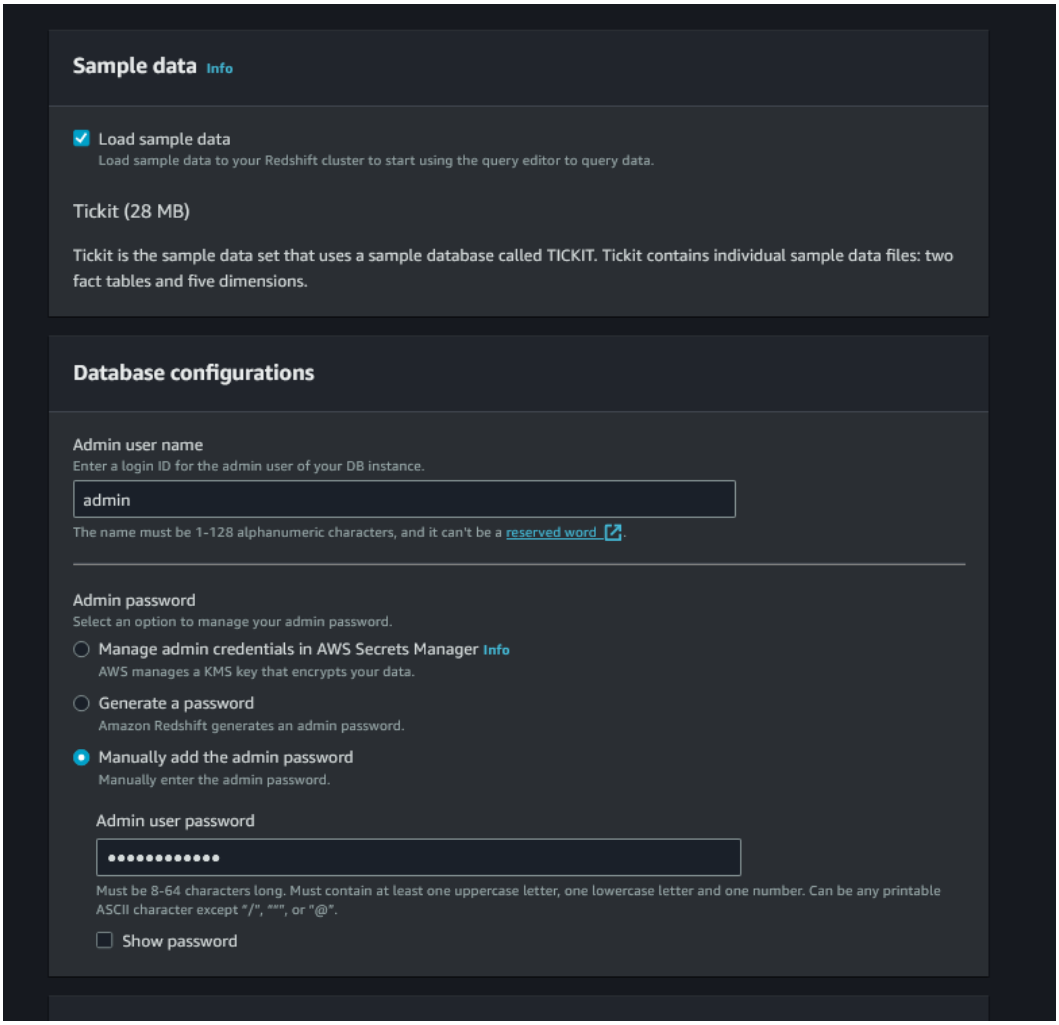

Figure 19.

In the above diagram (Figure 19), you could opt to enable sample data generation. Configure the master's name and master password.

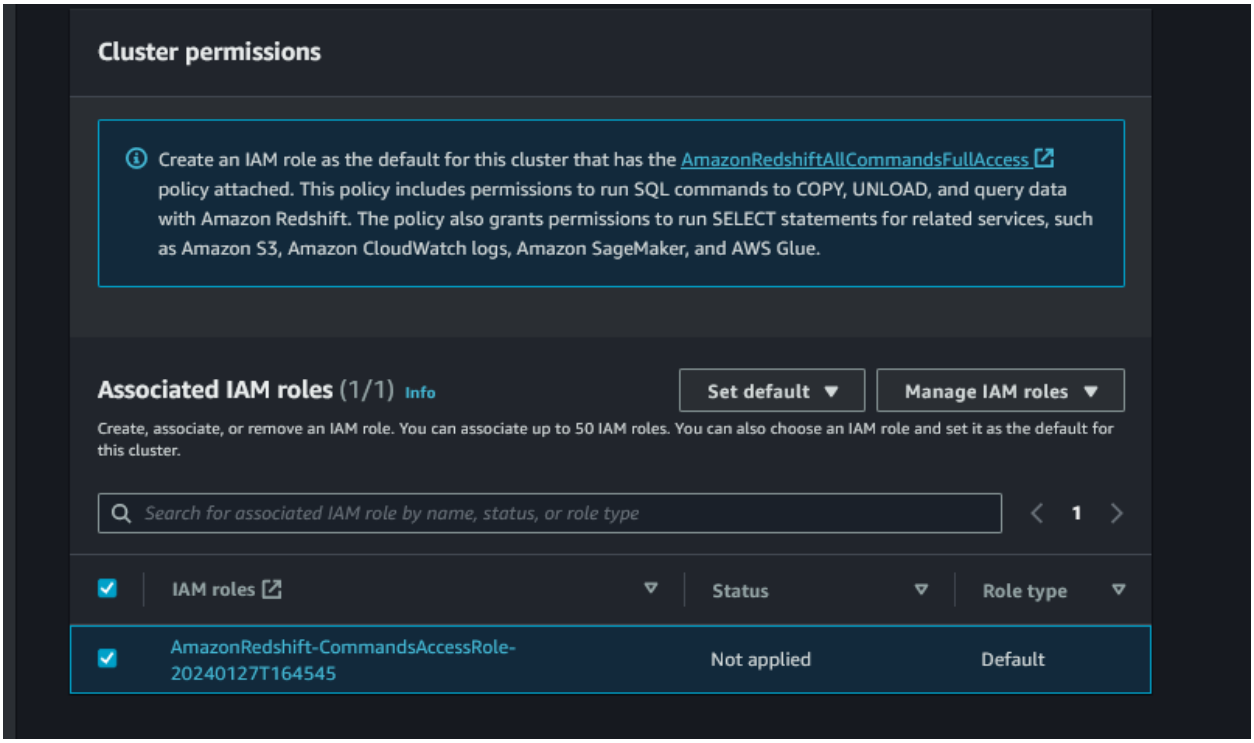

Figure 20.

Allocate the appropriate IAM role that was created earlier for Redshift's access to the S3 bucket.

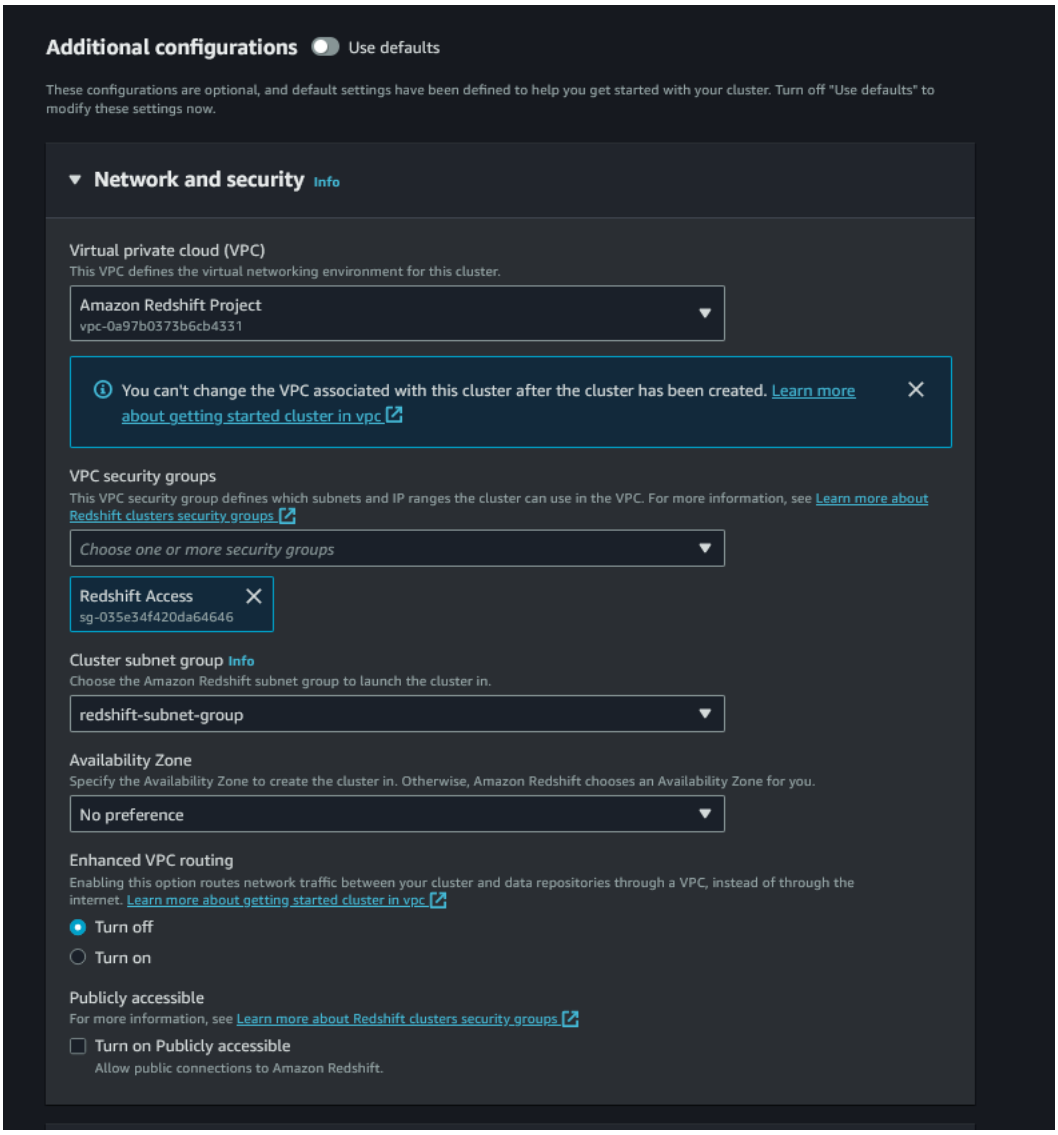

Figure 21.

Choose the VPC we created earlier (Amazon Redshift Project) and select Redshift Access as the VPC security group, deselect the default security group. Choose the redshift subnet group for the cluster and leave the rest of the option as is.

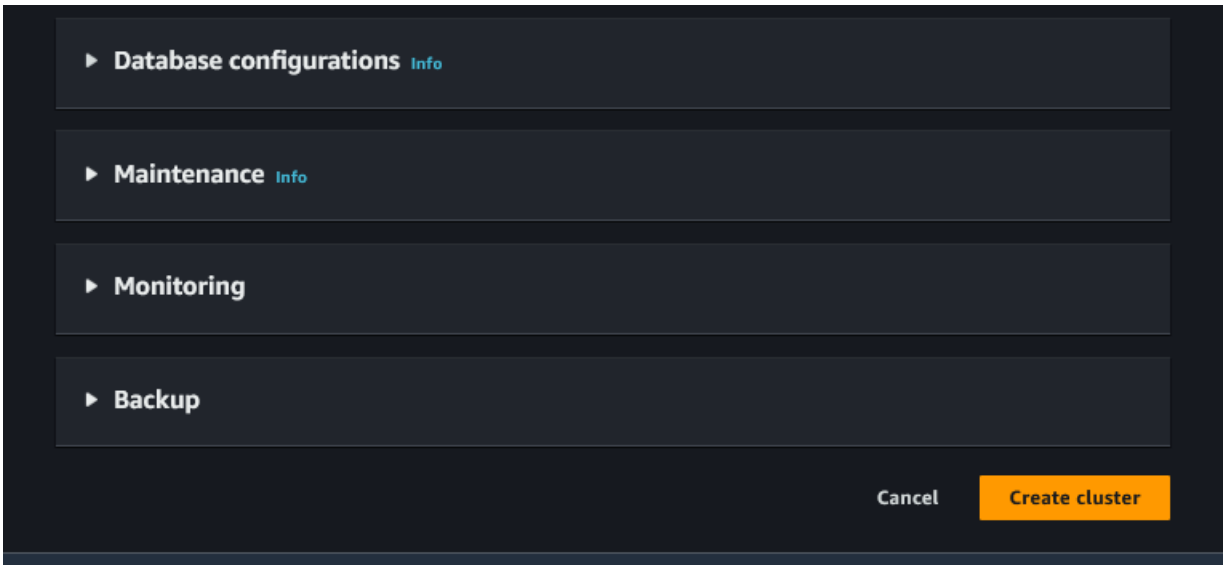

Figure 22. Leave this option in their default settings and click on the create cluster button.

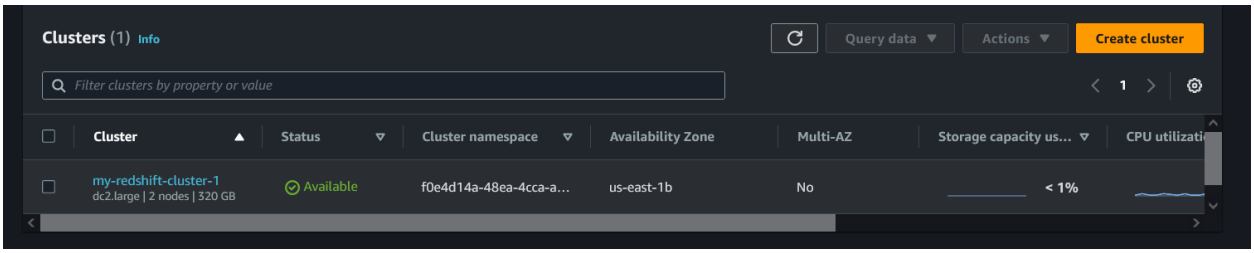

Figure 23. Wait until the cluster creation is complete and available.

INTERACTING WITH OUR NEWLY CREATED AMAZON REDSHIFT CLUSTER

On the left side of the console on the navigation panel, get to the query console by clicking on the query editor. (You can choose between the Query editor or Query editor v2).

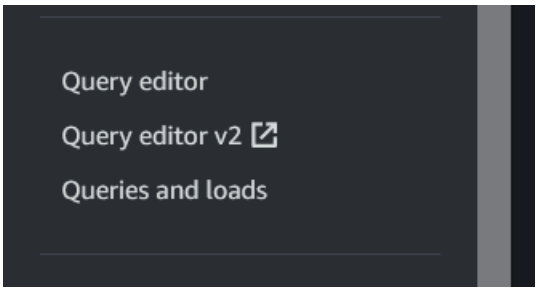

Figure 24.

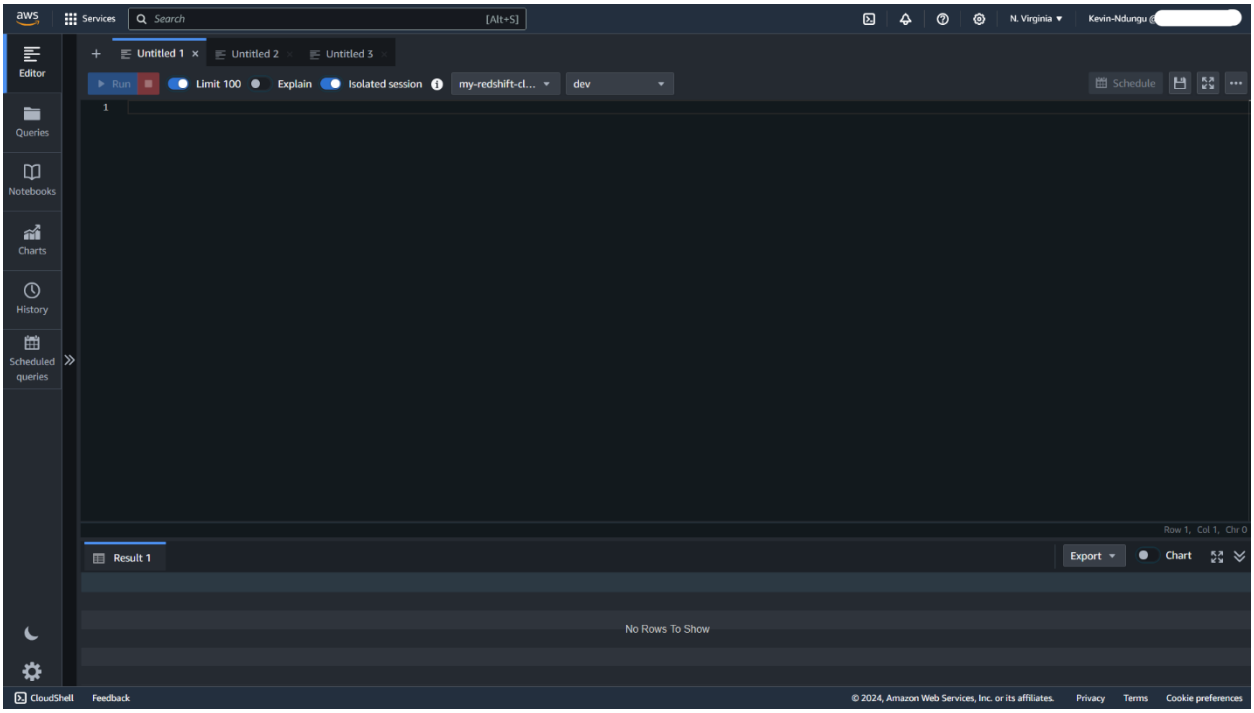

Figure 25.

Once the editor opens, the next step is to create a table (*named users*), in this case the below create table query was used:

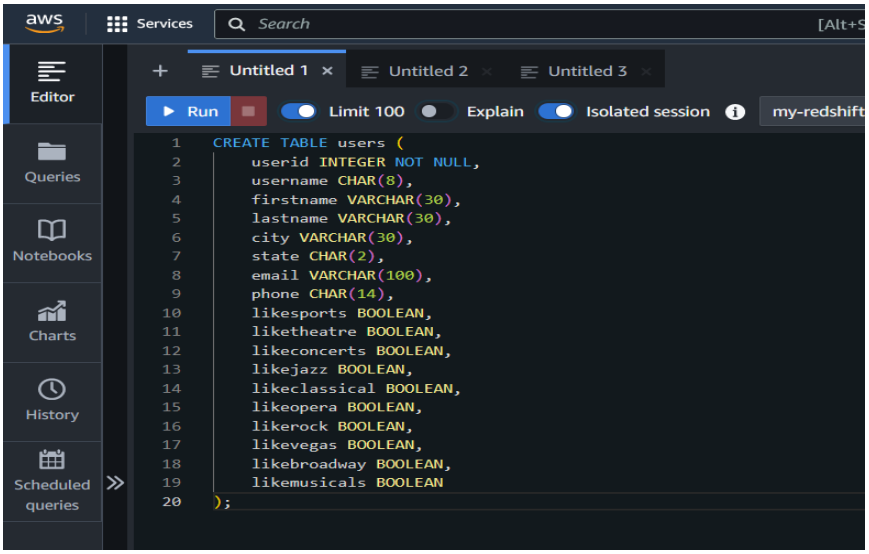

Figure 26. Click on the Run button at the top left to create table.

Once the table (users) is created it will be listed among other tables in the query's navigation panel as shown below:

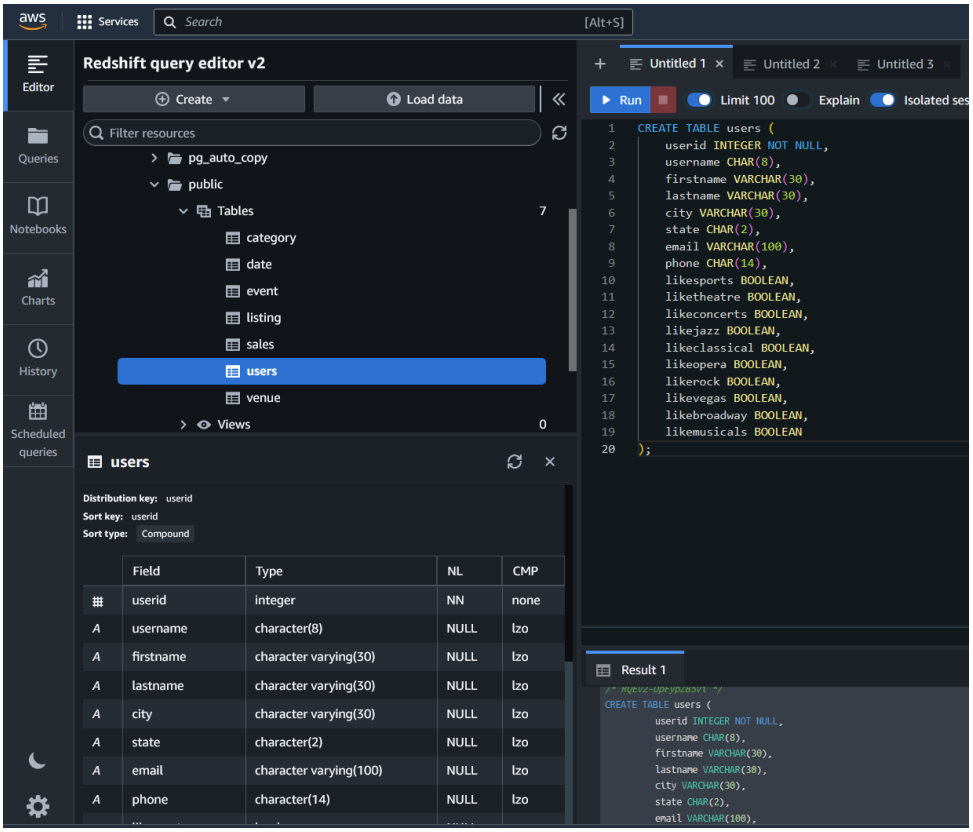

Figure 27.

Since the table is up, run the query in figure 28 was used to copy the data generated and stored in an S3 bucket to our newly created ‘users’ table.

So, I'm going to copy users’ data from my S3 bucket. The credential that I passed is a role and it gives the ARN the Amazon resource name of the role. And in the file, itself is delimited by a comma and I ran the query.

(Note*: Generate the data separately based on the information in the create table query and save it in a pipe delimited text or comma delimited format, store it in the S3 bucket created earlier*).

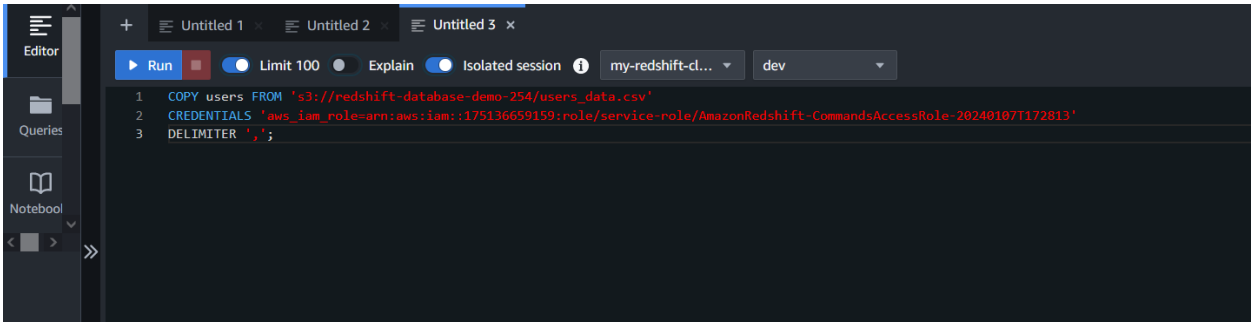

Figure 28.

Once the users table is populated, we can begin running some queries and get to view the results in the bottom display of the query editor.

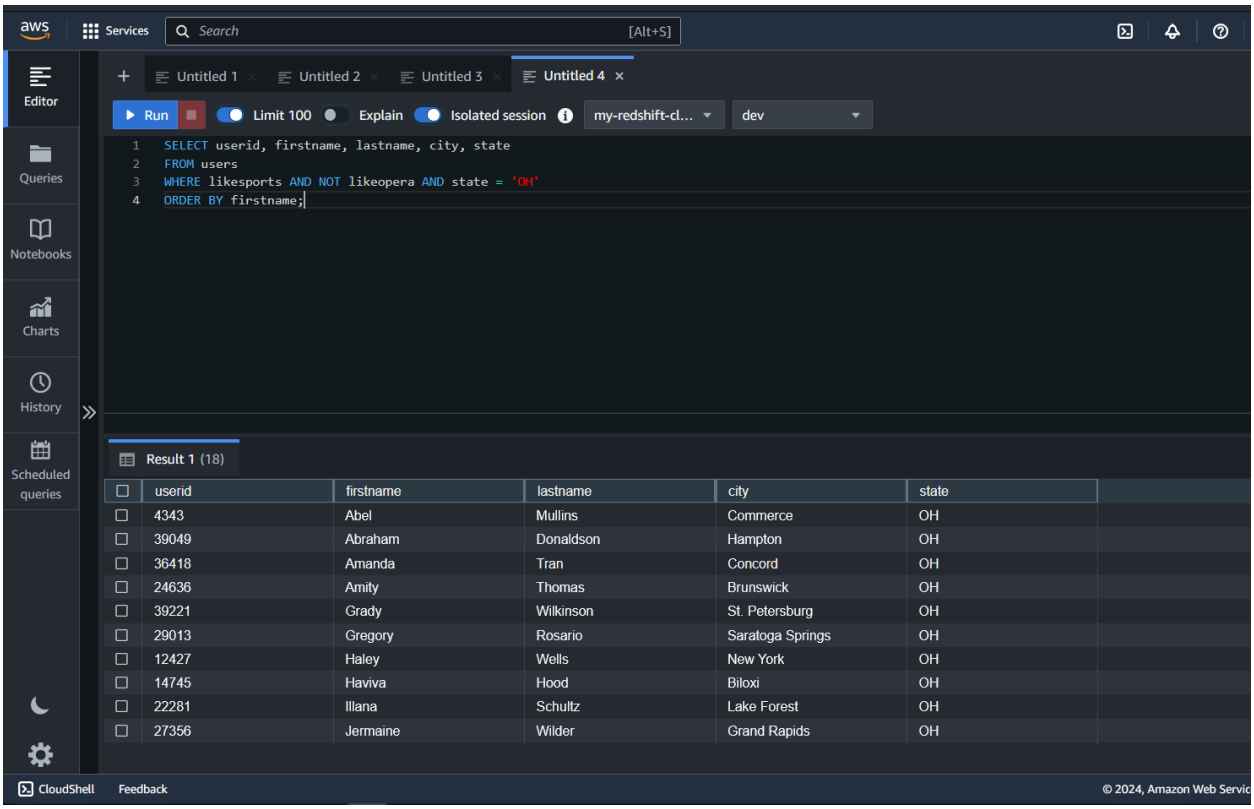

Figure 29.

In running the query code in the above diagram, we want to get information user ID, first name, last name, city state from users where people like sports and do not like opera and the state is Ohio and then giving an order by first name. The output is displayed in the Result tab below the editor area.

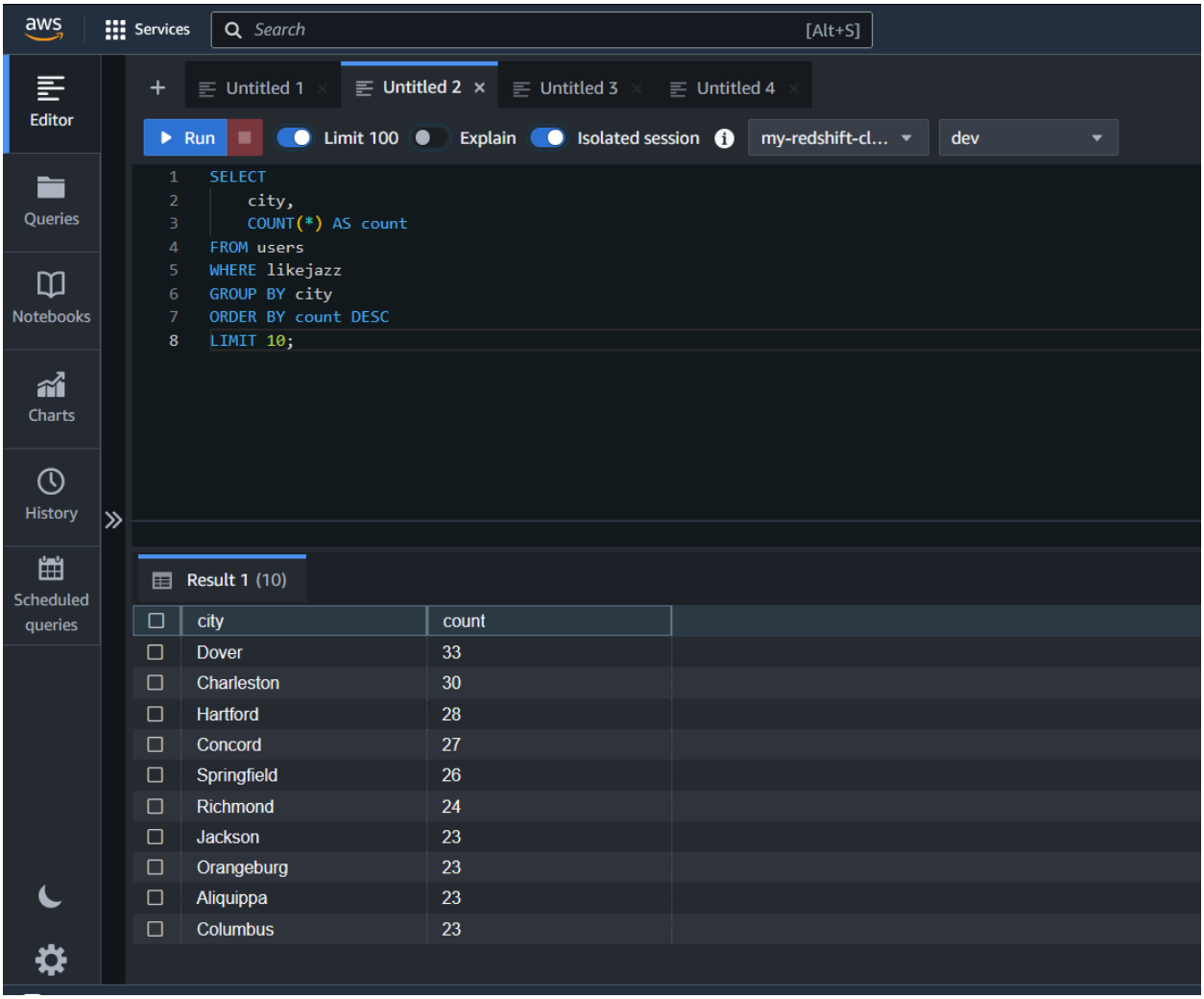

Figure 30.

In the above diagram I wanted to see the top 10, I ran the query and I as shown the top 10 are 33 in Dover down to 23 in Columbus.

CONCLUSION

In conclusion, the Redshift Data Pipeline project has reached a significant milestone with the successful configuration and creation of the Amazon Redshift cluster. Throughout this documentation, we've meticulously detailed the setup and configuration of our cloud environment, laying the foundation for a robust data infrastructure. From the inception of our Virtual Private Cloud (VPC) to the establishment of connectivity between our Redshift cluster and Amazon S3, each step has been carefully orchestrated to ensure seamless data management and analysis.

We're now poised to leverage the power of Amazon Redshift for advanced analytics and insights generation. As we continue our journey, this project serves as a testament to the efficacy of cloud-based data pipelines in driving business intelligence and decision-making processes.