ETL From AWS S3 to Amazon Redshift with AWS Lambda dynamically.

BY

Manu Mathew

Amal P

Gowtham Jayan

CHAPTER 1

INTRODUCTION OF AWS SHIFT

AWS Redshift is a data warehousing solution from Amazon Web Services. Redshift shines in its ability to handle huge volumes of data capable of processing structured and unstructured data in the range of exabytes (10^{18} bytes). However, the service can also be used for large-scale data migrations.

Similar to many other AWS services, it can be deployed with just a few clicks and provides a plethora of options to import data. Additionally, the data in Redshift is always encrypted for added security.

Redshift helps to gather valuable insights from a large amount of data. With the easy-to-use interface of AWS, you can start a new cluster in a couple of minutes, and you don't have to worry about managing infrastructure.

1.1 UNIQUE ABOUT REDSHIFT

Redshift is an OLAP-style (Online Analytical Processing) column-oriented database. It is based on PostgreSQL version 8.0.2. This means regular SQL queries can be used with Redshift. But this is not what separates it from other services. The fast delivery to queries made on a large database with exabytes of data is what helps Redshift stand out.

Fast querying is made possible by Massively Parallel Processing design or MPP. The technology was developed by ParAccel. With MPP, a large number of computer processors work in parallel to deliver the required computations. Sometimes processors situated across multiple servers can be used to deliver a process.

Unlike most MPP vendors, ParAccel does not sell MPP devices. Their software can be used on any hardware to harness the power of multiple processors. AWS Redshift uses the MPP technology of ParAccel. In fact, Redshift was started following capital investment by AWS in ParAccel and using MPP technology from ParAccel. Now the company is part of Actian.

1.2 WHEN WOULD YOU WANT TO USE AMAZON REDSHIFT

Amazon Redshift is used when the data to be analyzed is humongous. The data has to be at least of a petabyte-scale (10^{15} bytes) for Redshift to be a viable solution. The MPP technology used by Redshift can be leveraged only at that scale. Beyond the size of data, there are some specific use cases that warrant its use.

Real-time analytics

Many companies need to make decisions based on real-time data and often need to implement solutions quickly too. Take Uber for example.

Based on historical and current data, Uber has to make decisions quickly. It has to decide surge pricing, where to send drivers, what route to take, expected traffic, and a whole host of data.

Thousands of such decisions have to be made every minute for a company like Uber with operations across the globe. The current stream of data and historical data has to be processed in order to make those decisions and ensure smooth operations. Such instances can use Redshift as the MPP technology to make accessing and processing data faster.

Combining multiple data sources

There are occasions where structured data, semi-structured data, and/or unstructured data have to be processed to gain insights. Traditional business intelligence tools lack the capability to handle the varied structures of data from different sources. Amazon Redshift is a potent tool in such use cases.

Business intelligence

The data of an organization needs to be handled by a lot of different people. All of them are not necessarily data scientists and will not be familiar with the programming tools used by engineers.

They can rely on detailed reports and information dashboards that have an easy-to-use interface. Highly functional dashboards and automatic report creation can be built

using Redshift. It can be used with tools like Amazon Quick sight and also third-party tools created by AWS partners.

Log analysis

Behavior analytics is a powerful source for useful insights. Behavior analytics provide information on how a user uses an application, how they interact with it, the duration of use, their clicks, sensor data, and a plethora of other data.

The data can be collected from multiple sources — including a web application used on a desktop, mobile phone, or tablet — and can be aggregated and analyzed to gain insight into user behavior. This coalescing of complex datasets and computing data can be done using Redshift.

Redshift can also be used for traditional data warehousing. But solutions like the S3 data lake would likely be better suited for that. Redshift can be used to perform operations on data in S3, and save the output in S3 or Redshift.

1.3 THE BENEFITS OF USING AWS REDSHIFT

The very distinctive advantage of using AWS Redshift is the cost-benefit to your organization. It costs only a fraction (roughly one-twentieth) of the cost of competitors like Teradata and Oracle.

In addition to cost, there are a number of benefits to using Redshift.

- **Speed**. With the use of MPP technology, the speed of delivering output on large data sets is unparalleled. No other cloud service providers can match the speed at the cost AWS provides the service.
- **Data Encryption**. Amazon provides the facility for data encryption for any part of Redshift operation. You as the user can decide which operations need encryption and those that do not need encryption. Data encryption provides an added layer of security.
- **Use familiar tools**. Redshift is based on PostgreSQL. All the SQL queries work with it. Additionally, you are free to choose any SQL, ETL (Extract, Transform, Load),

and Business Intelligence (BI) tools you are familiar with. There is no requirement to use the tools provided by Amazon.

- **Intelligent Optimization**. For a large data set, there would be a number of ways to query data with the same parameters. The different commands will have different levels of data utilization. AWS Redshift provides tools and information to improve queries. It will also provide tips to improve the database automatically. These can be utilized for an even faster operation that is less intensive on resources.
- **Automate repetitive tasks**. Redshift has the provisions by which you can automate tasks that have to be done repeatedly. This could be administrative tasks like generating, daily, weekly, or monthly reports. It could be resource and cost auditing. It can also be regular maintenance tasks to clean up data. You can automate all these with the provisions offered by Redshift.
- **Concurrent Scaling**. AWS Redshift will scale up automatically to support increasing concurrent workloads.

- **Query Volume**. The MPP technology shines in this aspect. You can send thousands of queries to the dataset at any given time. Still, Redshift will not slow down in any shape or form. It will dynamically allocate processing and memory resources to handle higher demand.
- **AWS Integration**. Redshift works well with the rest of the tools from AWS. You can set up the integrations between all the services according to your needs and optimal

- **Redshift API**. Redshift has a robust API with extensive documentation. It can be used to send queries and bain results using API tools. The API can also be used within a Python program for easier coding.
- **Security**. The security of the cloud is handled by Amazon and the security of the applications within the cloud has to be provided by users. Amazon provides provision for access control, data encryption, and virtual private cloud to provide an added level of security.
- **Machine Learning**. Redshift uses machine learning to predict and analyze queries. This, in addition to MPP, makes the performance of Redshift faster than other solutions in the market.

- **Easy Deployment**. A Redshift cluster can be deployed in any part of the world from anywhere in a matter of just minutes. You can have a high-performing data warehousing solution at the fraction of the price set by competitors in mere minutes.

- **Consistent Backup**. Amazon automatically backs up data regularly. This can be used to restore in the event of any faults, failures, or corruption. The backups are spread across different locations. So this eliminates the risk of faults at a location as a whole.
- **AWS Analytics**. AWS offers plenty of analytical tools. All of these can work well with Redshift. Amazon provides support to integrate other analytical tools with Redshift. Redshift has native integration capabilities with AWS analytics services.
- **Open Formats**. Redshift supports and can provide outputs in many open formats for data. The most common formats supported are Apache Parquet and Optimized Row Columnar (ORC) file formats.
- **Partner Ecosystem**. AWS is one of the oldest cloud service providers. A lot of customers depend on Amazon for their infrastructure. In addition to that AWS has a strong network of partners that builds third-party applications and offers implementation services. This partner ecosystem can also be tapped to see if you can find an implementation solution that is perfect for your organizations.

The data collected will grow every day. Redshift is a hedge against the growing data with increasing analytical complexity. It can be used to build infrastructure that lasts into the future.

Additionally, Redshift offers best in class performance at a fraction of the cost of competitors. This makes it a value proposition for any organization that has to handle large volumes of data.

1.4 LIMITATIONS OF AWS REDSHIFT

Redshift has some drawbacks that need to be considered before choosing it as your data warehousing solution.

- **Parallel Uploads**. Redshift does not support all databases for parallel upload. Amazon S3, EMR, and DynamoDB are supported by Redshift for parallel uploads

using ultra-fast MPP. For other sources, separate scripts have to be used to upload data. This can be a very slow process.

- **Uniqueness**. One of the basic tenets of a database is to have unique data and avoid redundancies. AWS Redshift does not provide any tool or means to ensure the uniqueness of data. If you are migrating overlapping data from different sources to Redshift, there will be redundant data points.
- **Indexing**. This becomes a problem when Redshift is used for data warehousing needs. Redshift uses distribution and sorts keys to index and store data. You will need to know the concepts behind the keys to work on the database. AWS does not provide any system to change the keys or manage them with minimal knowledge.
- **OLAP Limitations**. OLAP databases (which Redshift is) are optimized for analytical queries on a large volume of data. Compared to traditional OLTP (Online Transaction Processing) databases, OLAP lacks in performing basic database tasks. Insert/update/delete operations have performance limitations in OLAP databases. It is often easier to recreate a table with changes than to insert/update tables in Redshift. While OLAP works well with static data, OLTP databases perform better for data modification operations.
- **Migration cost**. Redshift is used in cases where the data to be stored or worked with is humongous. It will at least be in the range of petabytes. At this level, bandwidth becomes a problem. You will need to transfer this data to AWS locations before you can begin the project. This could be a potential problem for businesses that have network caps for bandwidth. The additional cost will have to be borne by the user. AWS does provide the option to send the data using physical storage devices.

1.5 AWS REDSHIFT PRICING MODEL

AWS offers a very flexible pricing scheme for Redshift. The price starts at $0.25 per hour for a terabyte of data and it can be scaled from there. First, you need to opt for the node type you want. AWS Redshift offers three types of nodes.

- **RA3 nodes with managed storage**. Here, you will need to pick the level of performance you require, and the managed storage will be billed on a pay-as-you-go basis. The number of RA3 clusters you have to choose will depend on the amount of data processed on a daily basis.

- **DC2 Nodes**. These should be chosen when you need high performance. Local SSD (Solid State Drive) storage is included with the nodes. You will need to add more nodes when the size of the data grows. DC2 nodes are best suited when the data is relatively small in size and needs superlative performance.
- **DS2 nodes**. It should be opted for when there is a large data set that needs to be stored. DS2 provides only HDD (Hard Disk Drives) and has a slower performance compared to other nodes.

	vCPU	Memory	Addressable storage capacity	I/O	Price
Dense Compute DC2					
dc2.large	2	15 GiB	0.16TB SSD	0.60 GB/s	$0.33 per Hour
dc2.8xlarge	32	244 GiB	2.56TB SSD	7.50 GB/s	$6.40 per Hour
Dense Storage DS2					
ds2.xlarge	4	31 GiB	2TB HDD	0.40 GB/s	$1.25 per Hour
ds2.8xlarge	36	244 GiB	16TB HDD	3.30 GB/s	$10.00 per Hour
RA3 with Redshift Managed Storage*					
ra3.xlplus	4	32 GiB	32TB RMS	0.65 GB/s	$1.202 per Hour
ra3.4xlarge	12	96 GiB	128TB RMS	2.00 GB/s	$3.606 per Hour
ra3.16xlarge	48	384 GiB	128TB RMS	8.00 GB/s	$14.424 per Hour

Table .1.1

Redshift also has a pay-as-you-go pricing model according to the requirements.

- **Amazon Redshift spectrum pricing**. When you need to run SQL queries on a large dataset in an S3 data lake, you pay according to the usage. Even if the data in S3 is of exabyte range, you will only need to pay according to the amount of data scanned. The price was $5 per terabyte scanned for the North California location.
- **Concurrency scaling pricing**. Concurrency scaling enables you to dynamically allocate resources according to the demand. AWS automatically provides additional resources even if the number of queries and users multiply. You will only need to pay according to the usage. In addition, each cluster has one scaling credit every day. This will be sufficient for 97% of the customers according to past AWS data.
- **Redshift managed storage pricing**. This pricing model will divide the computing and storage costs of RA3 nodes. That way you need not add more nodes when data

requirement increases. RA3 nodes are costlier for storage purposes compared to using separate managed storage.

- **Redshift ML**. You can use SQL queries to train ML models. You will be able to use the free credits of Amazon Sagemaker before you have to pay for creating ML models.

CHAPTER 2

SYSTEM ARCHITECTURE

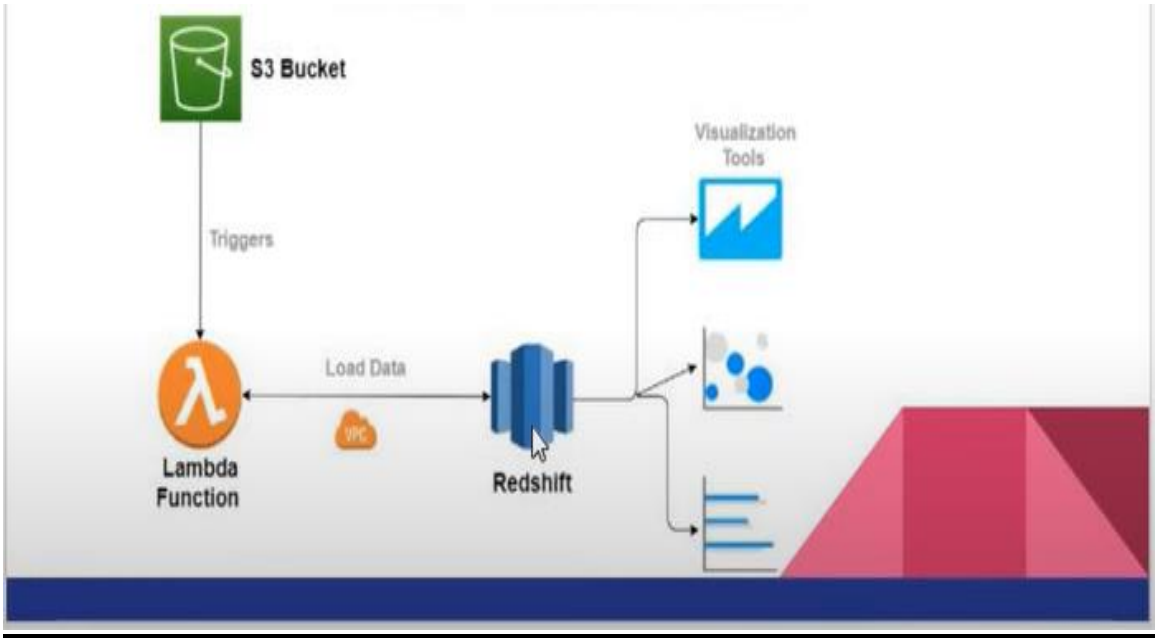

Fig 2.1

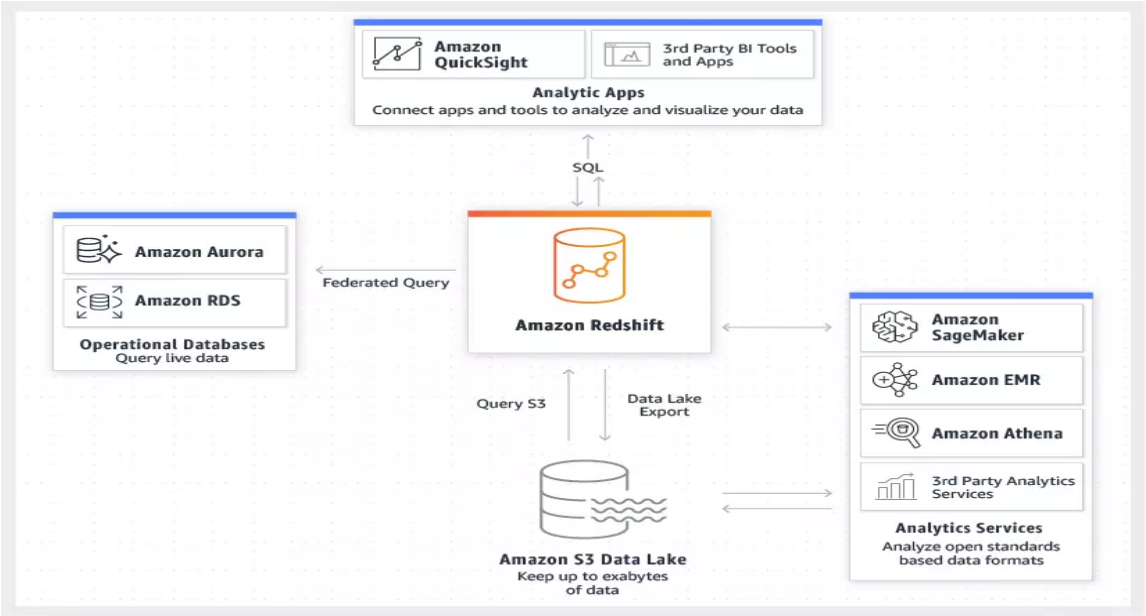

Fig 2.2

CHAPTER 3

IMPLEMENTATION

STEP 1 : Create an IAM role.

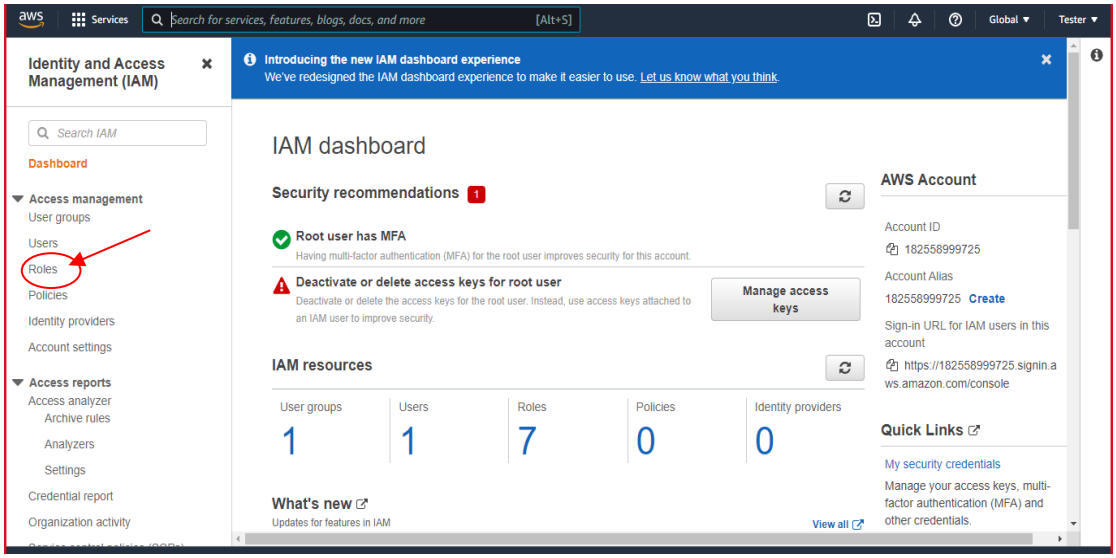

Fig 3.1.1

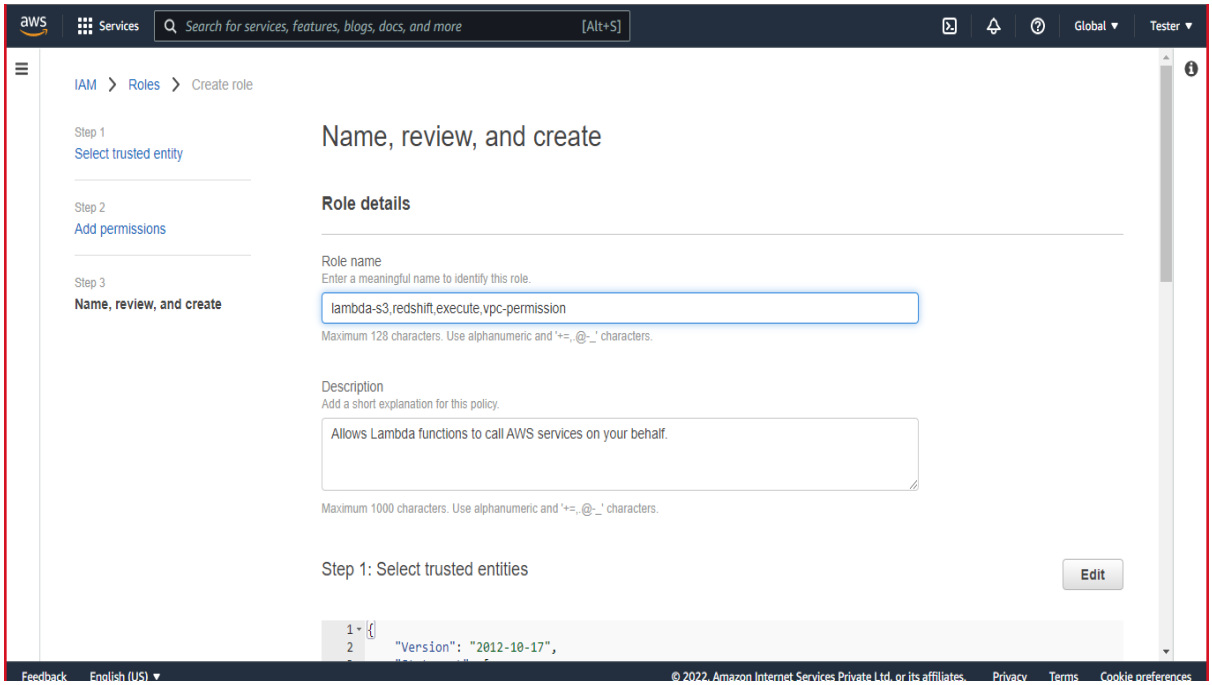

Fig 3.1.2

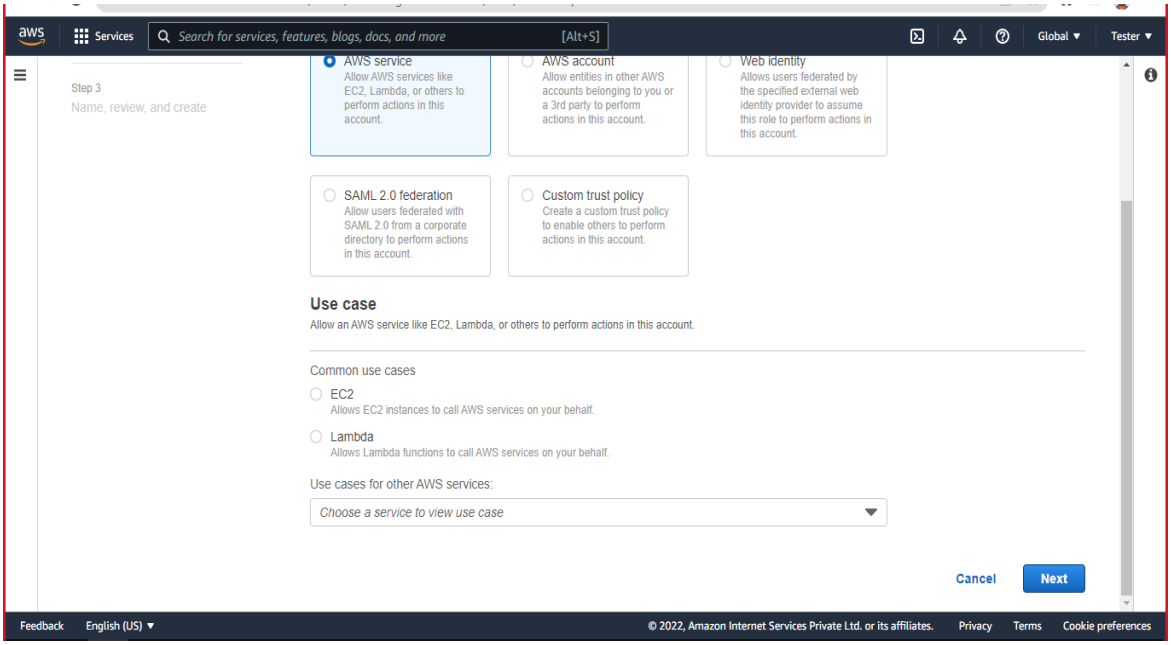

Fig 3.1.3

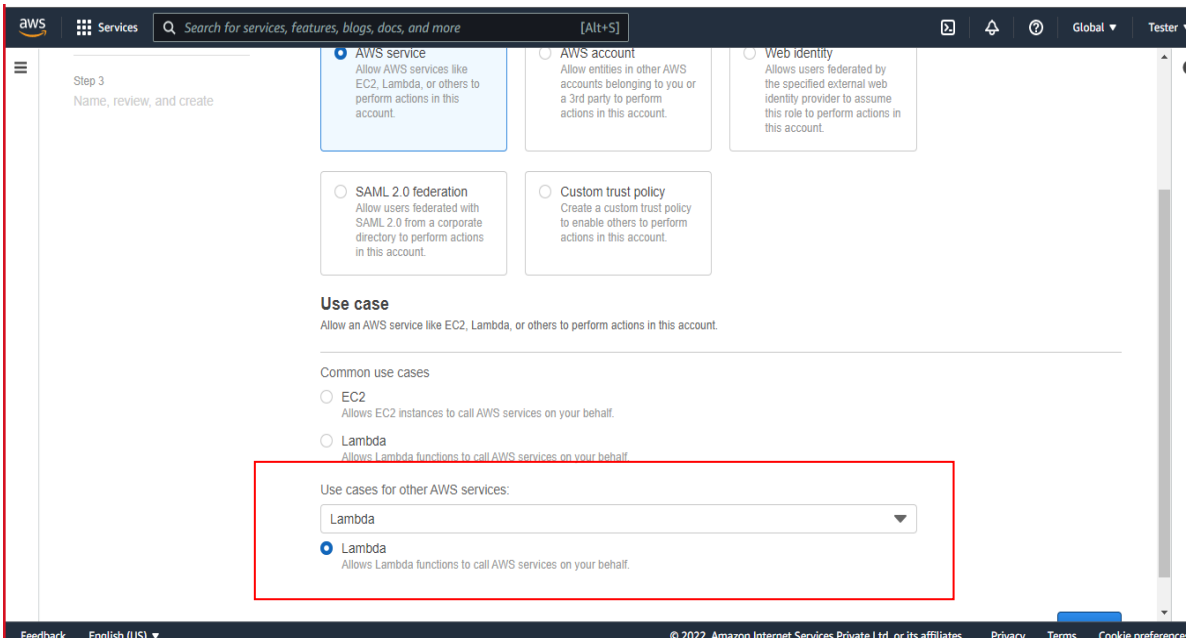

Fig 3.1.4

Allow the these permissions :

- awss3fullaccess
- amazonredshiftfullaccess
- awslambdaexecuted
- awslambdavpcaccessexcecuted

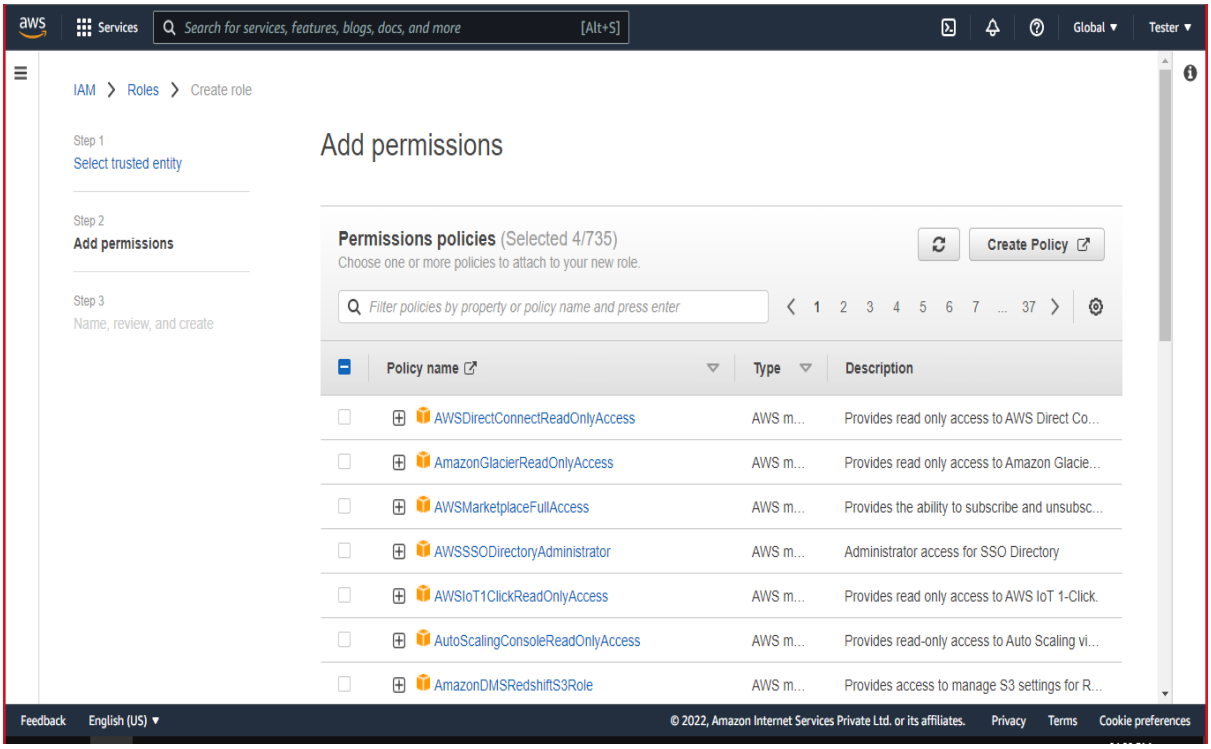

Fig 3.1.5

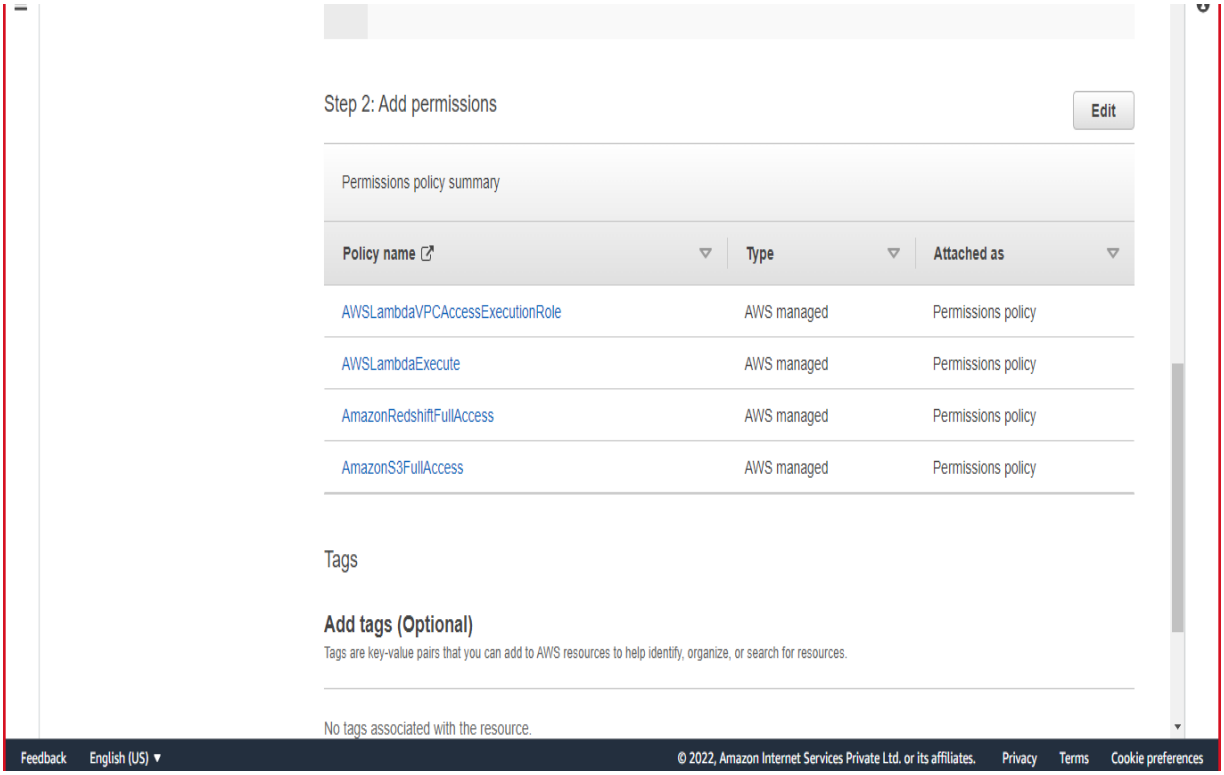

Fig 3.1.6

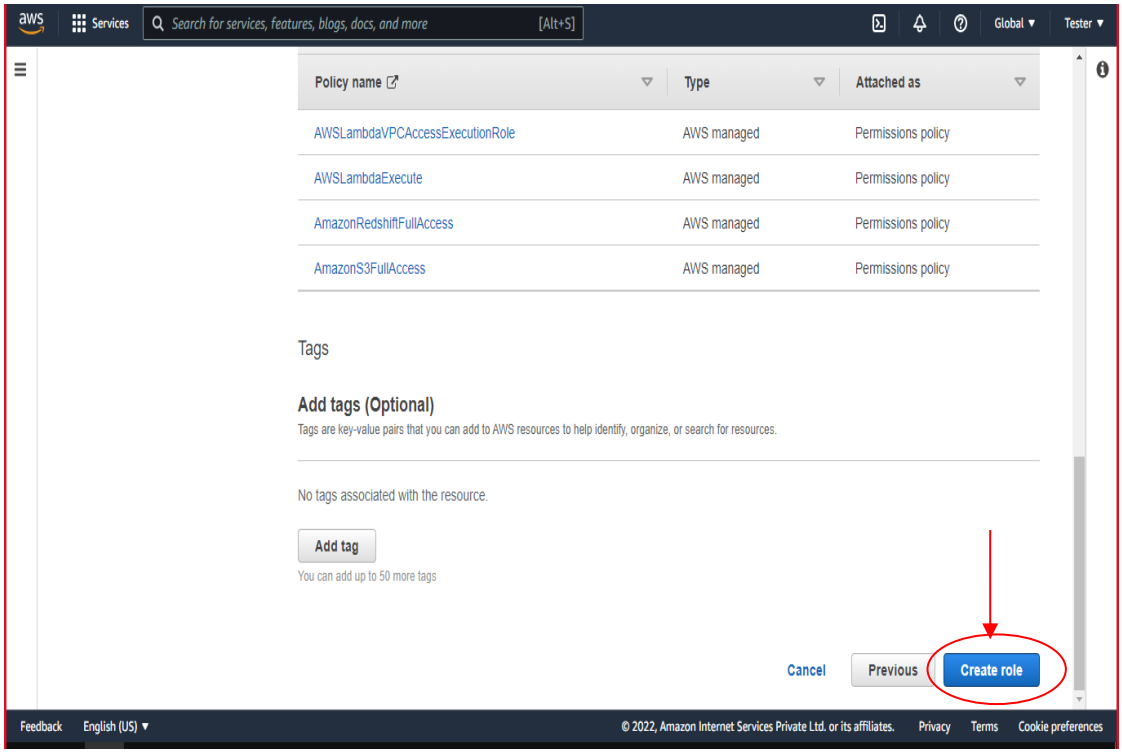

Fig 3.1.7

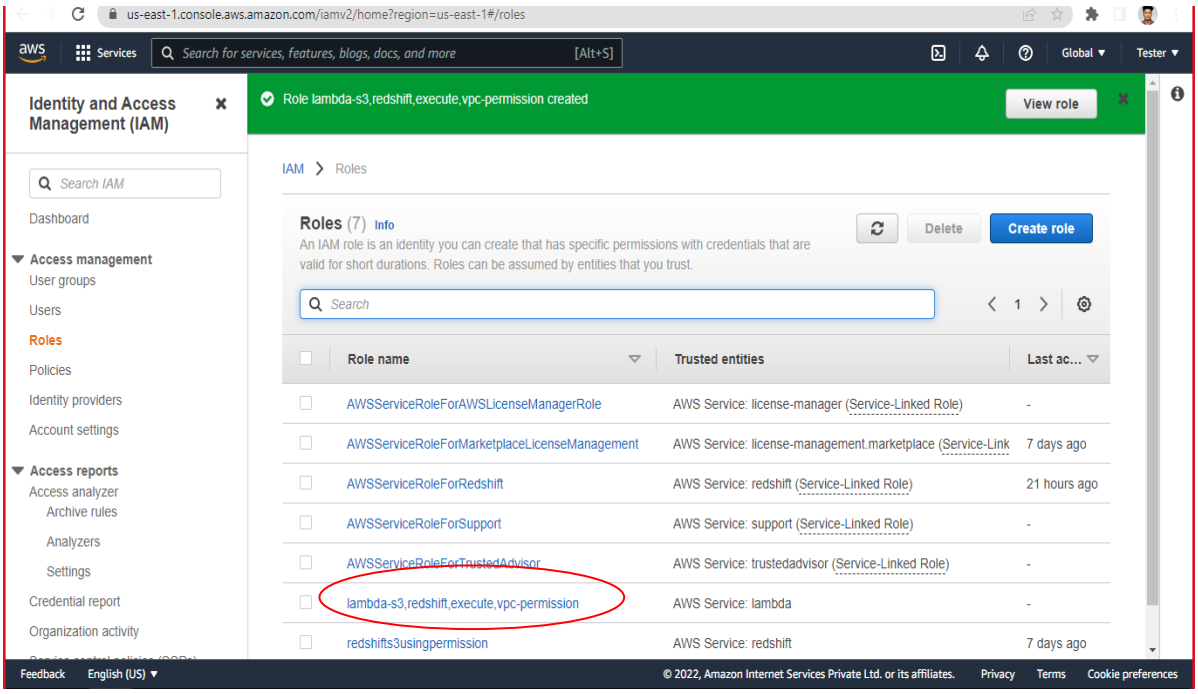

Fig 3.1.8

STEP 2 : Create an S3.

In this section, we have to create storage space. In s3 bucket we can do with basic configuration.

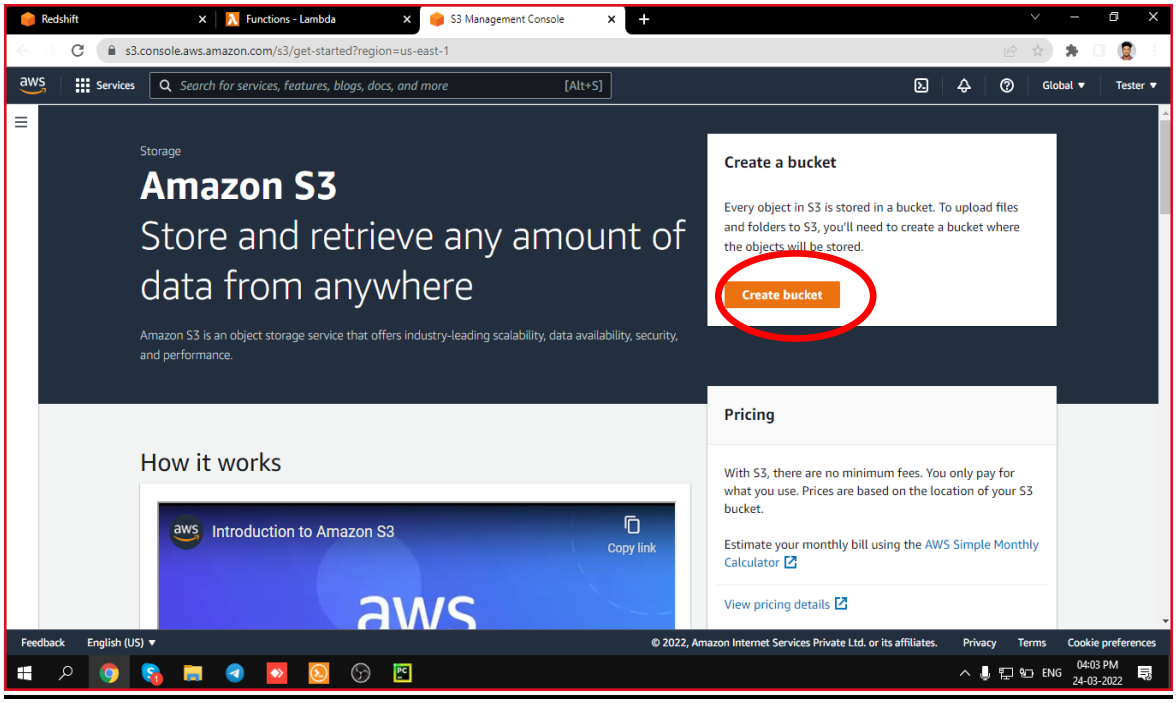

Fig 3.2.1

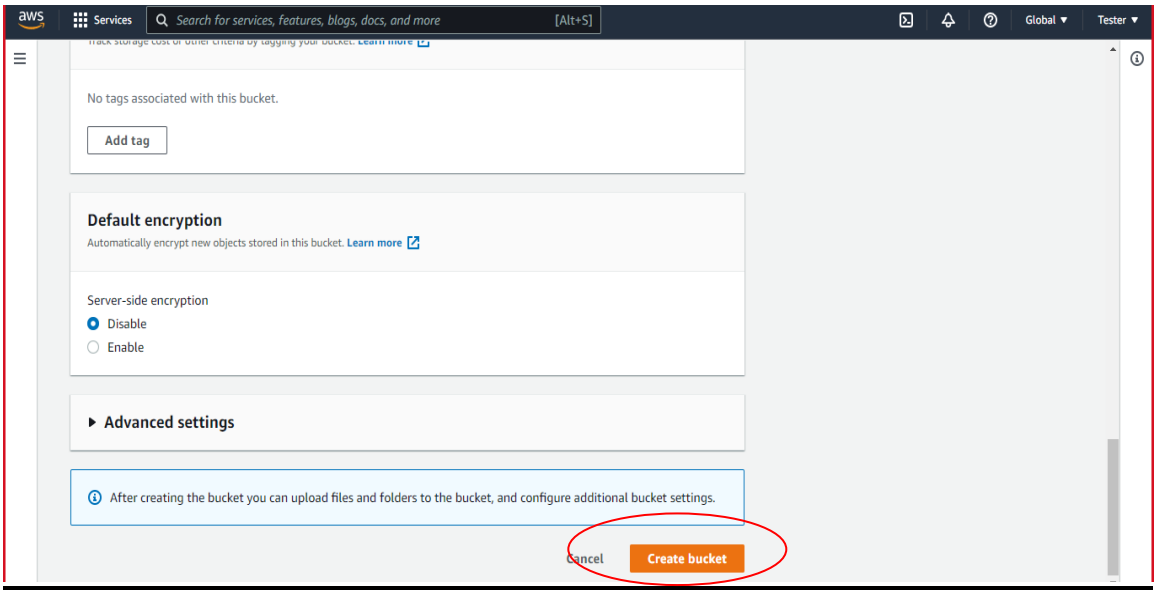

Fig 3.2.2

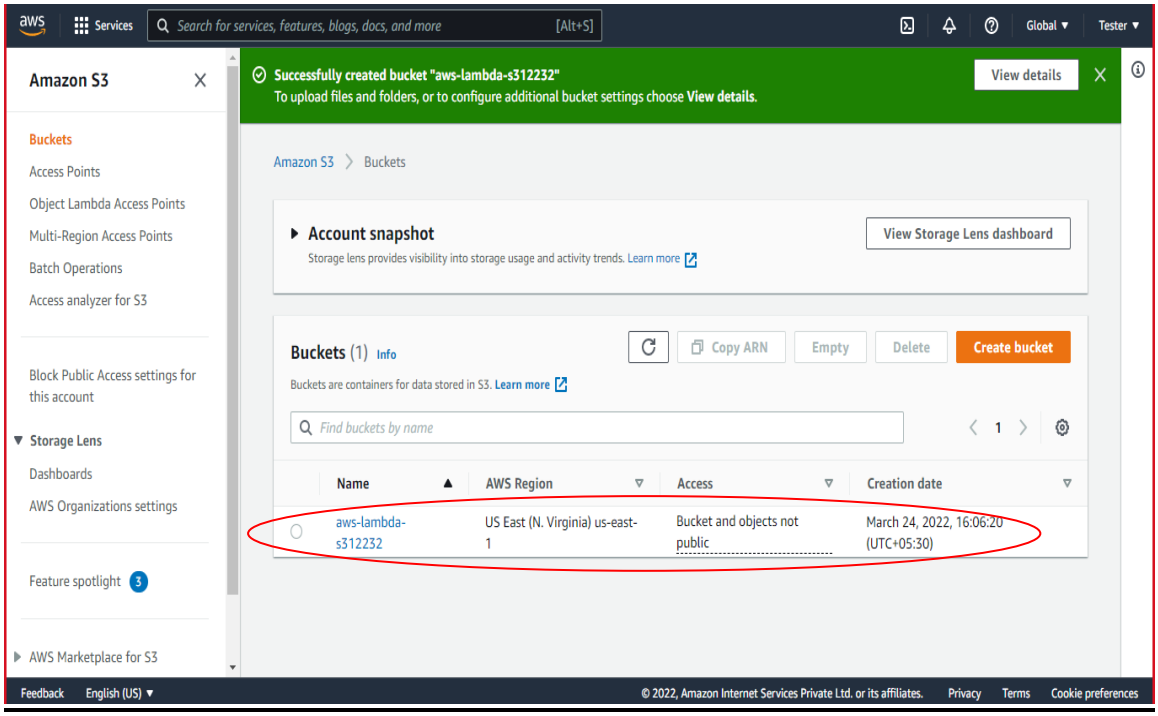

Fig 3.2.3

STEP 3 : Create an Lambda Function.

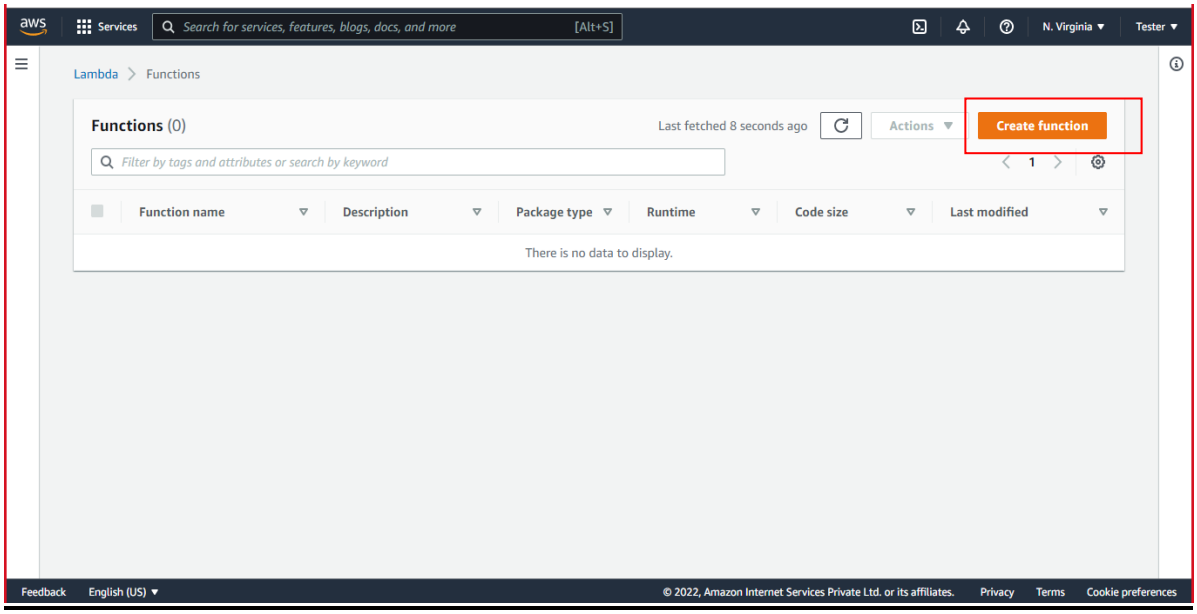

Fig 3.3.1

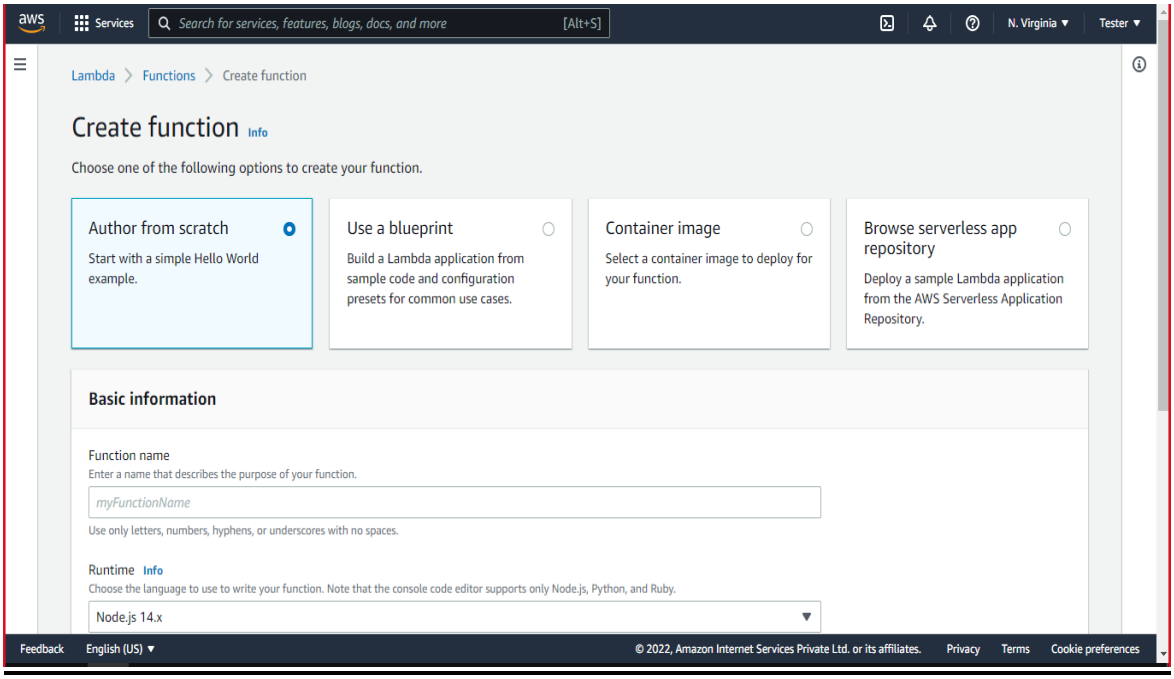

Fig 3.3.2

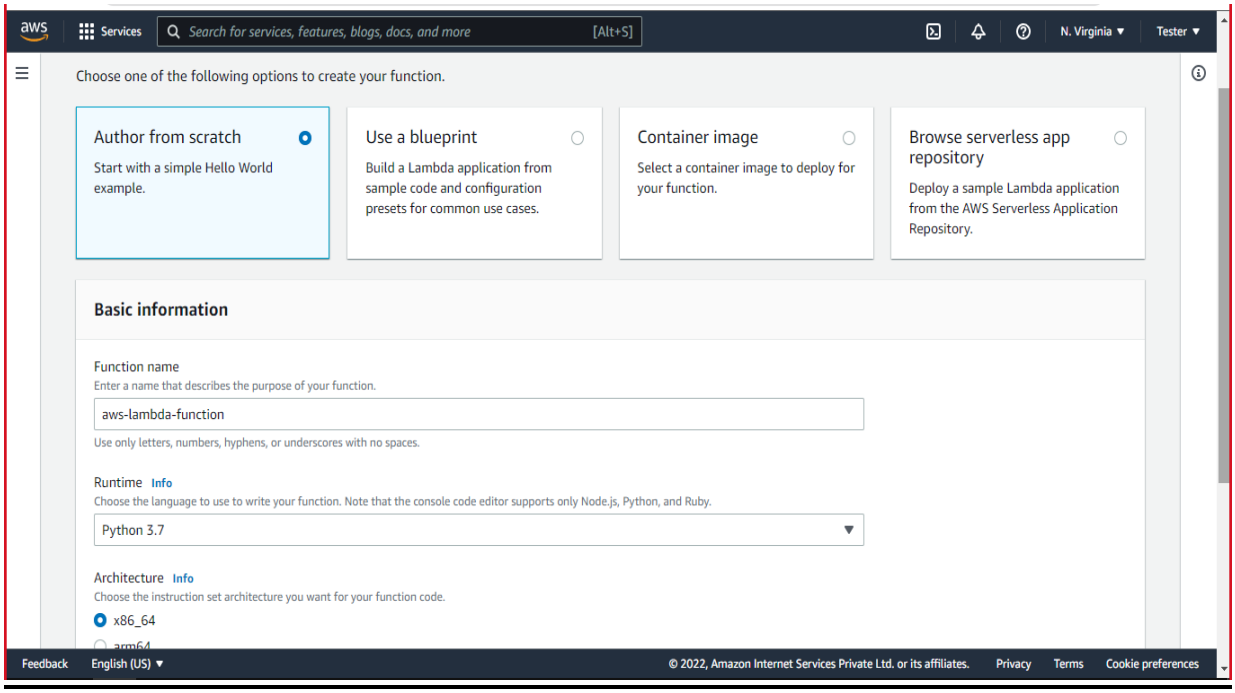

Fig 3. 3.3

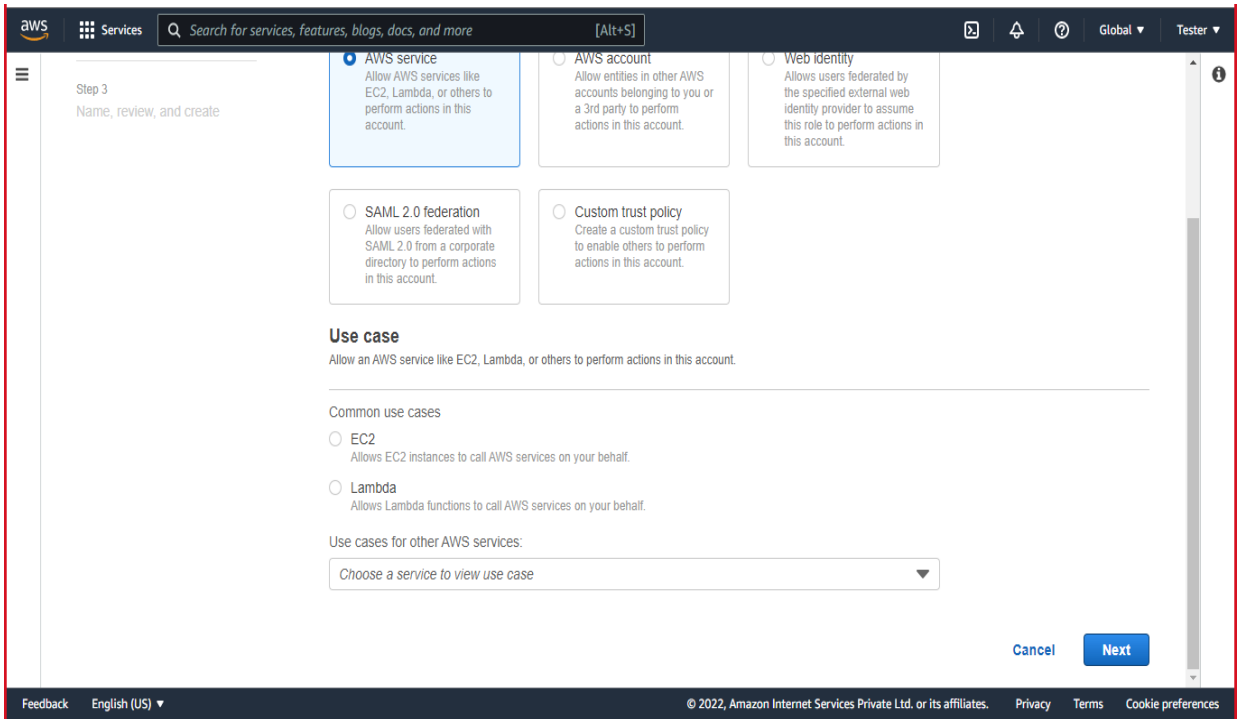

Fig 3.3.4

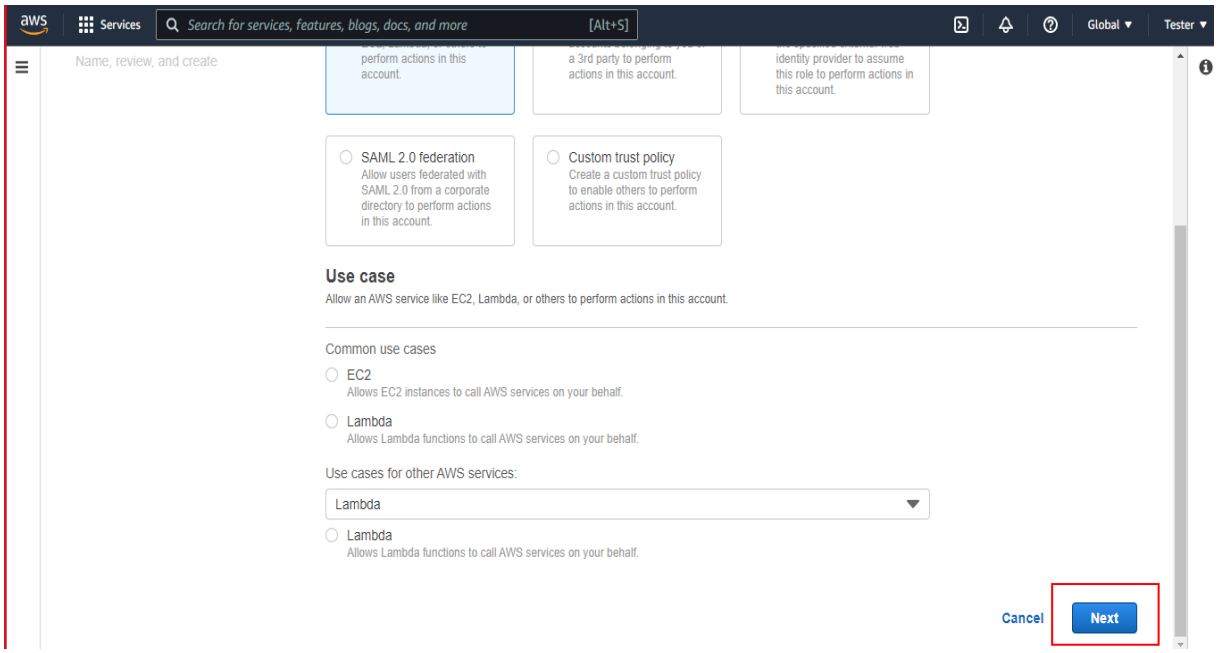

Fig 3.3.5

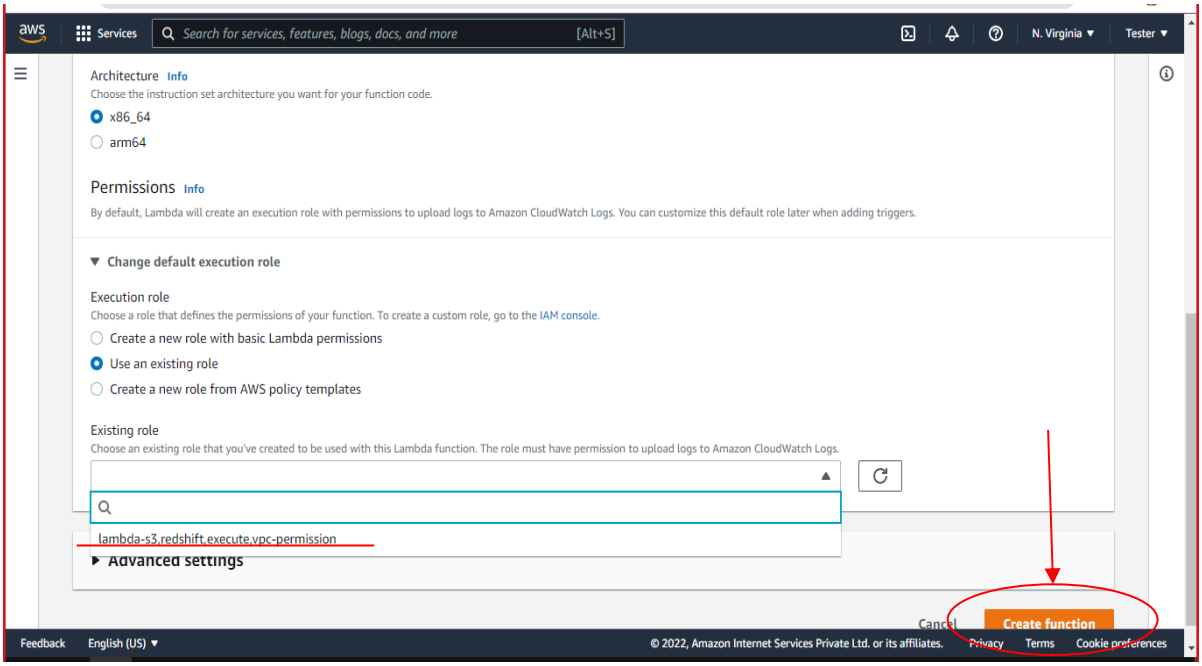

Fig 3.3.6

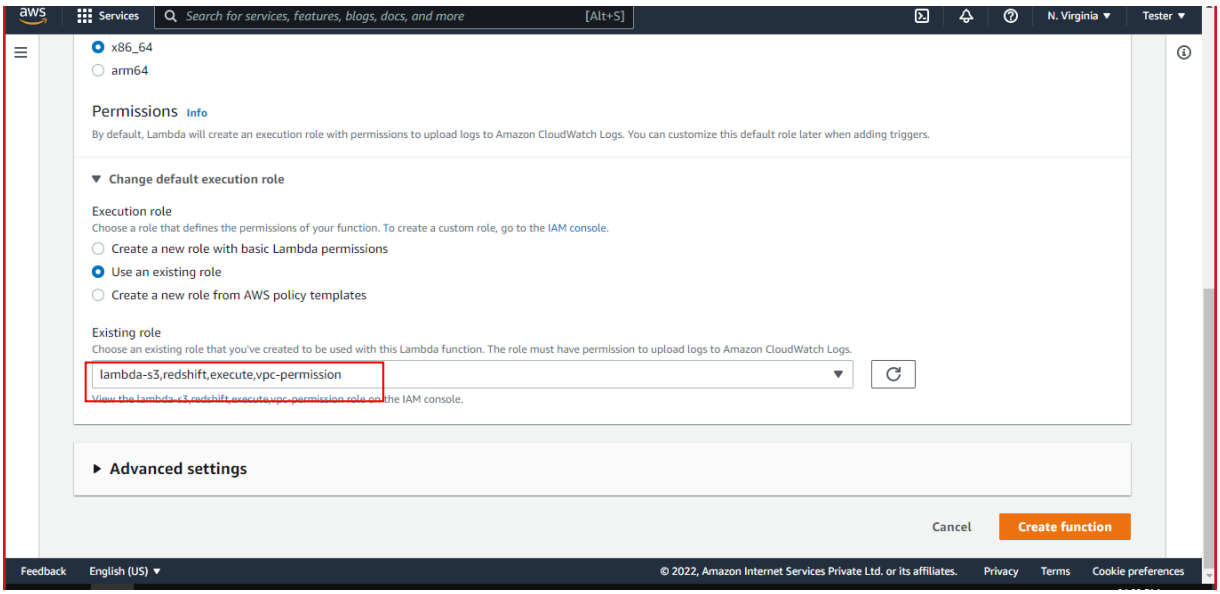

Fig 3.3.7

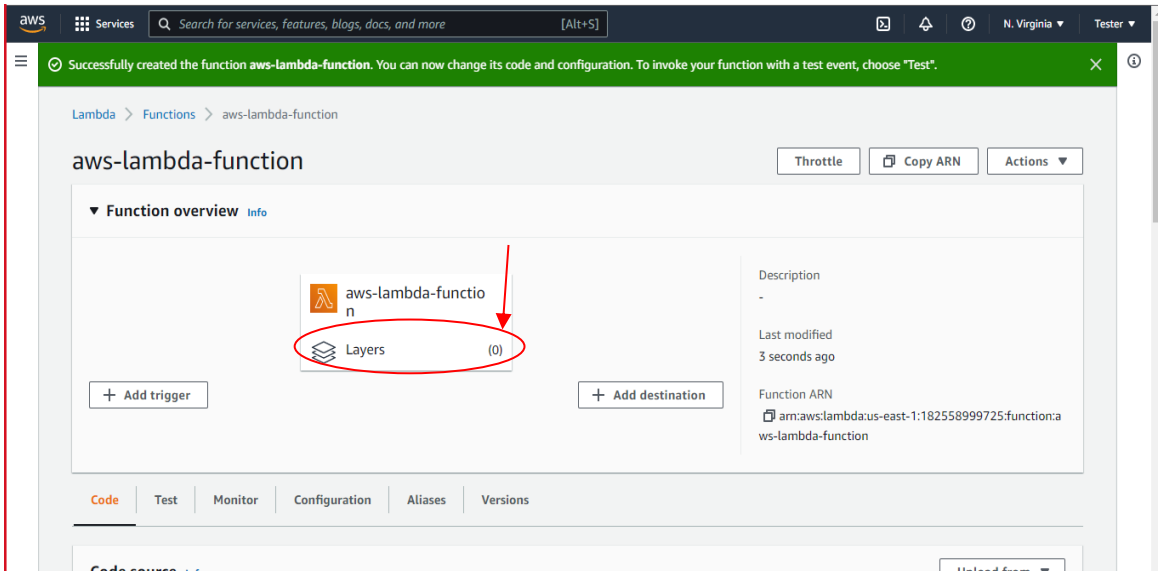

Fig 3.3.8

STEP 4 : Create an Lambda Function's Layer.

If we want to execute this code (fig). we need to get some python library files.

example: psycopg2. therefore, zip that code and put it as a layer.

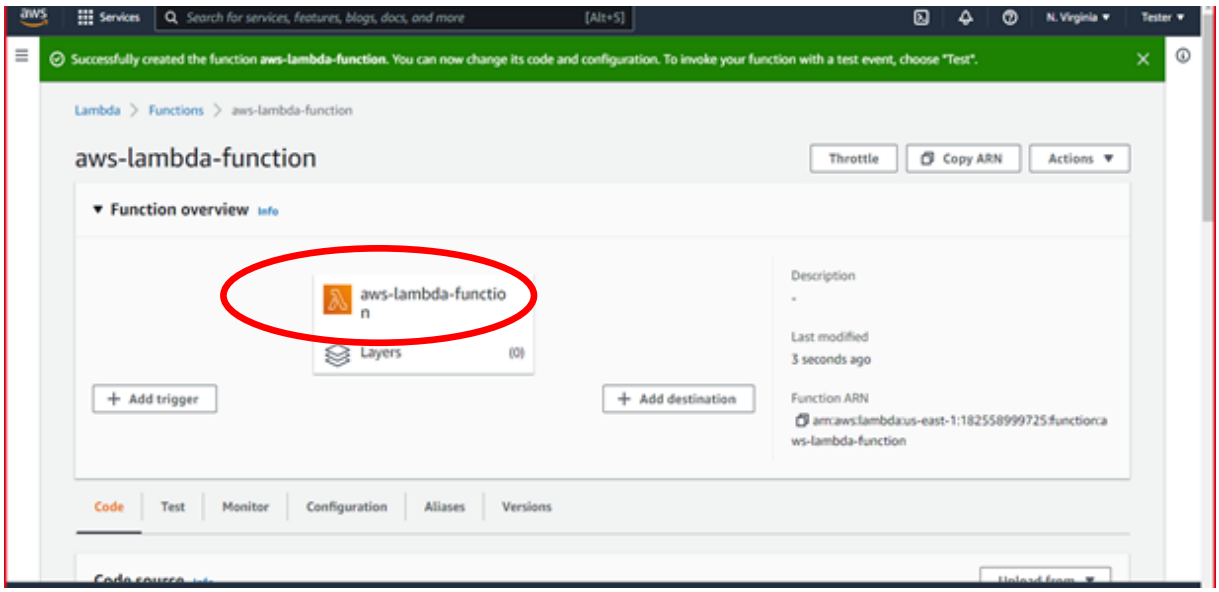

Fig 3.4.1

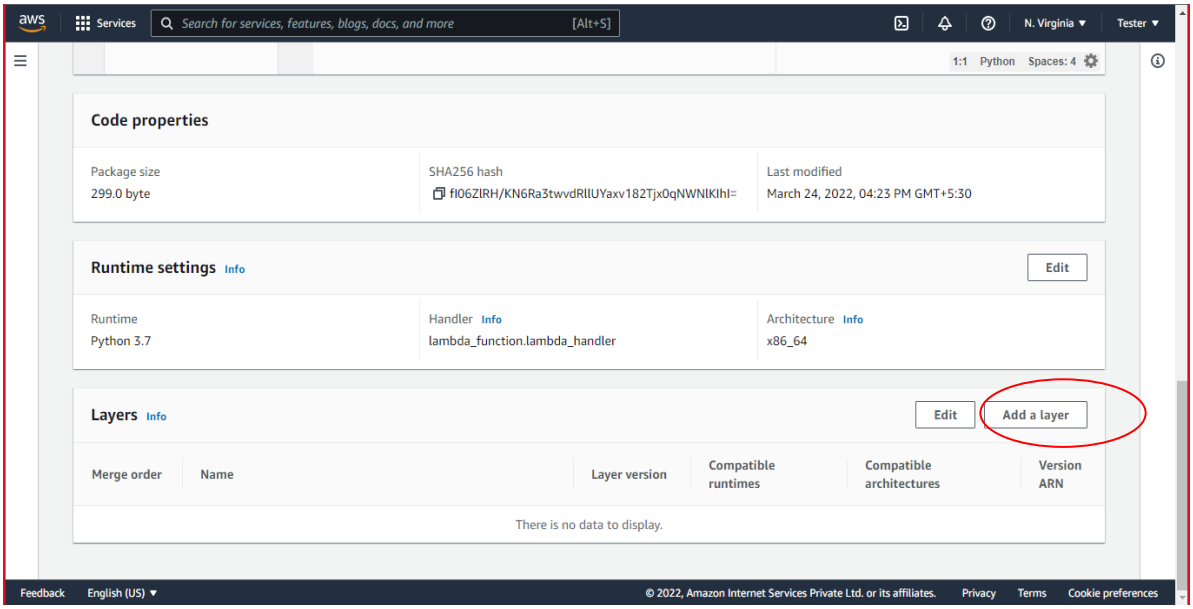

Fig 3.4.2

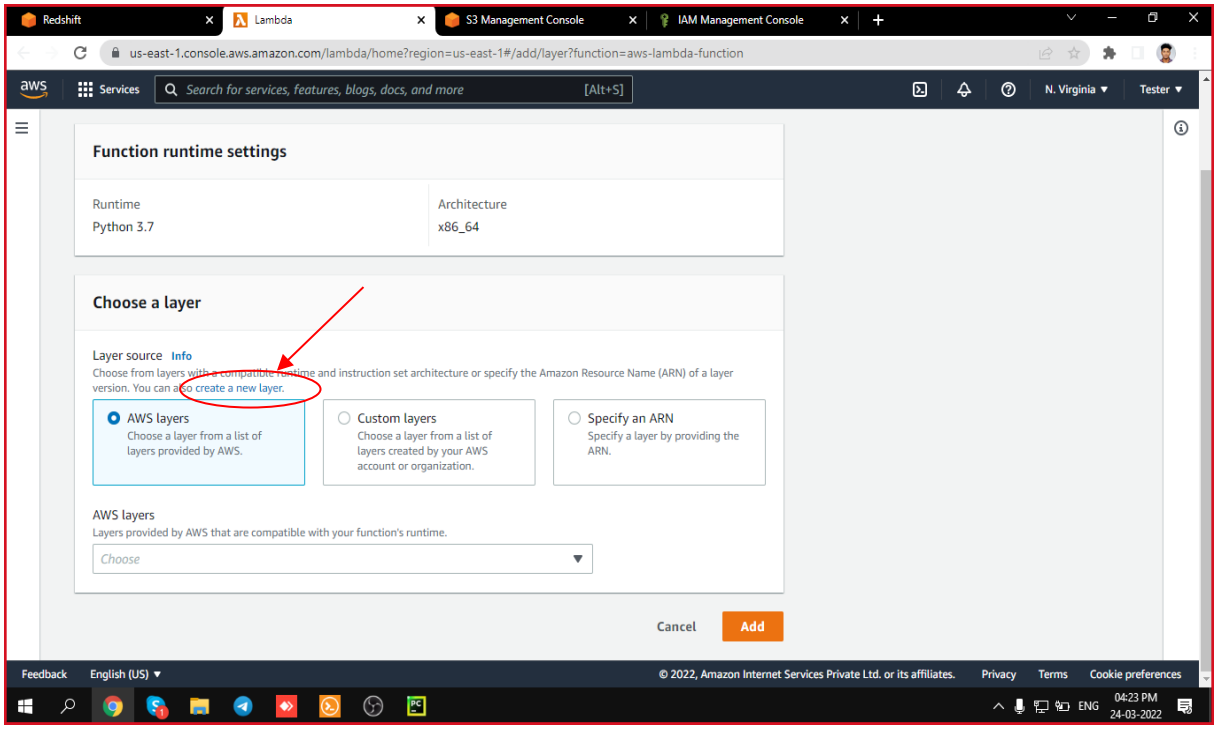

Fig 3.4.3

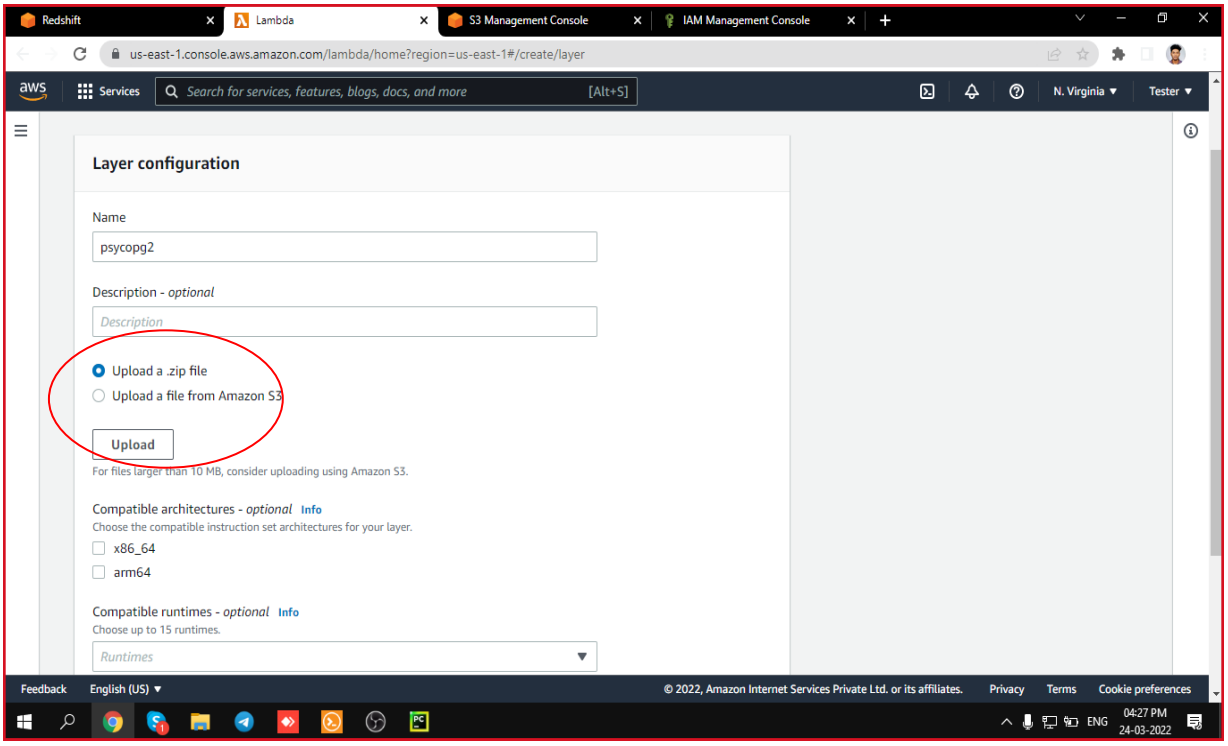

Fig 3.4.4

Download or clone for our github: https://github.com/Amal-p/Redshift.git

Selected python.zip and update

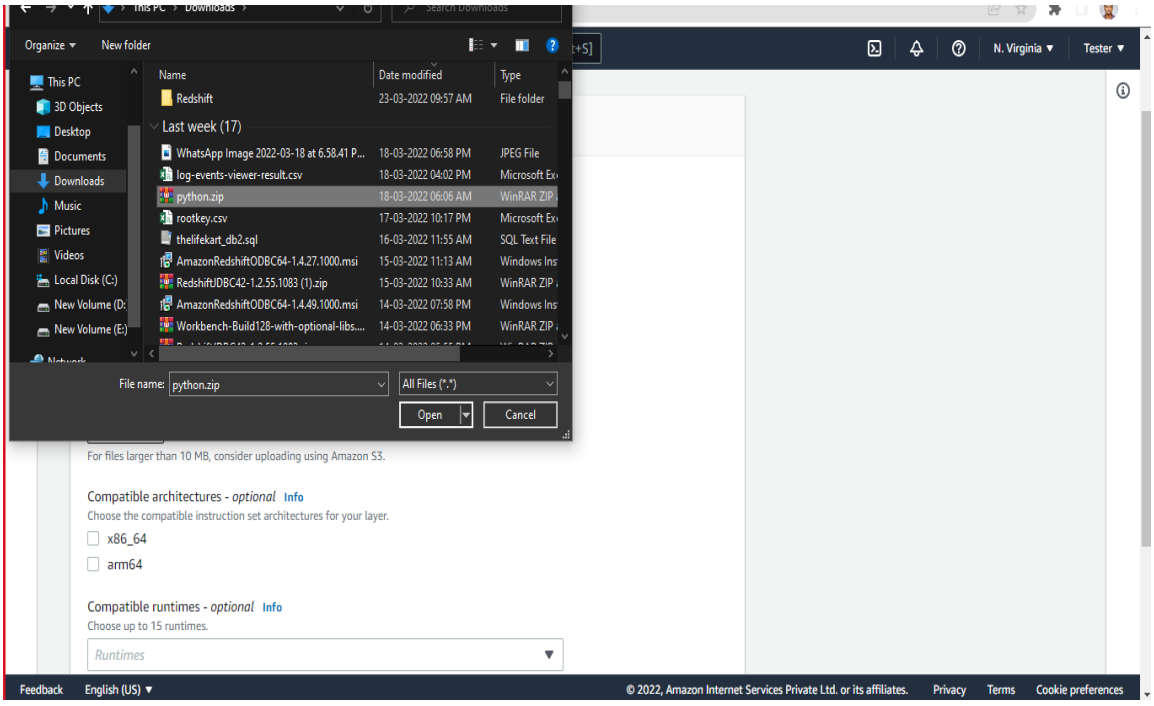

Fig 3.4.5

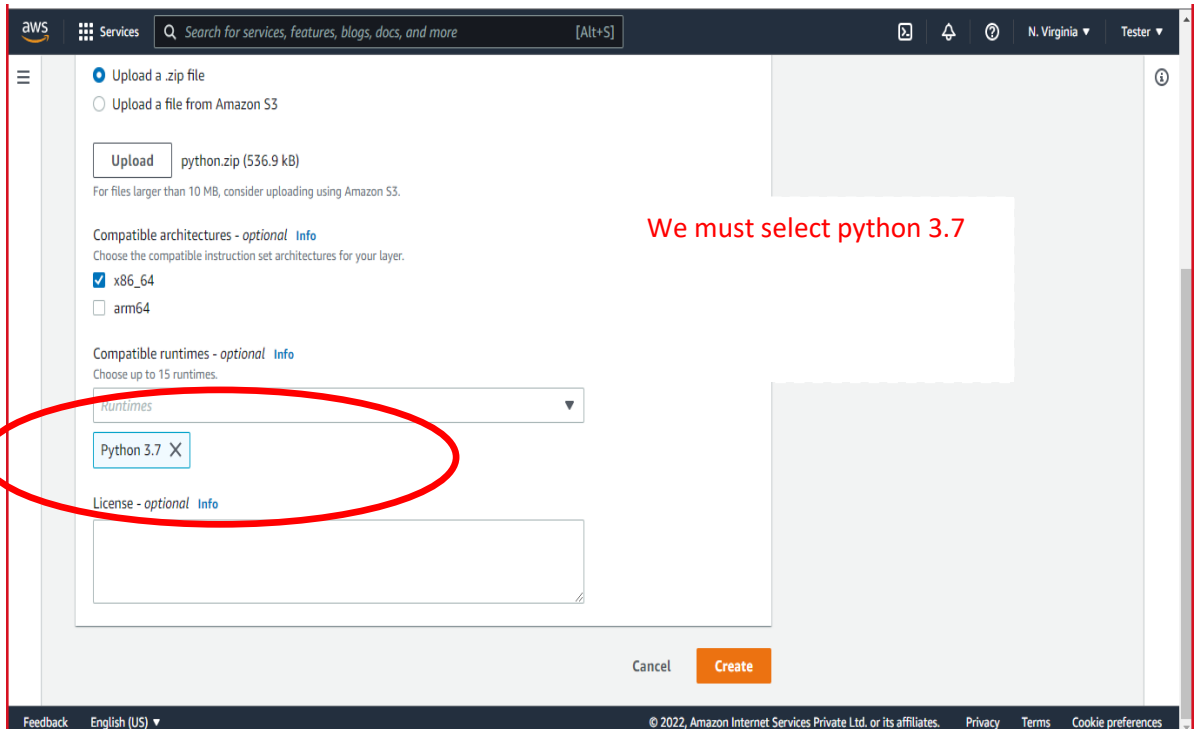

Fig 3.4.6

STEP 5 : Add Layer.

Note : The credential may be changed by you give it to the layer(version..etc).

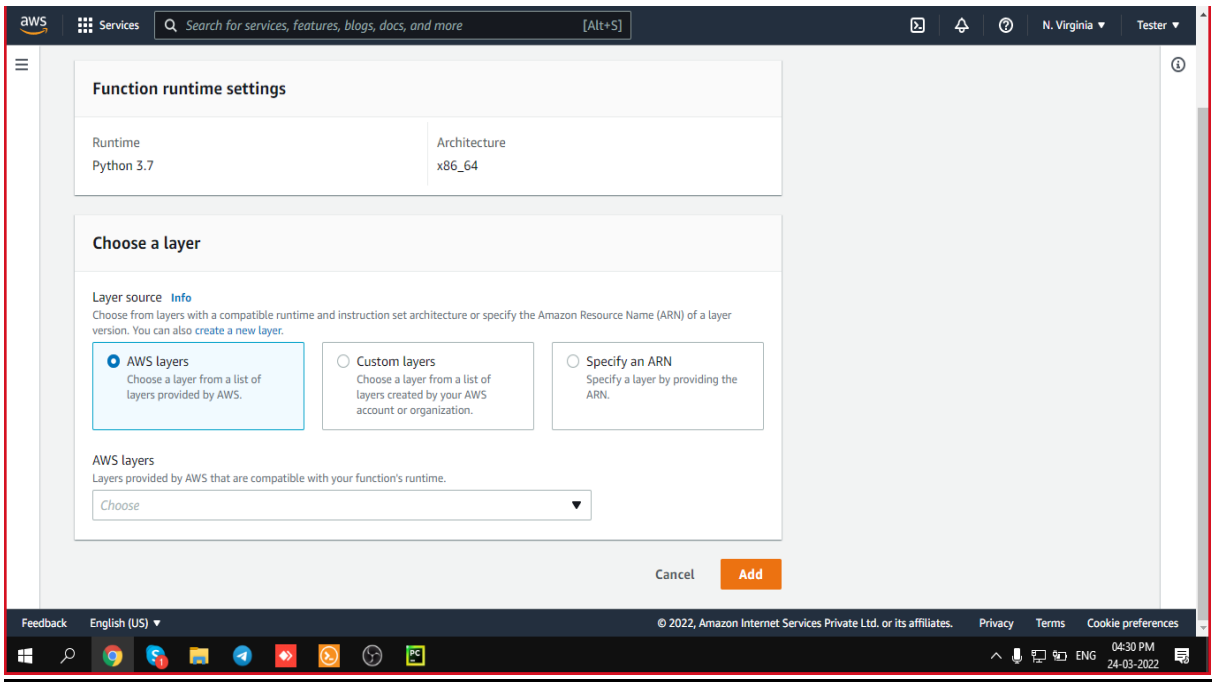

Fig 3.5.1

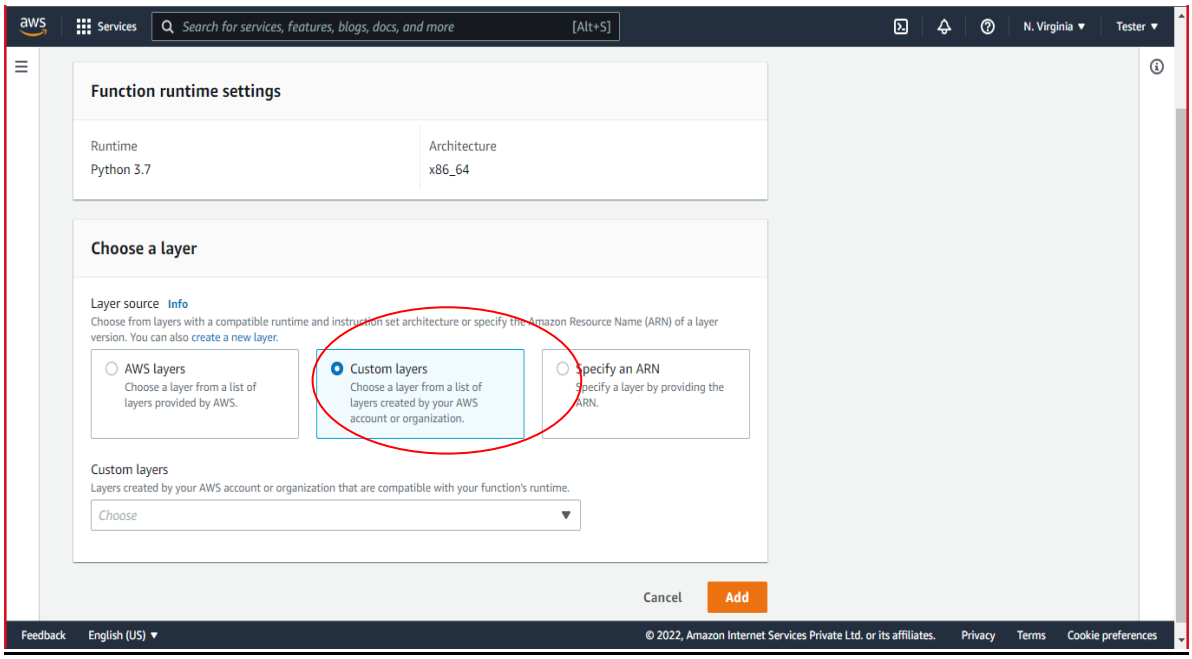

Fig 3.5.2

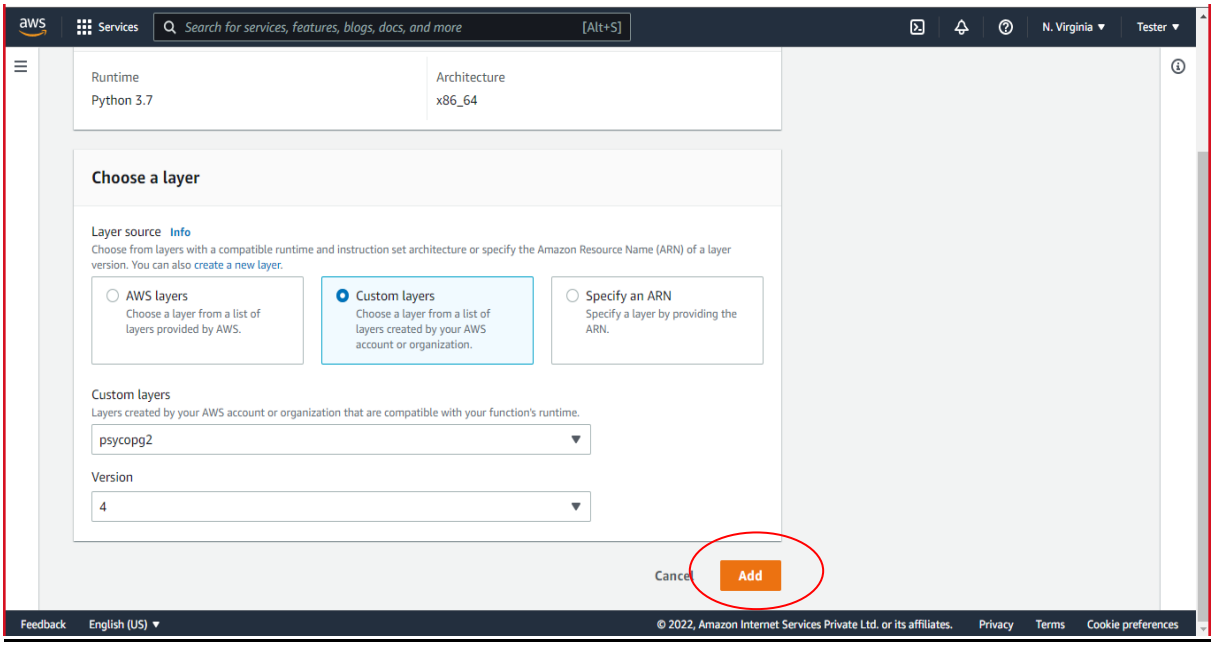

Fig 3.5.3

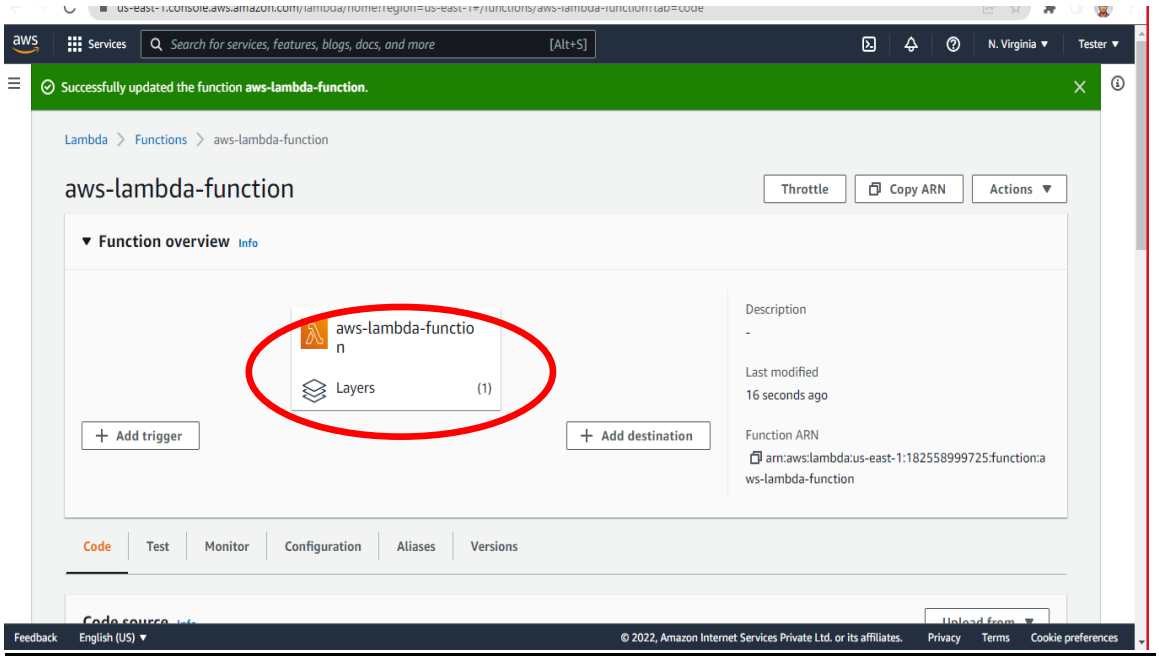

Fig 3.5.4

STEP 6 : Trigger Setup S3 – Lambda.

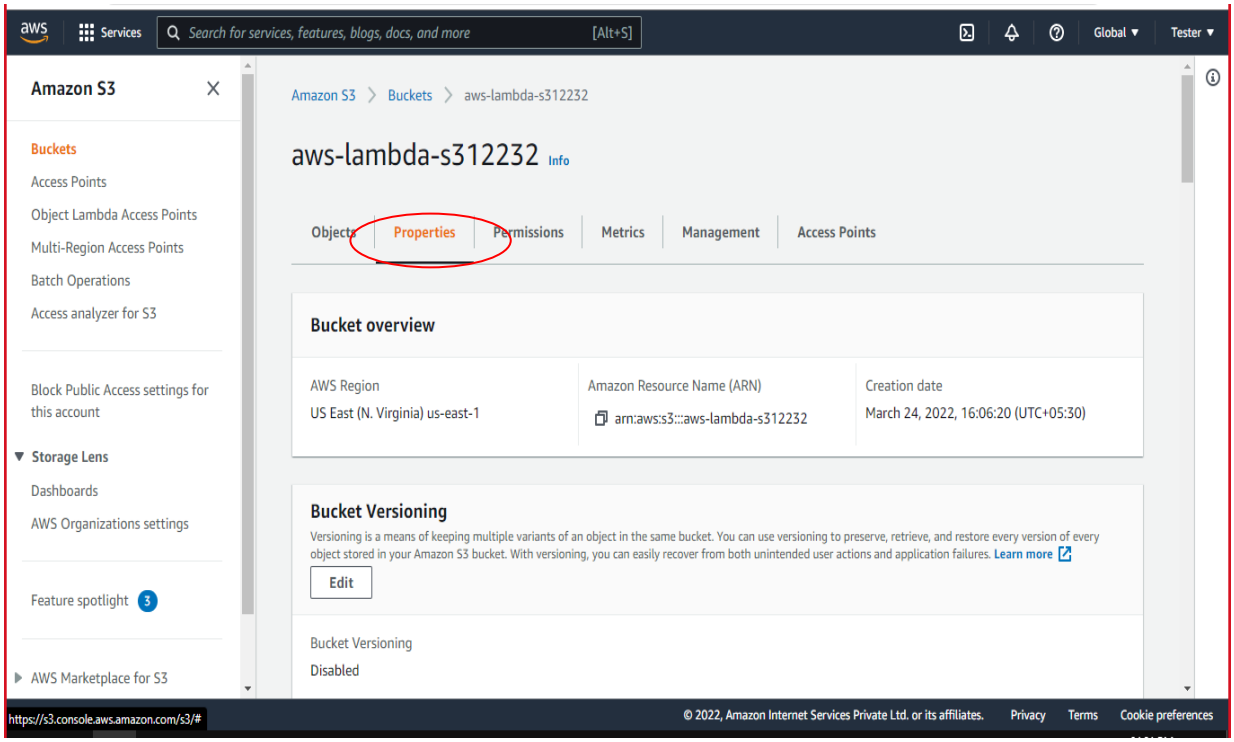

Fig 3.6.1

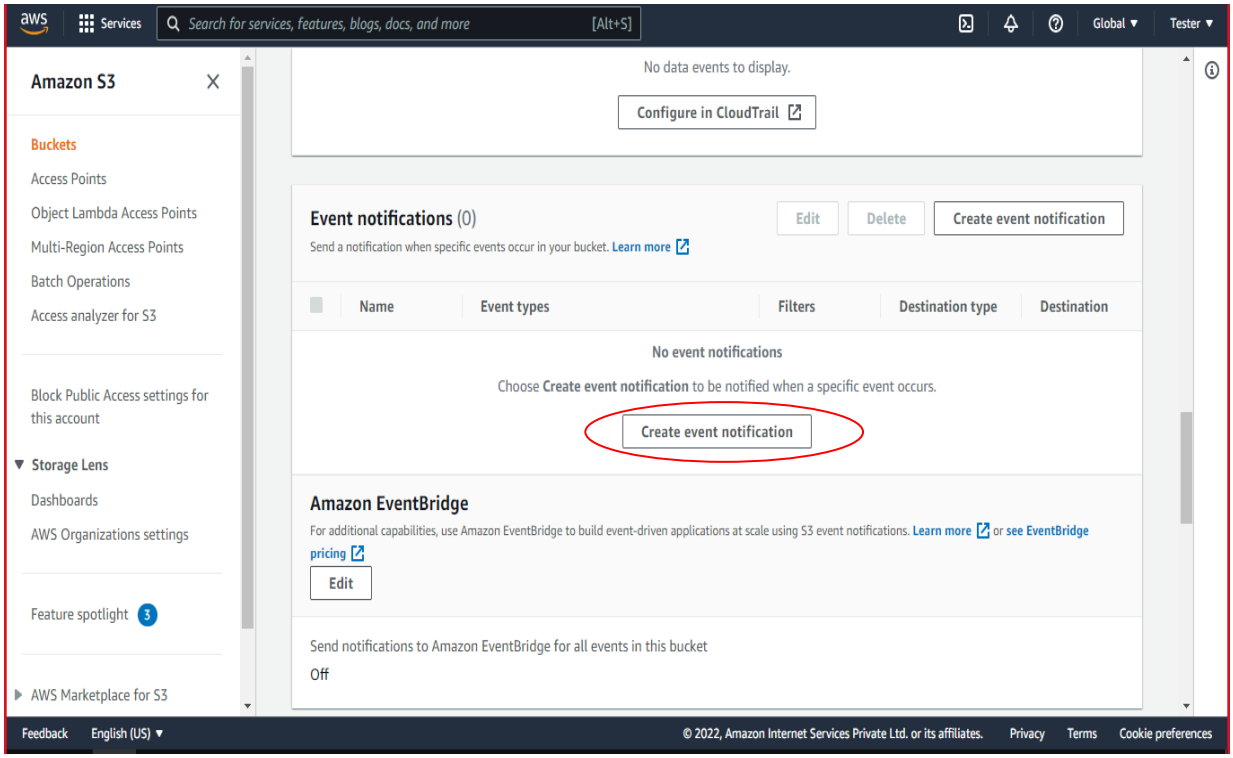

Fig 3.6.2

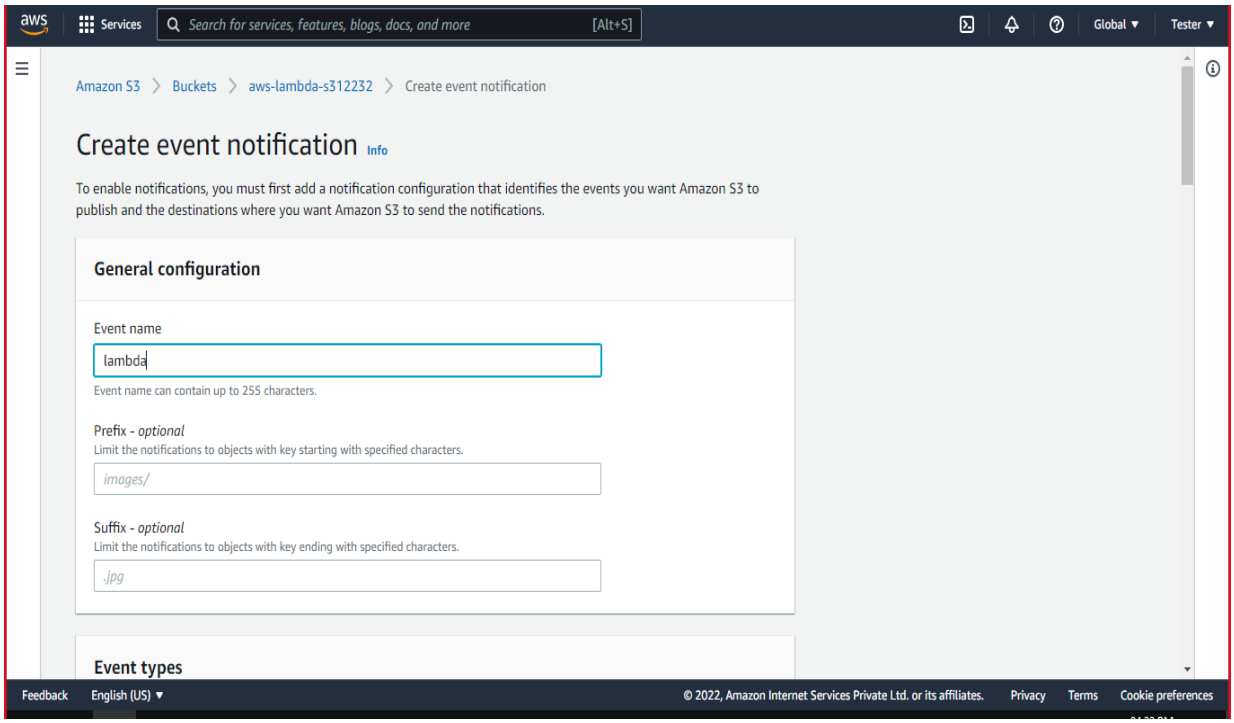

Fig 3.6.3

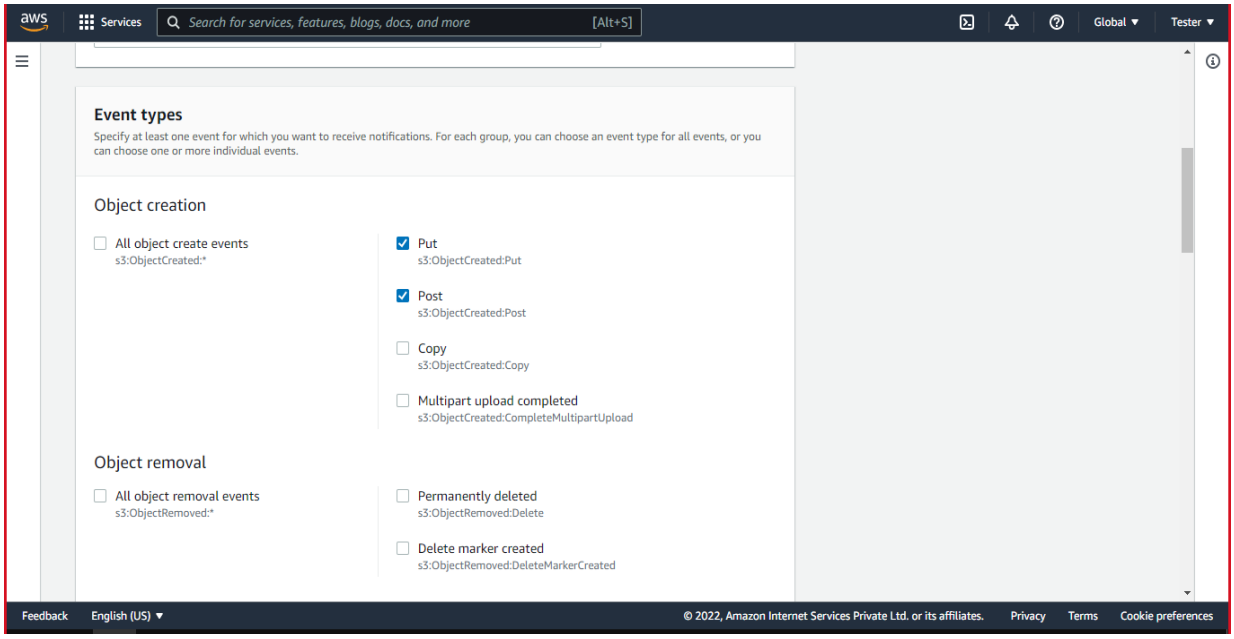

Fig 3.6.4

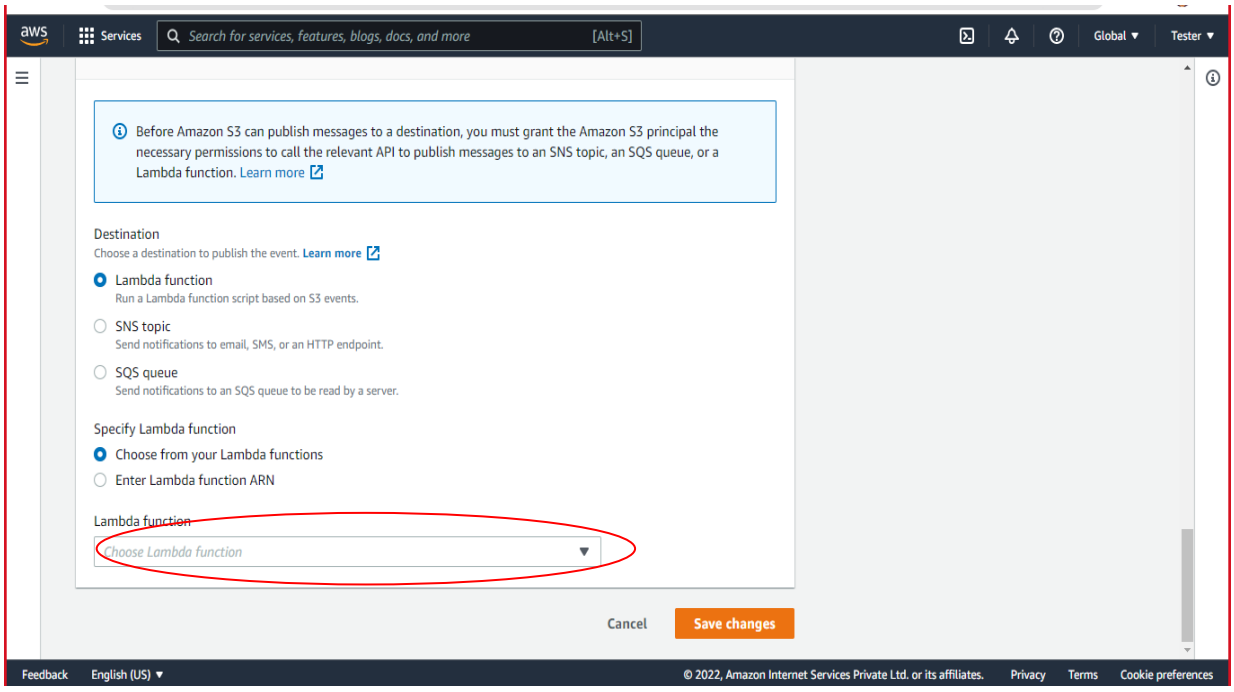

Fig 3.6.5

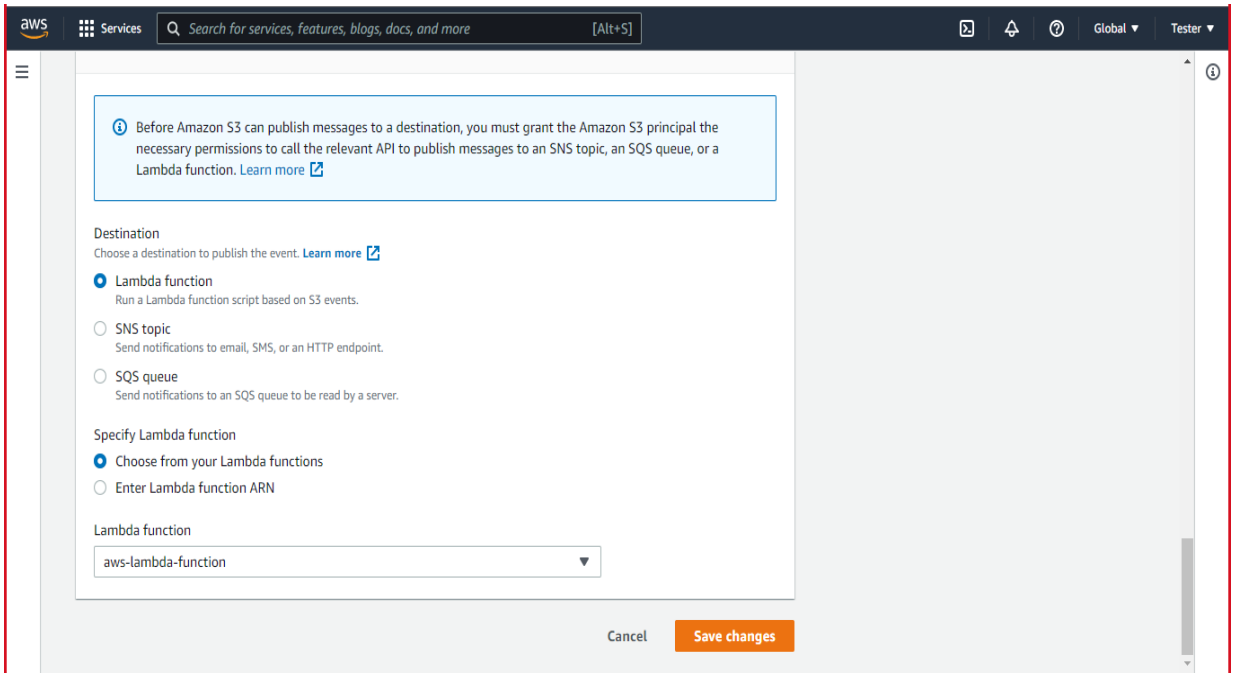

Fig 3.6.6

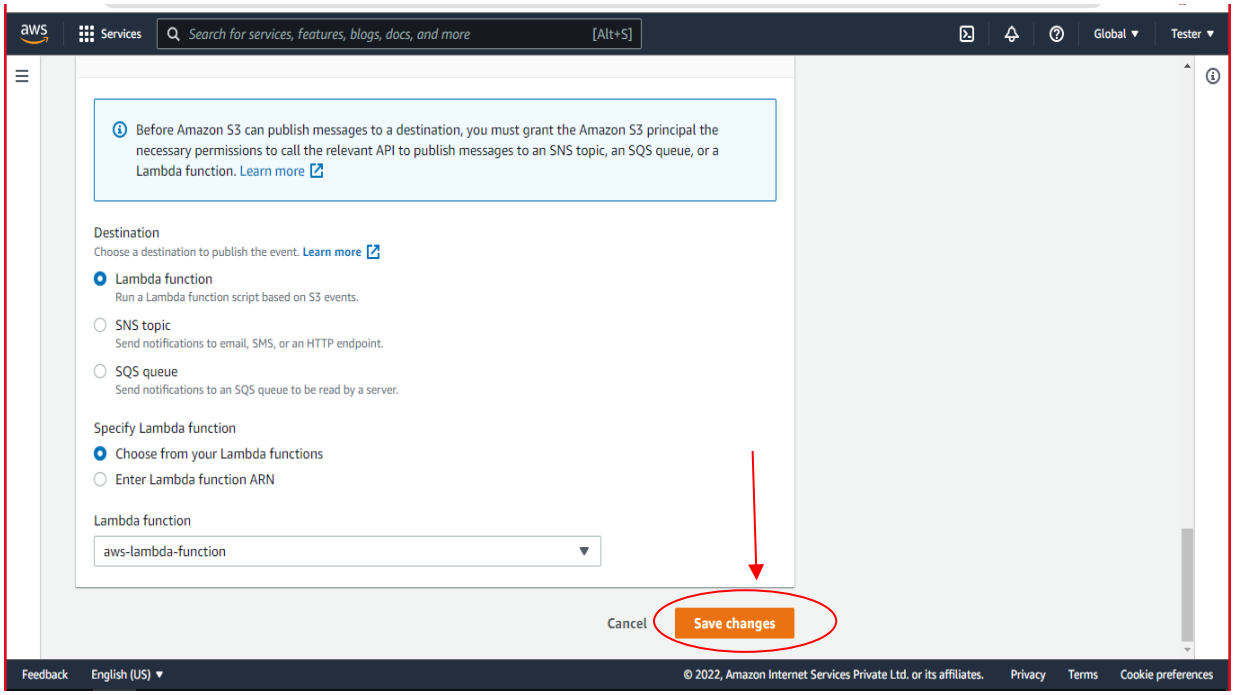

Fig 3.6.7

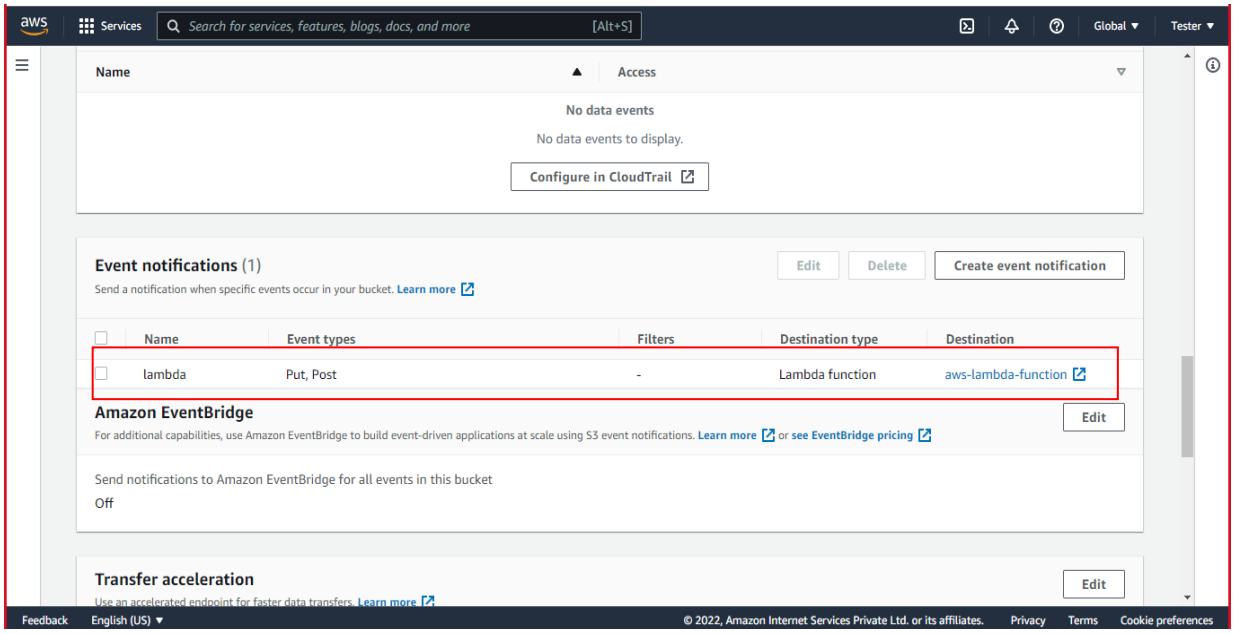

Fig 3.6.8

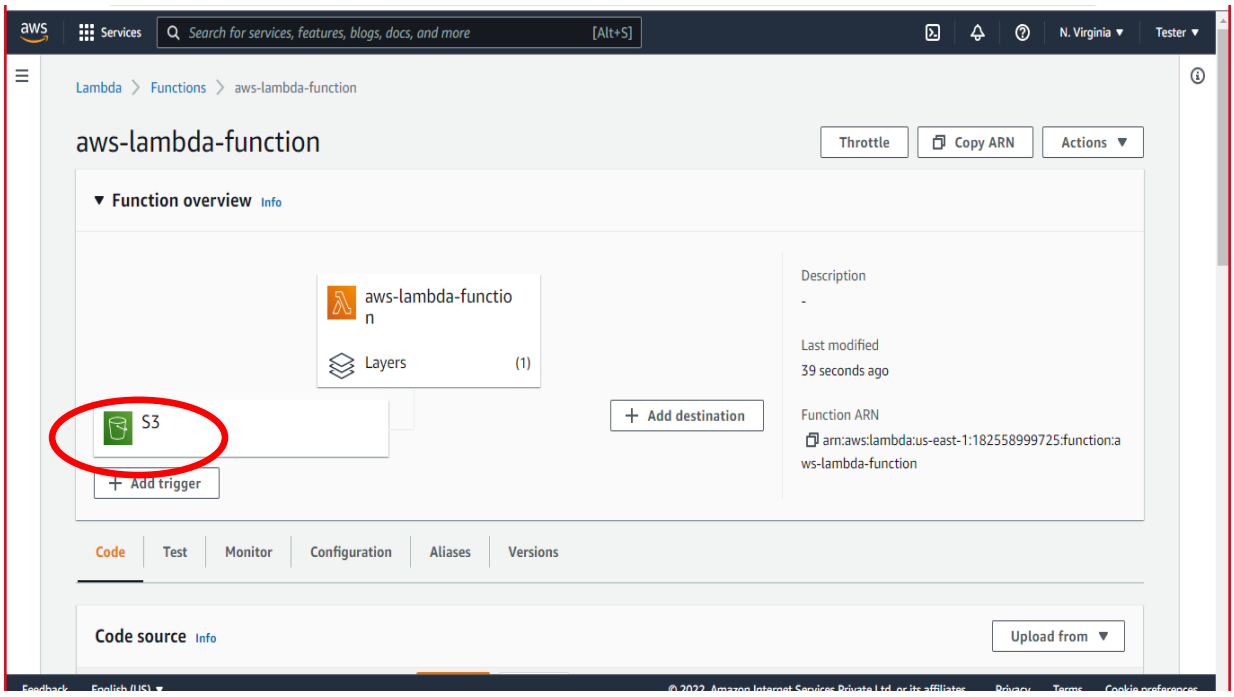

Fig 3.6.9

STEP 7 : Create A Redshift Cluster.

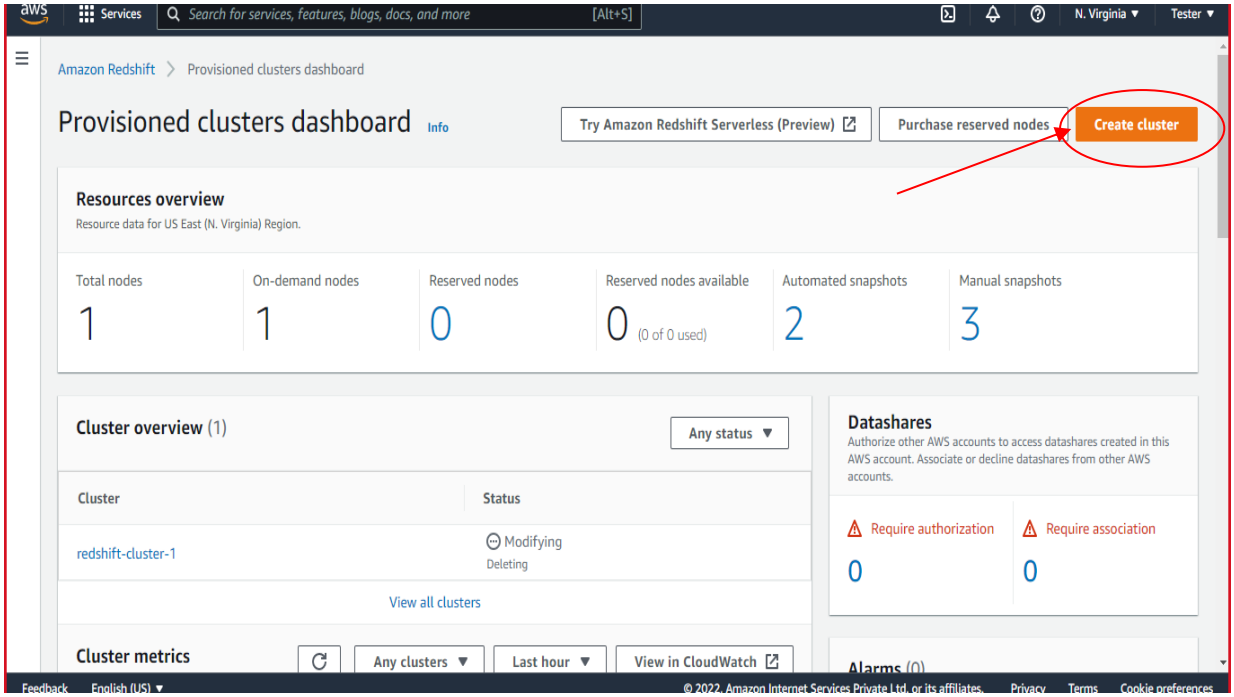

Fig 3.7.1

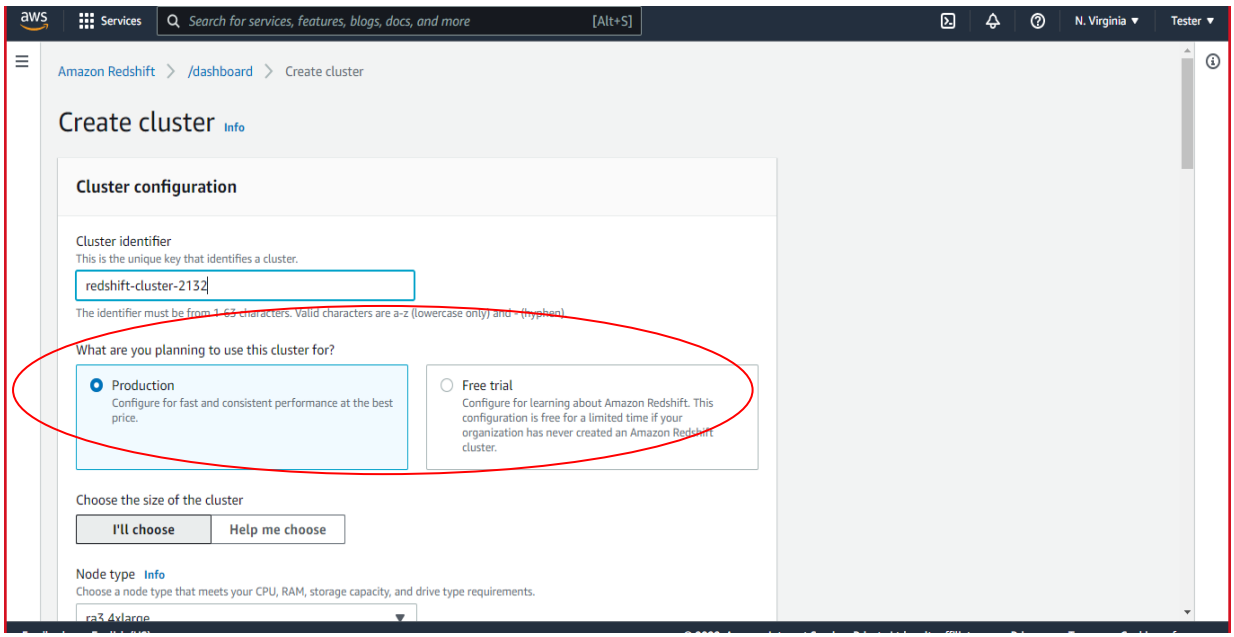

Fig 3.7.2

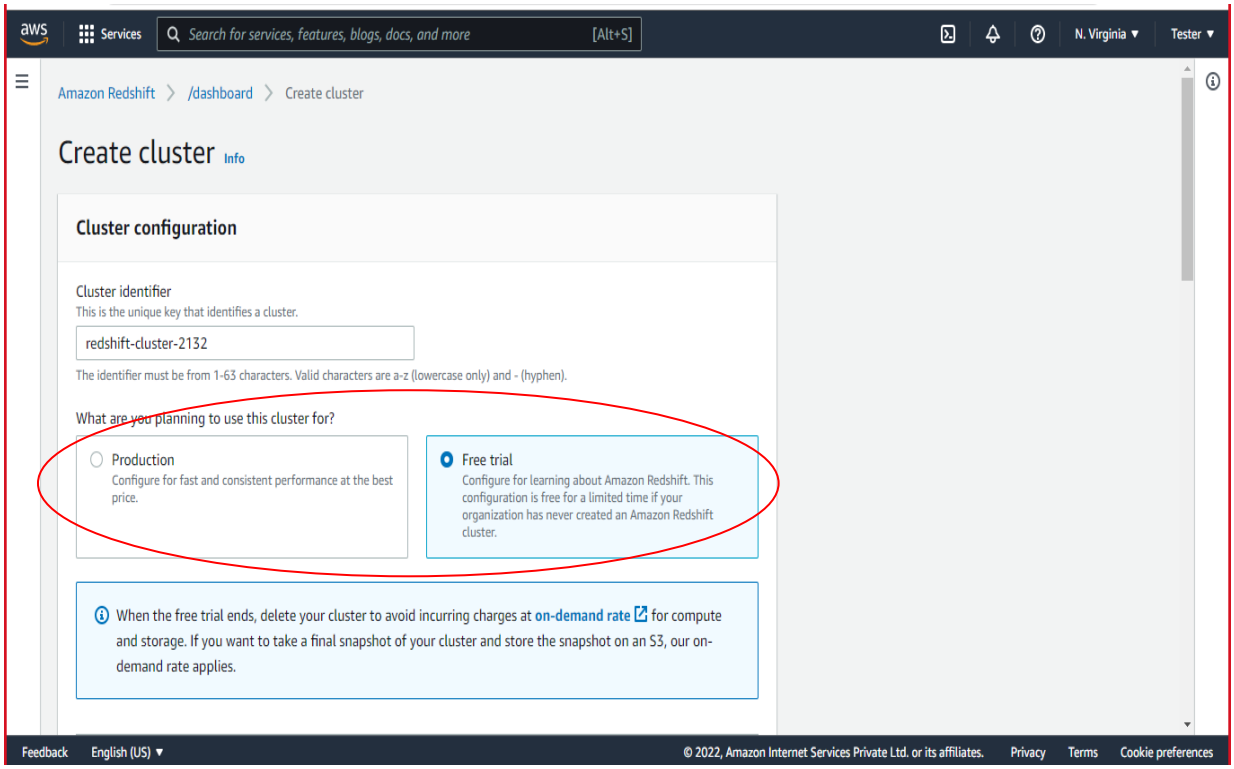

Fig 3.7.3

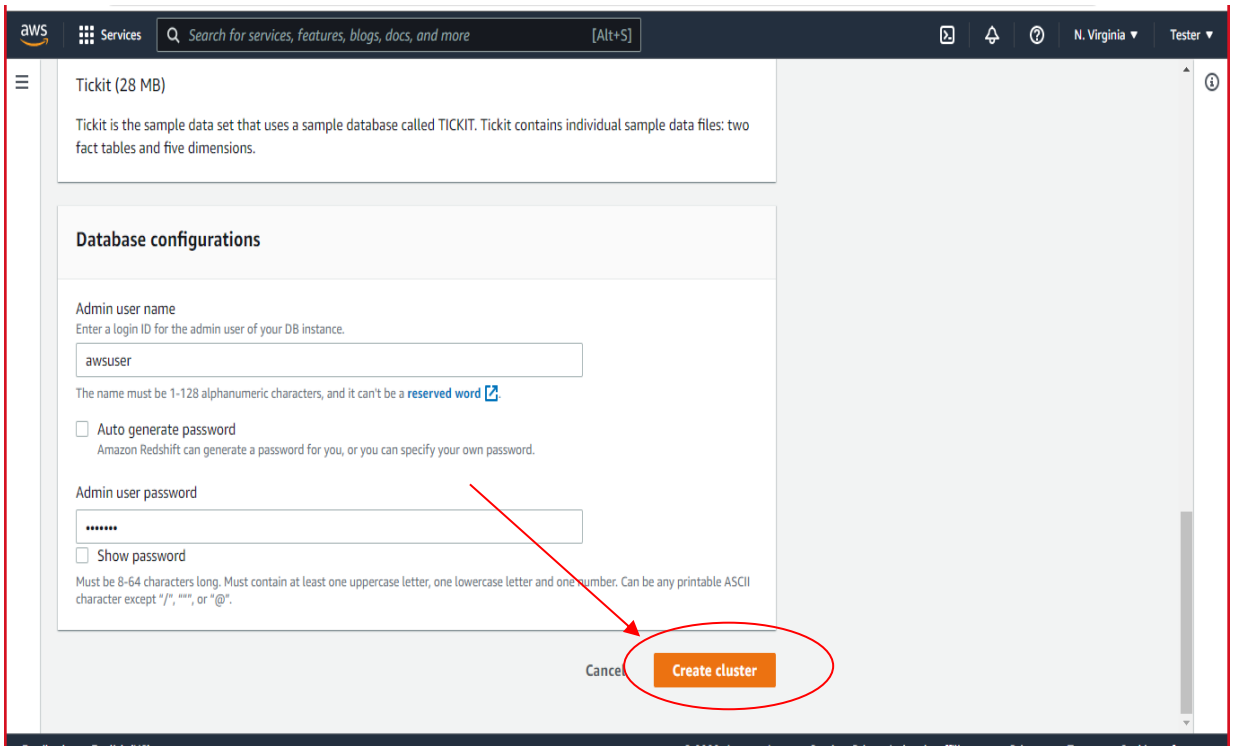

Fig 3.7.4

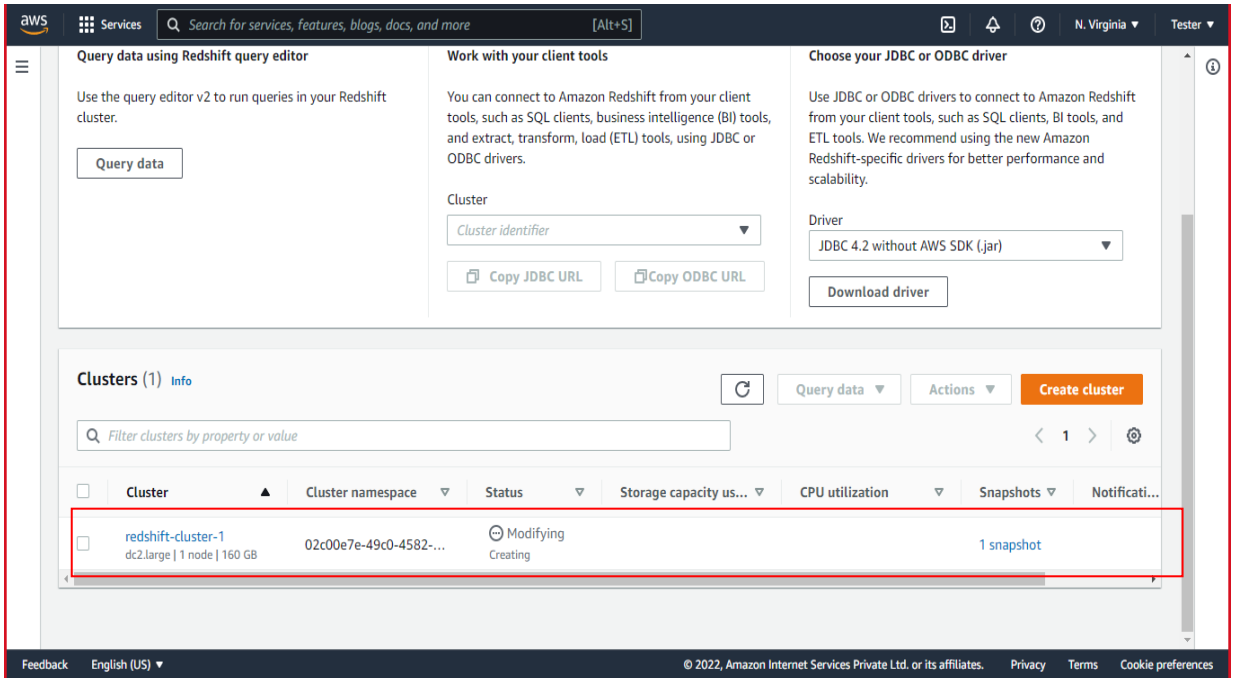

Fig 3.7.5

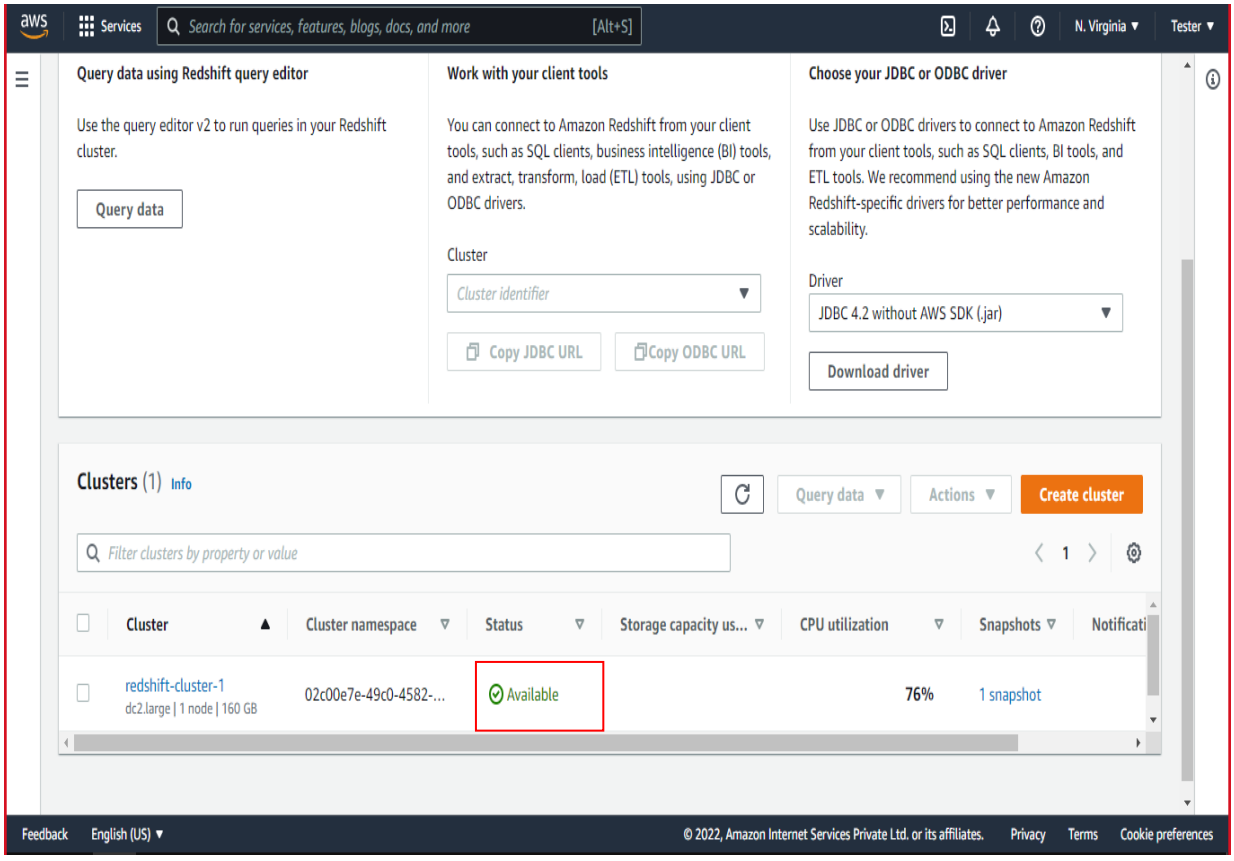

Fig 3.7.6

STEP 8: Edit Publicly Assessible.

Note: Without this step the redshift can't access from outside(publicly).

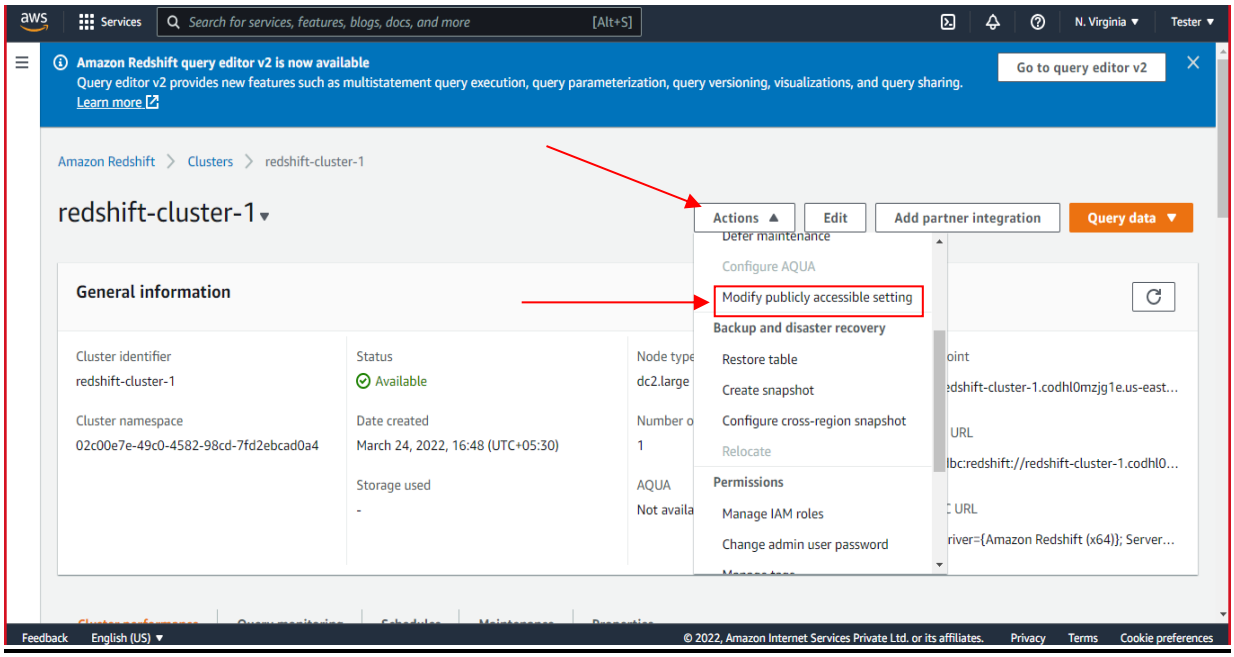

Fig 3.8.1

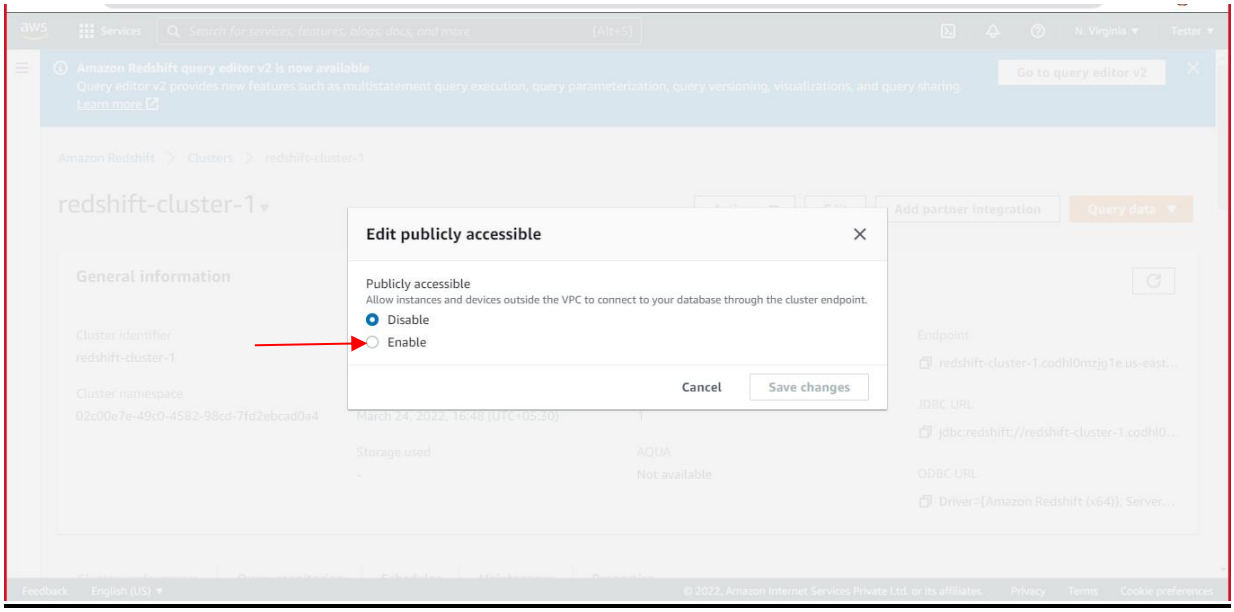

Fig 3.8.2

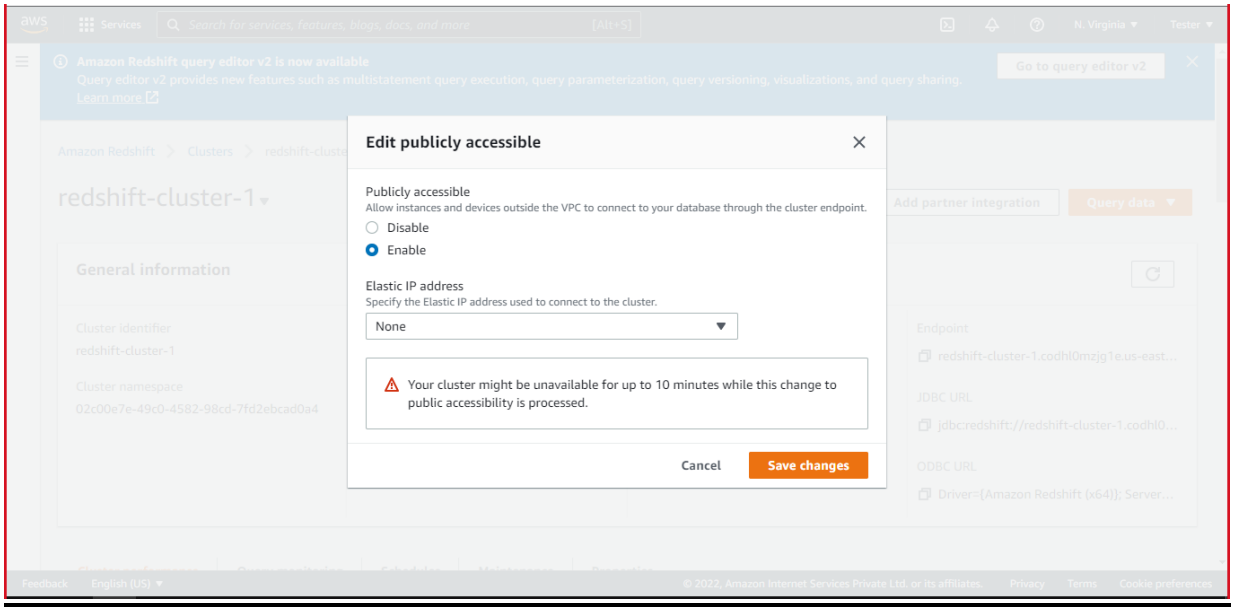

Fig 3.8.3

STEP 9: Set-Up VPC In Lambda.

NOTE - it's similar to Redshift VPC

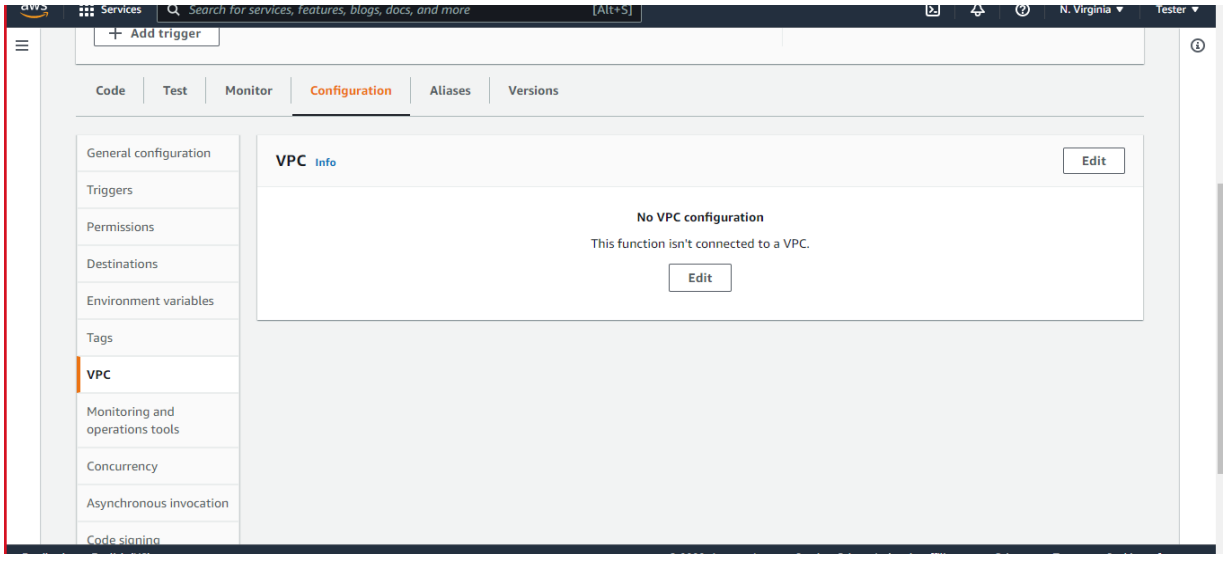

Fig 3. 9.1

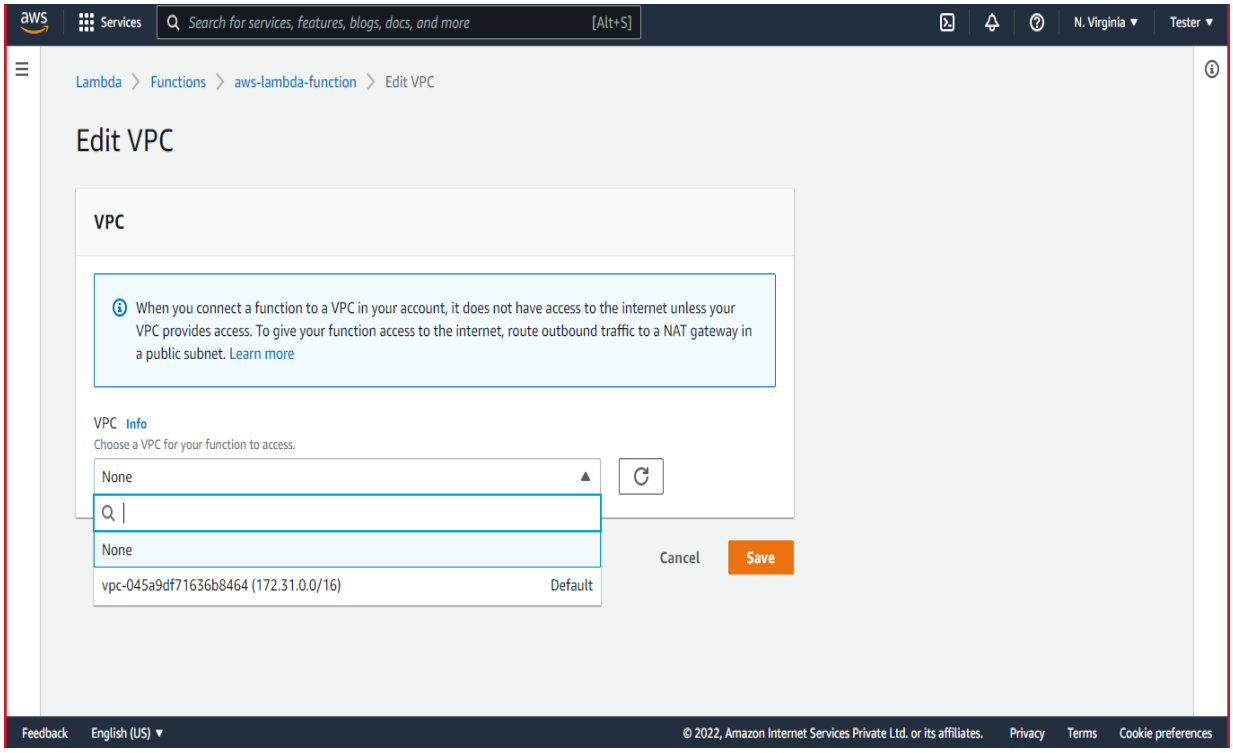

Fig 3.9.2

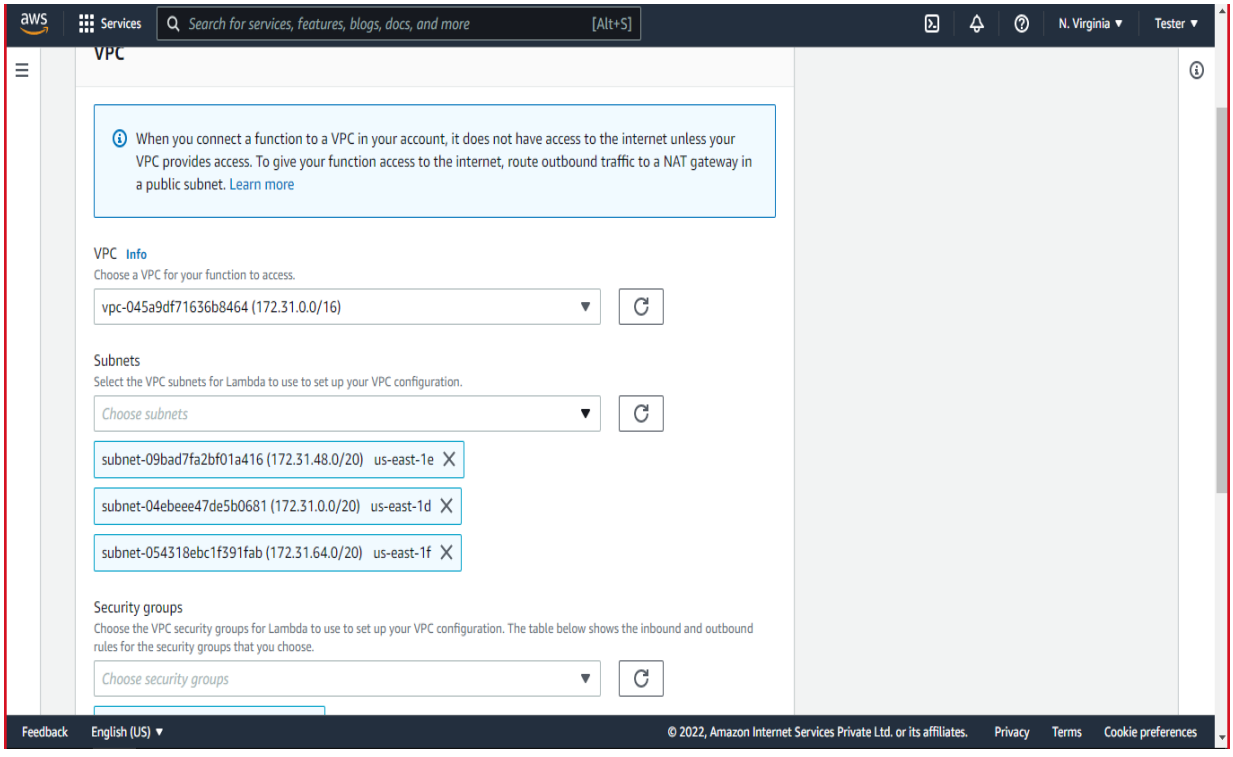

Fig 3.9.3

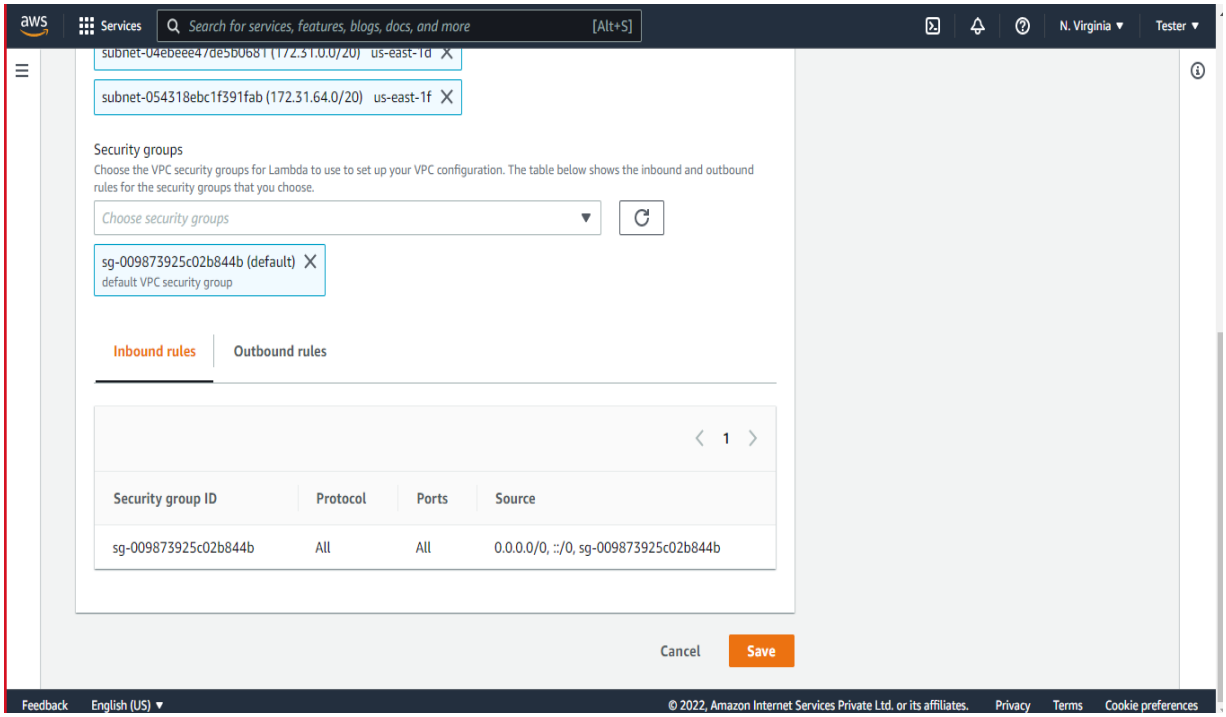

Fig 3.9.4

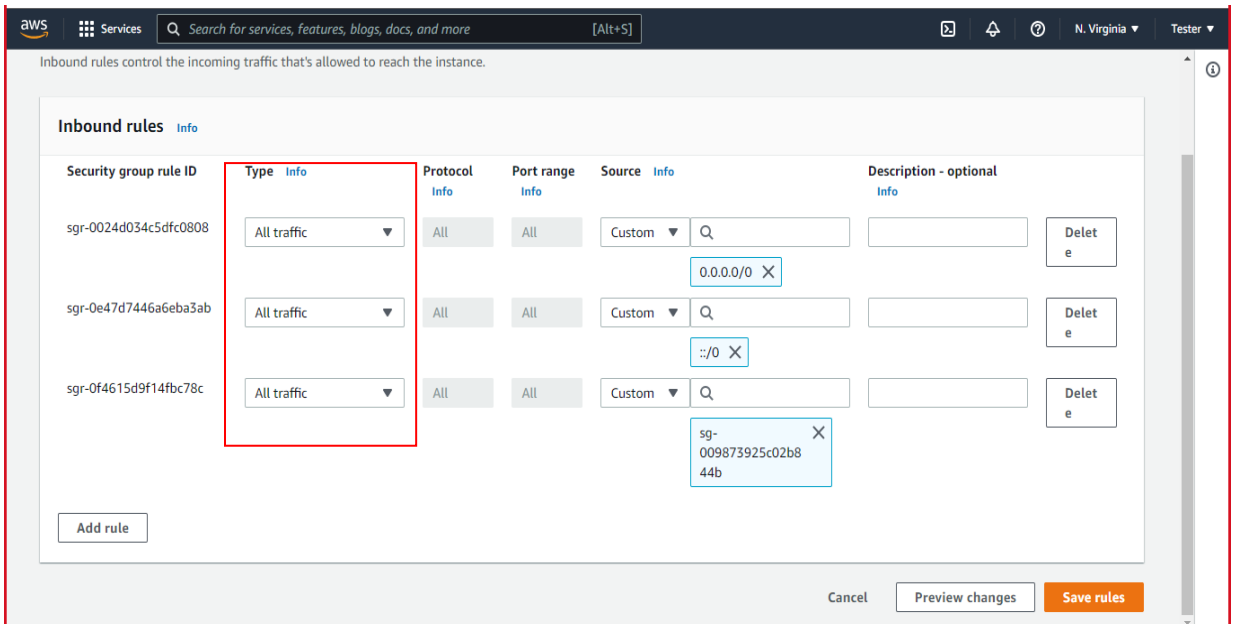

Fig 3.9.5

STEP 10: Programming For Automation Pipelines .

Get the code below link :

https://github.com/Amal-p/Redshift/blob/main/lambda_function.py

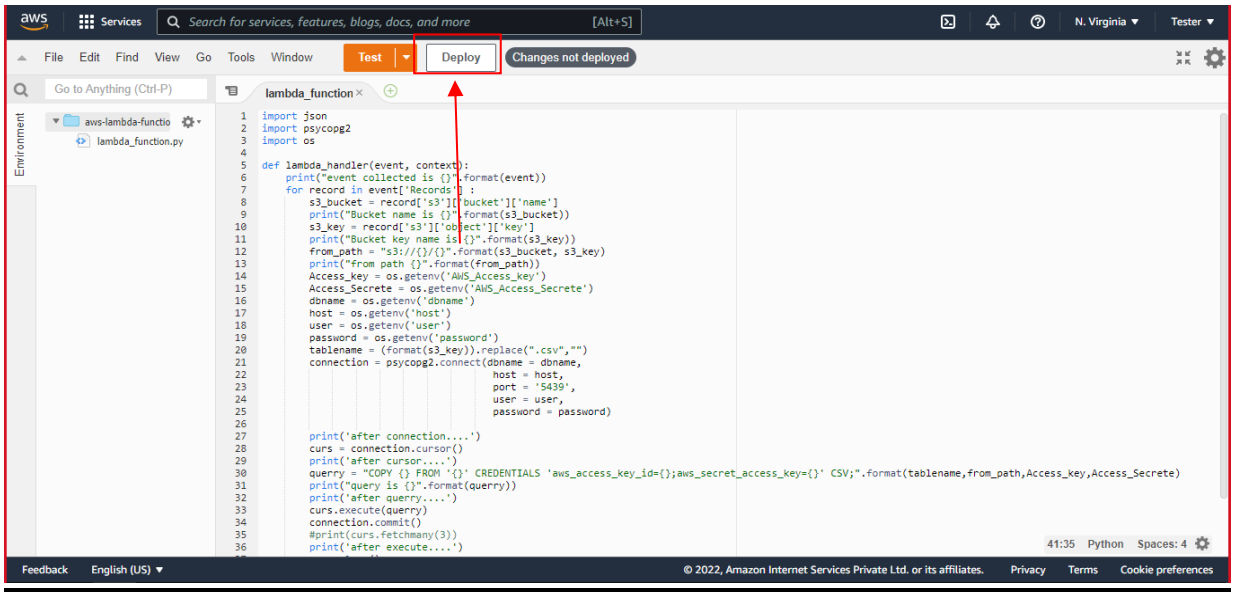

Fig 3.10.1

STEP 11: Setup ENV values.

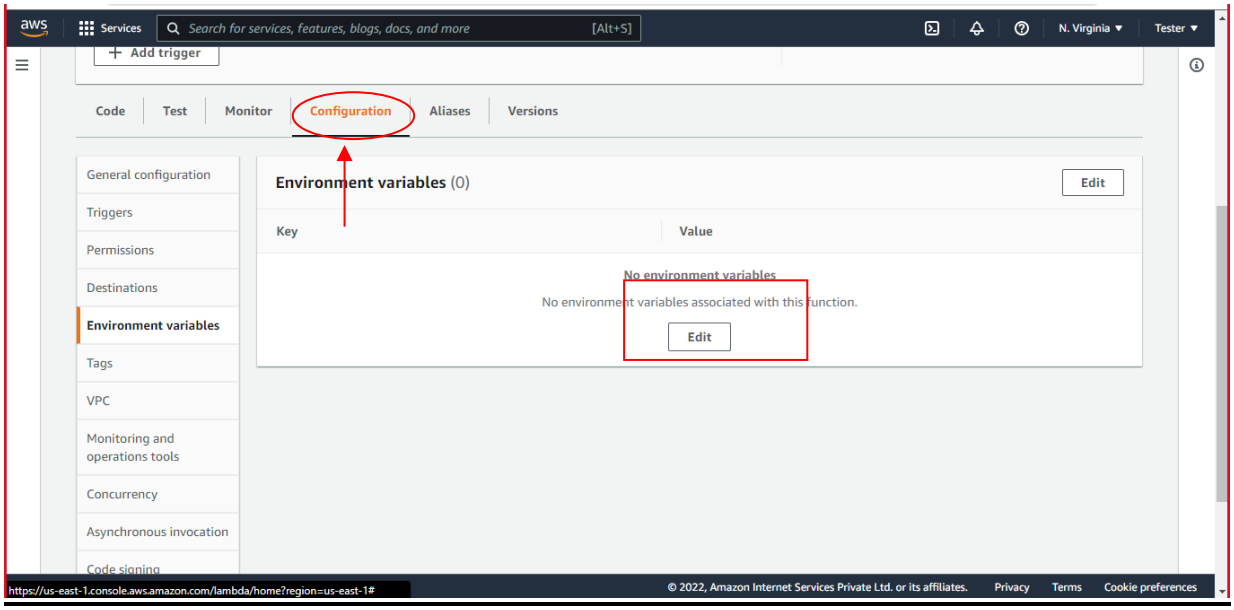

Fig 3.11.1

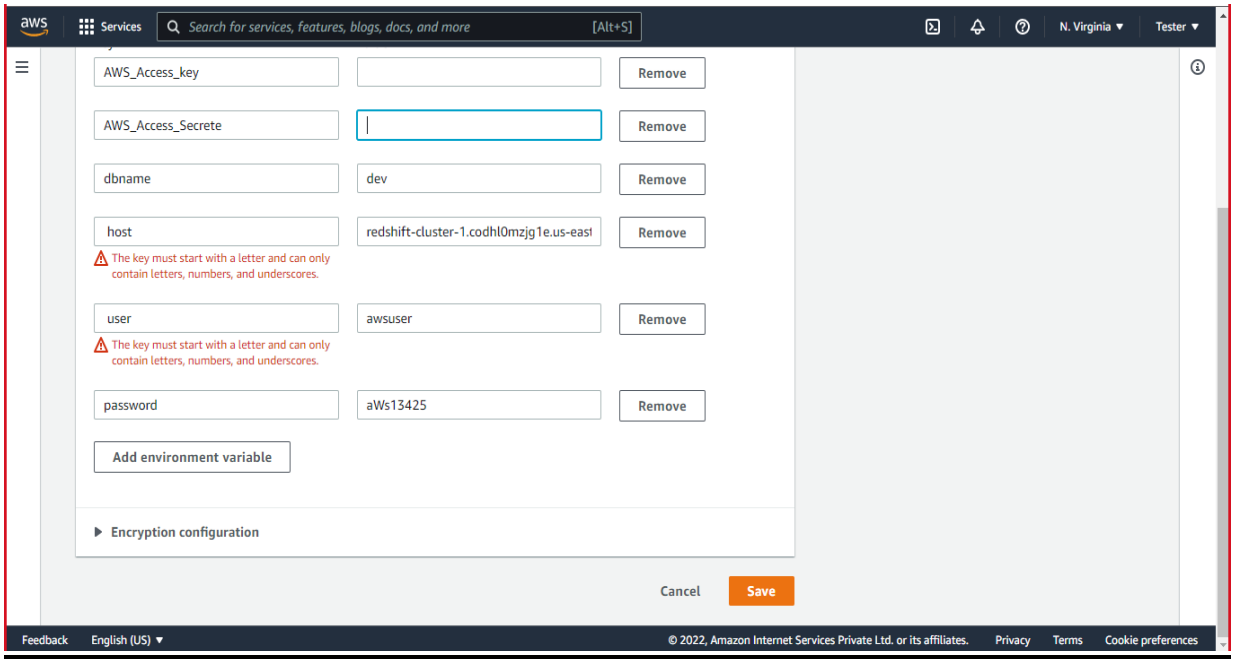

Fig 3.11.2

Note : host value will get in fig 11.3

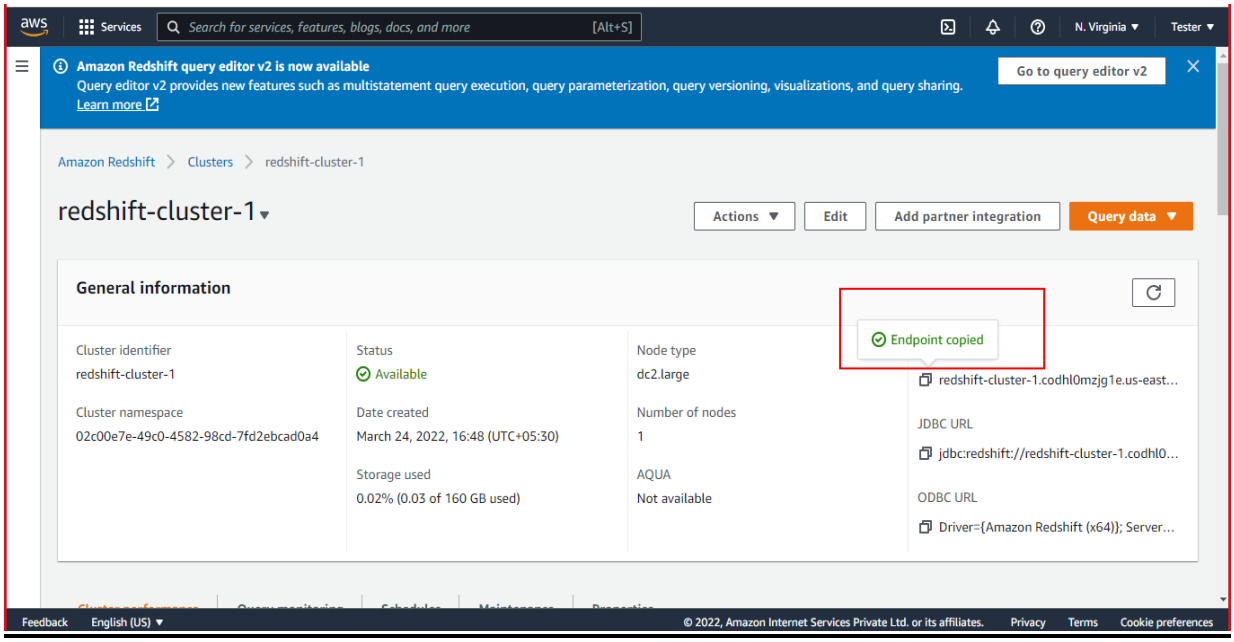

Fig 3.11.3

CHAPTER 4

OUTPUT

Add the files and folders you want to upload to s3.(if you have decided to upload similar file(same name) then you must have delete the existing file on s3 first).

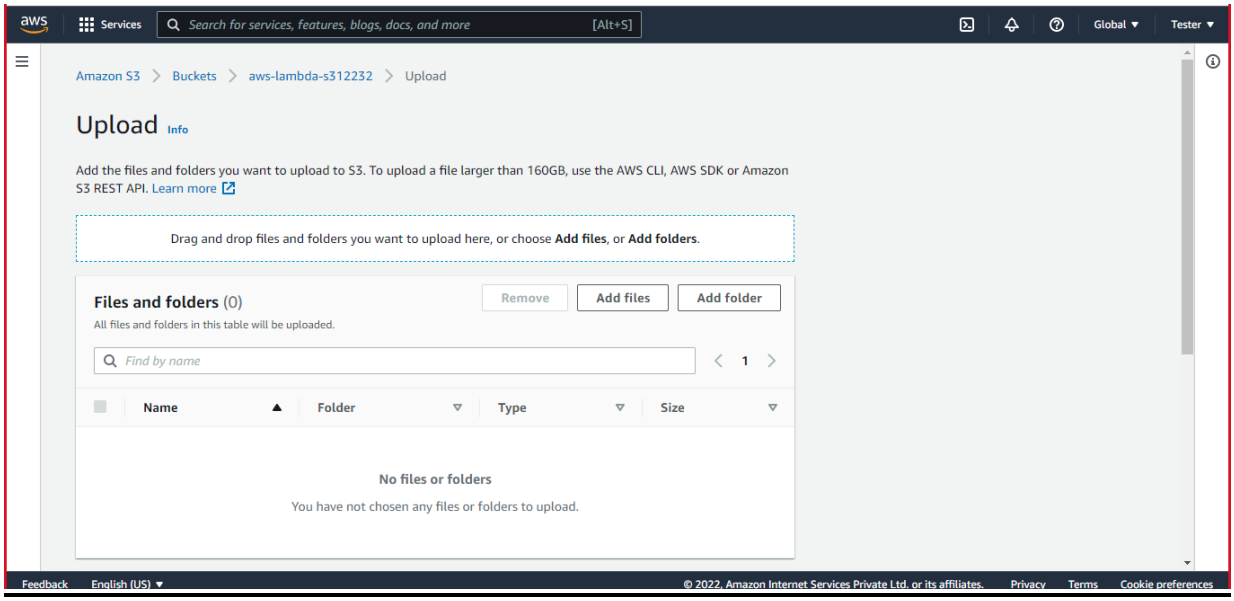

Fig.4.1

Choose any csv file from local machine(According to program file name must be similar to table name).

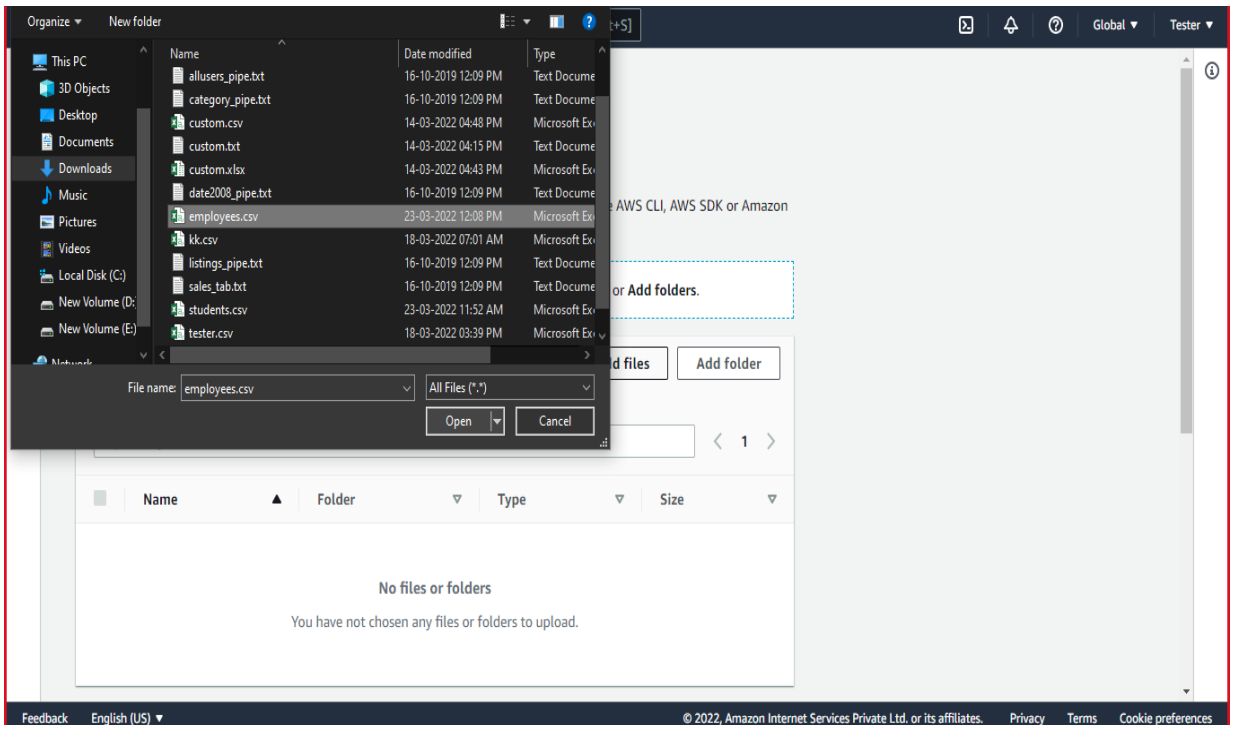

Fig.4.2

Upload the selected file to s3 with default settings.

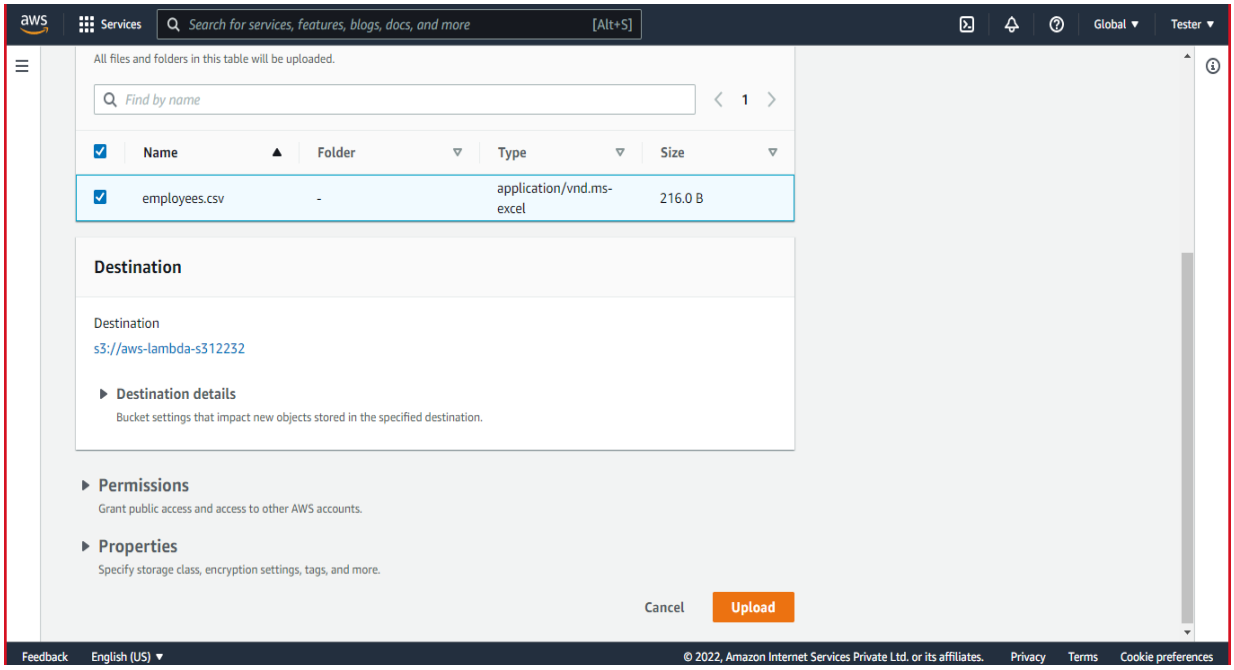

Fig.4.3

Objects are the fundamental entities stored in amazon s3.

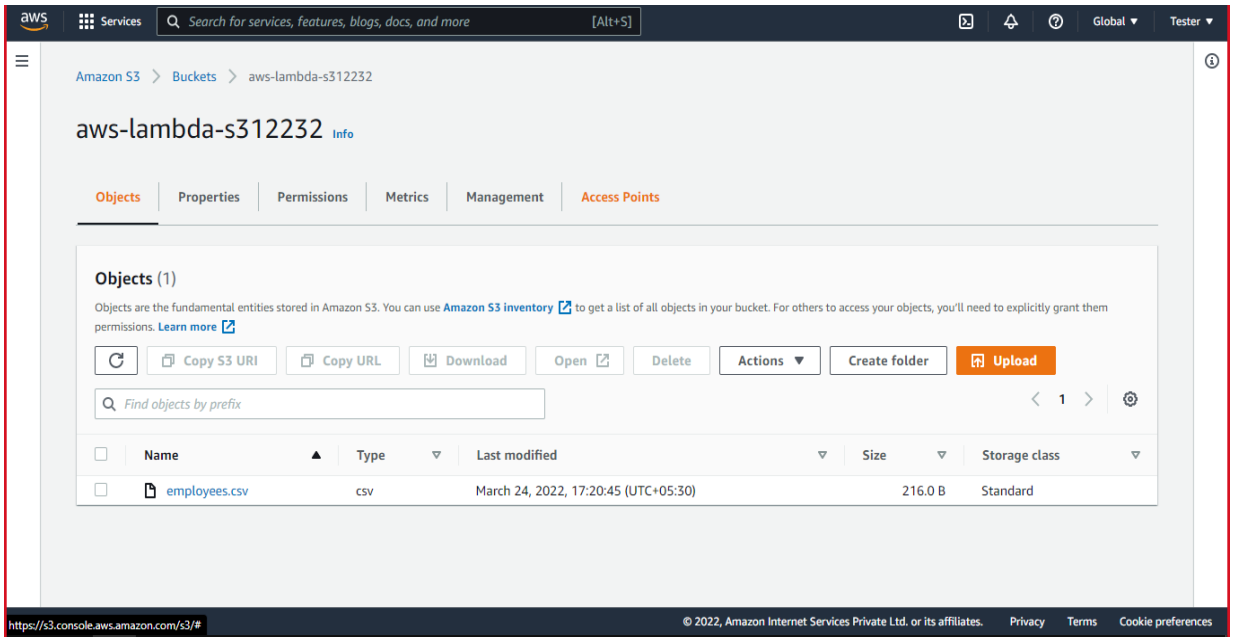

Fig 4.4

When you finish the upload you can see the work tree of lambda function inside cloudwatch(log).

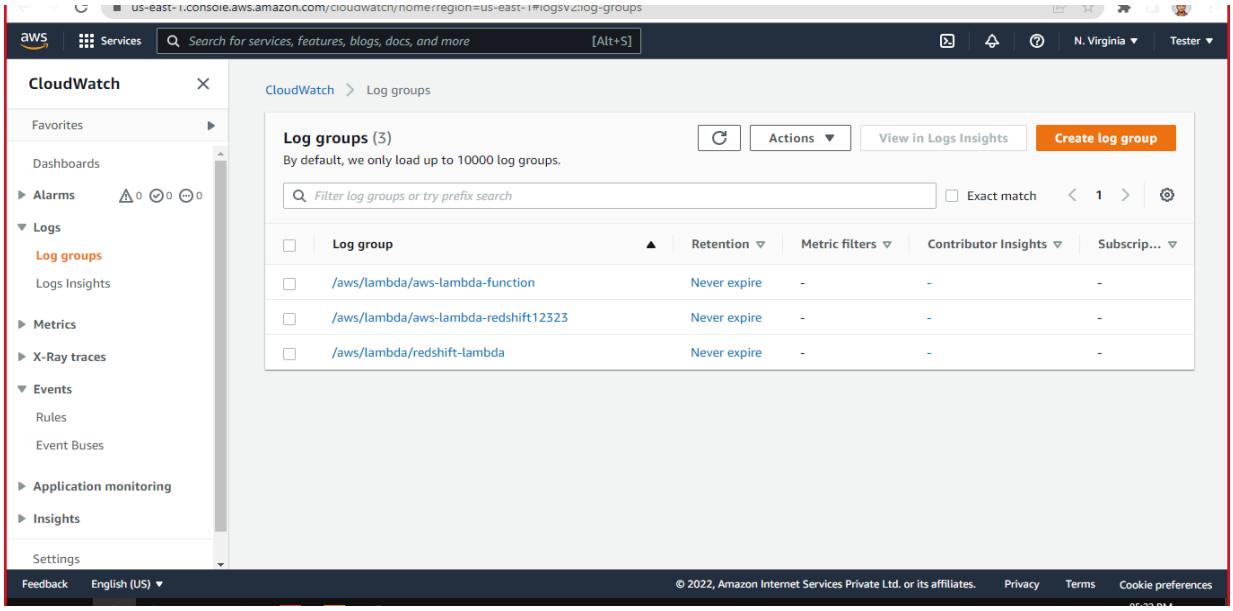

Fig.4.5

If there is no error then the procedure start with triggering of lambda function by s3 bucket. Then the lambda code takes the file as an event and loop through how many files you uploaded and copy each file content to redshift data ware house.

After finishing the procedure you can check data is present or not by putting this sql statement ("select * from <table name>")

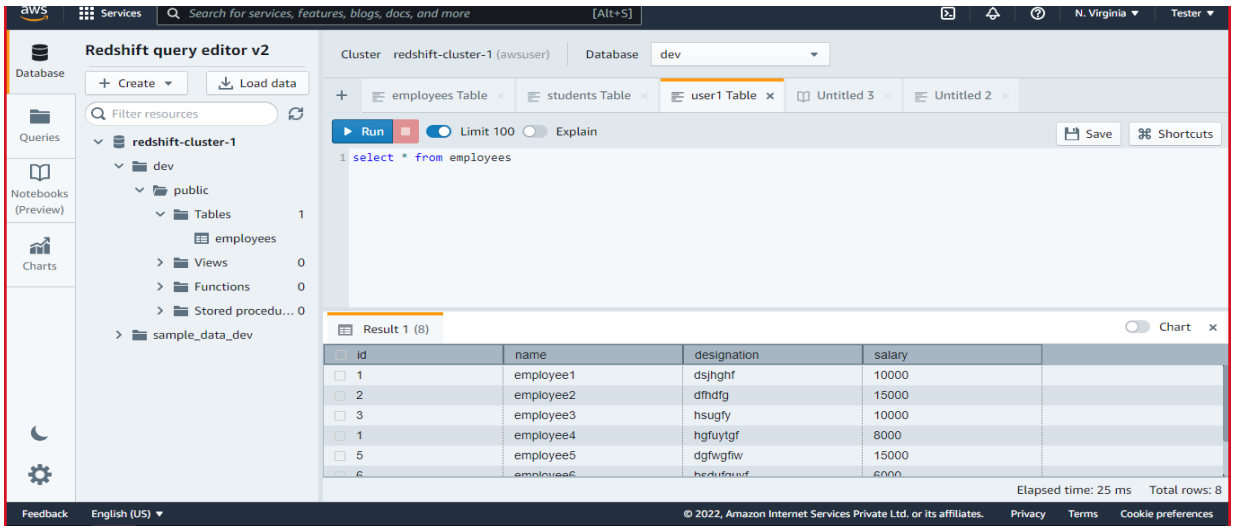

Fig.4.6