Managing tightly coupled architecture using Amazon SQS

By Anand Vishwakarma

## Course-end Project 1

**Description**:

Use AWS SQS to manage a tightly coupled architecture.

**Tools required:**

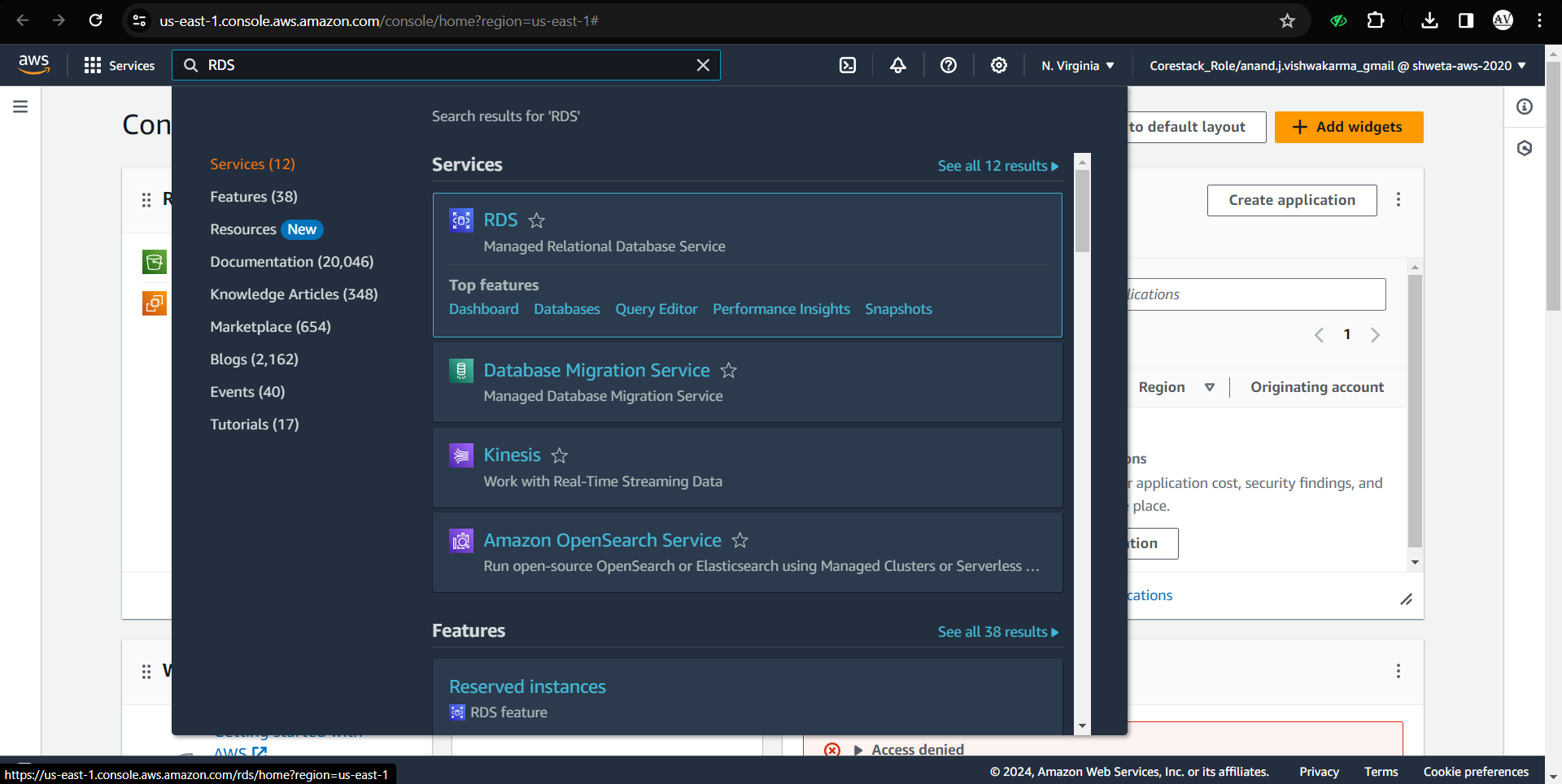
1. AWS SQS
2. AWS EC2
3. AWS IAM
4. AWS RDS

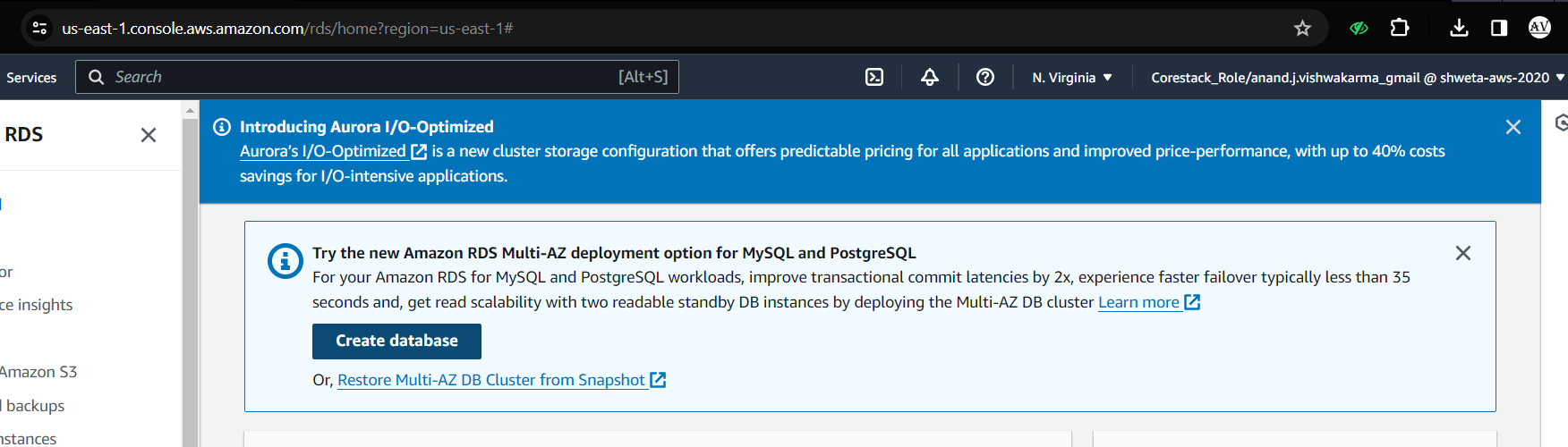
**Solution Steps:**

1. Create and configure AWS RDS instance.
2. Create and configure AWS SQS Queue.
3. Create and configure IAM Role.
4. Create and configure 2 EC2 instance.

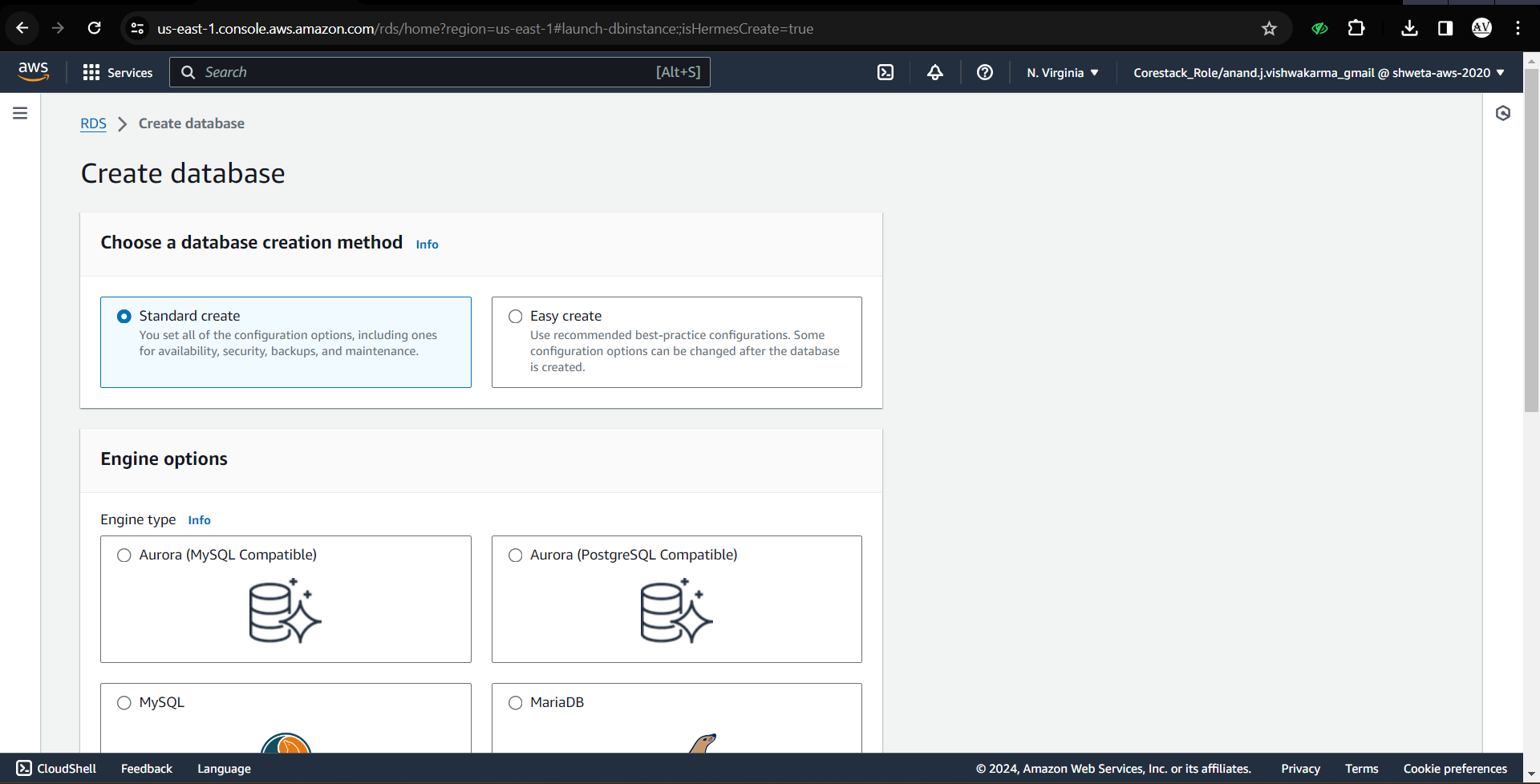
**Step 1:** Creating and configuring AWS RDS instance

Create a new Amazon RDS

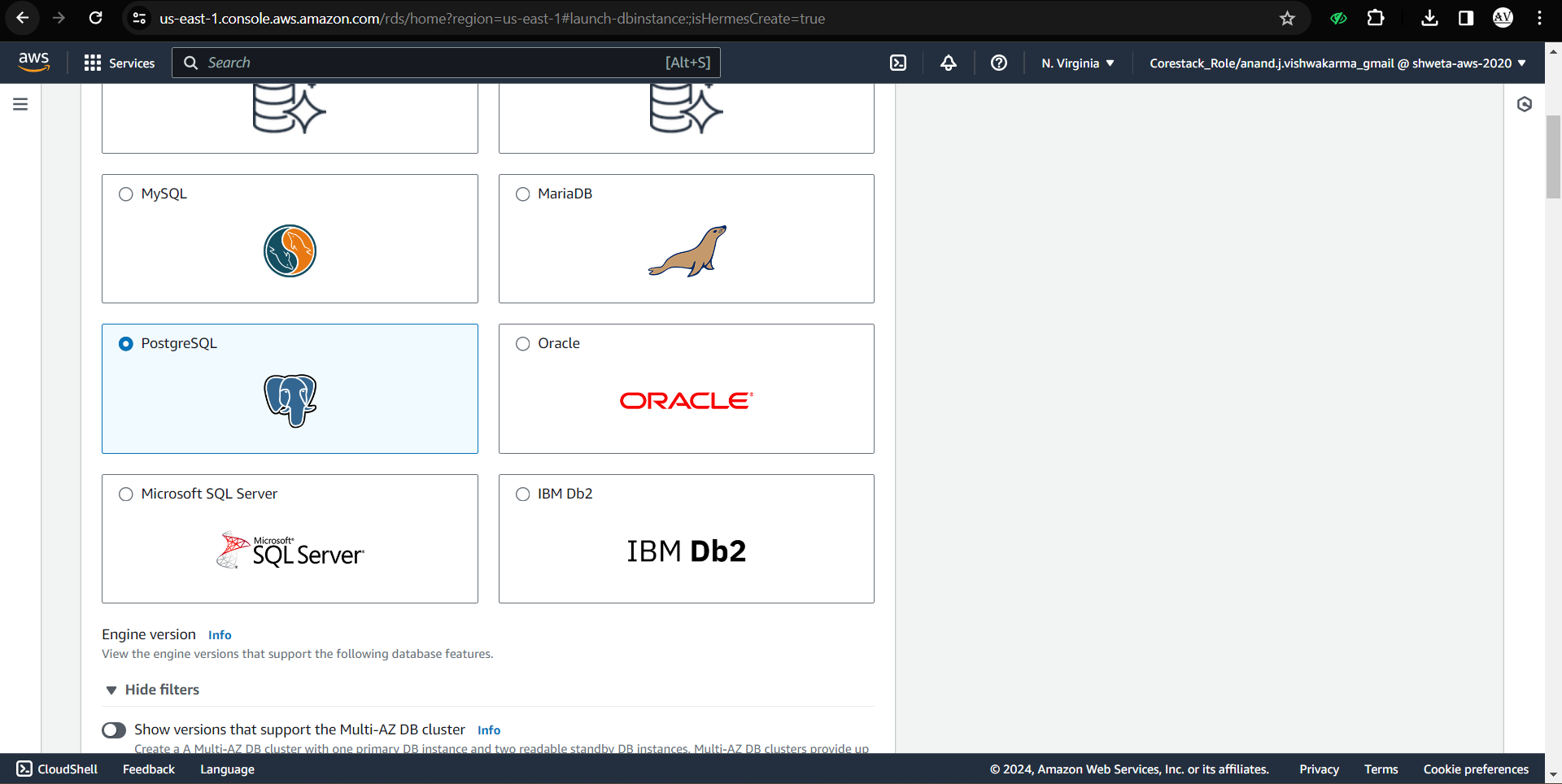




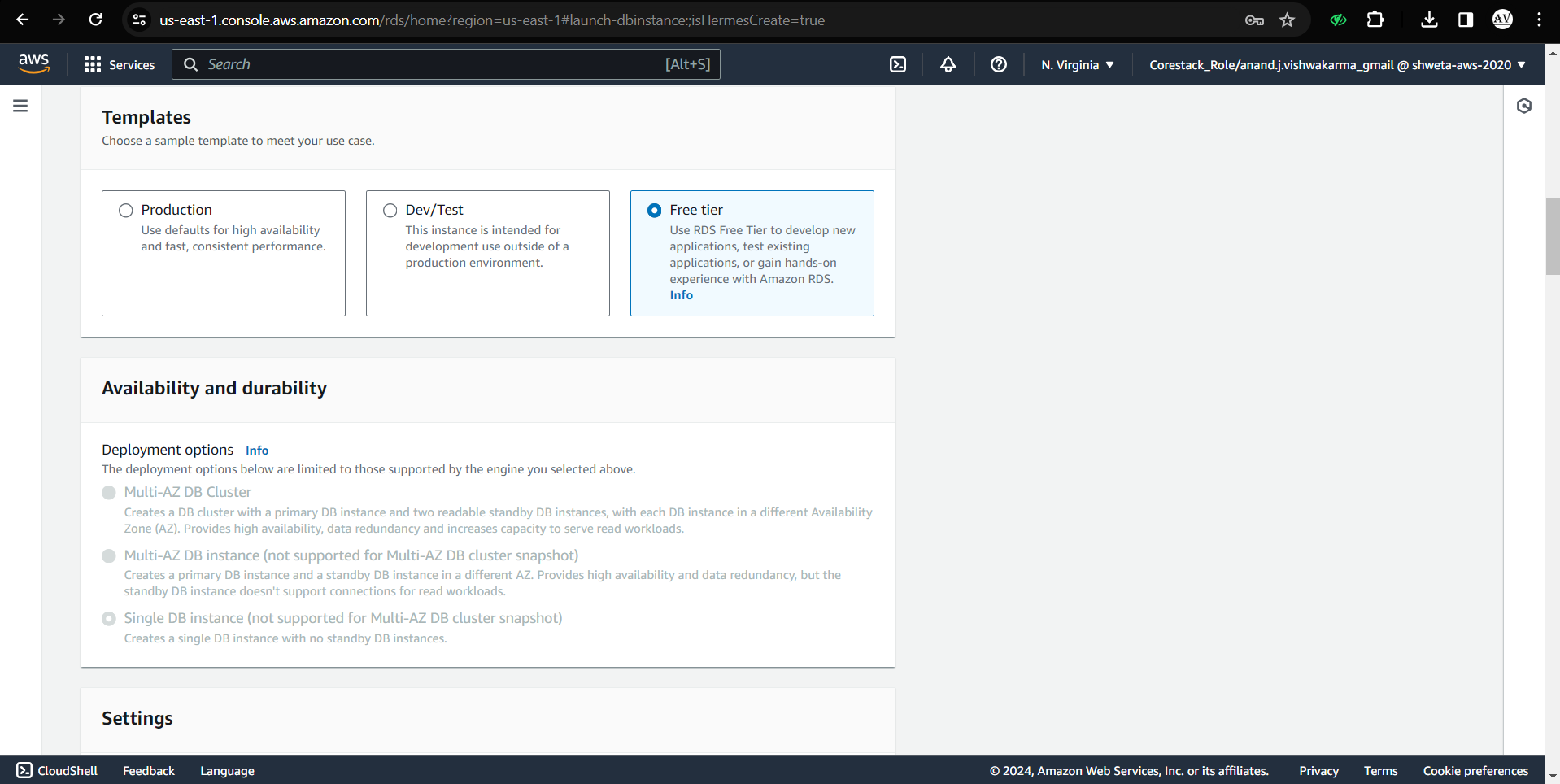
Select Standard create plan.



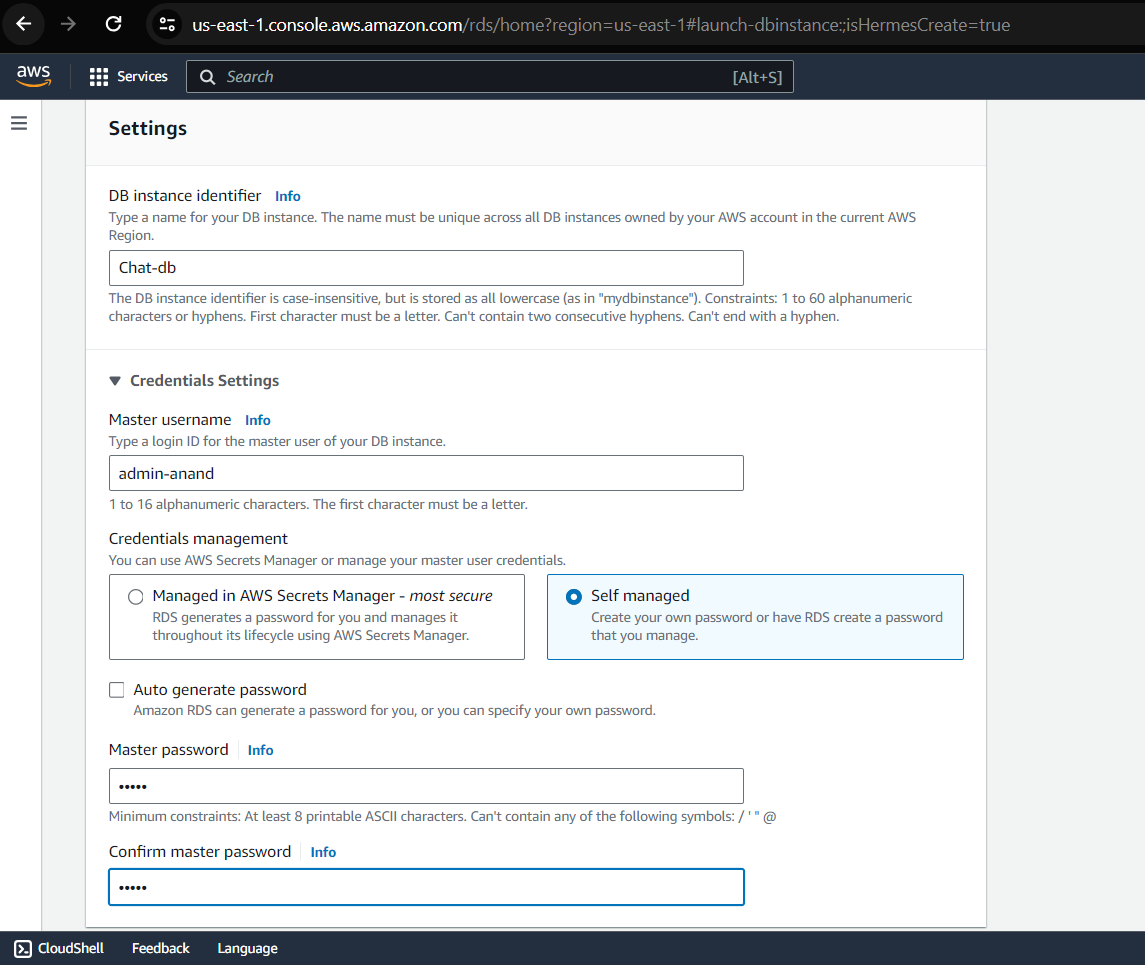
Select PostgreSQL database for this use-case.



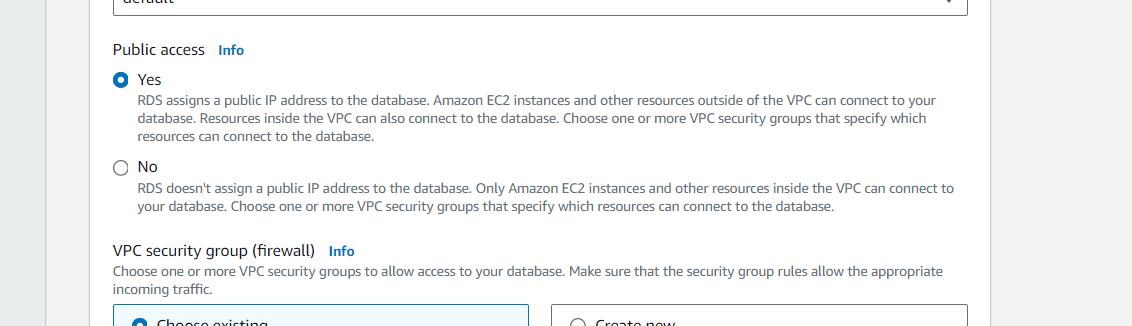
Select free tier.

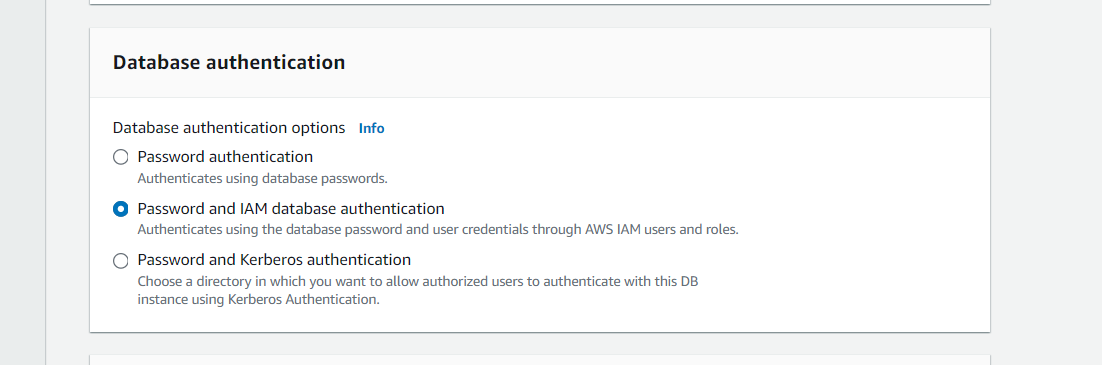


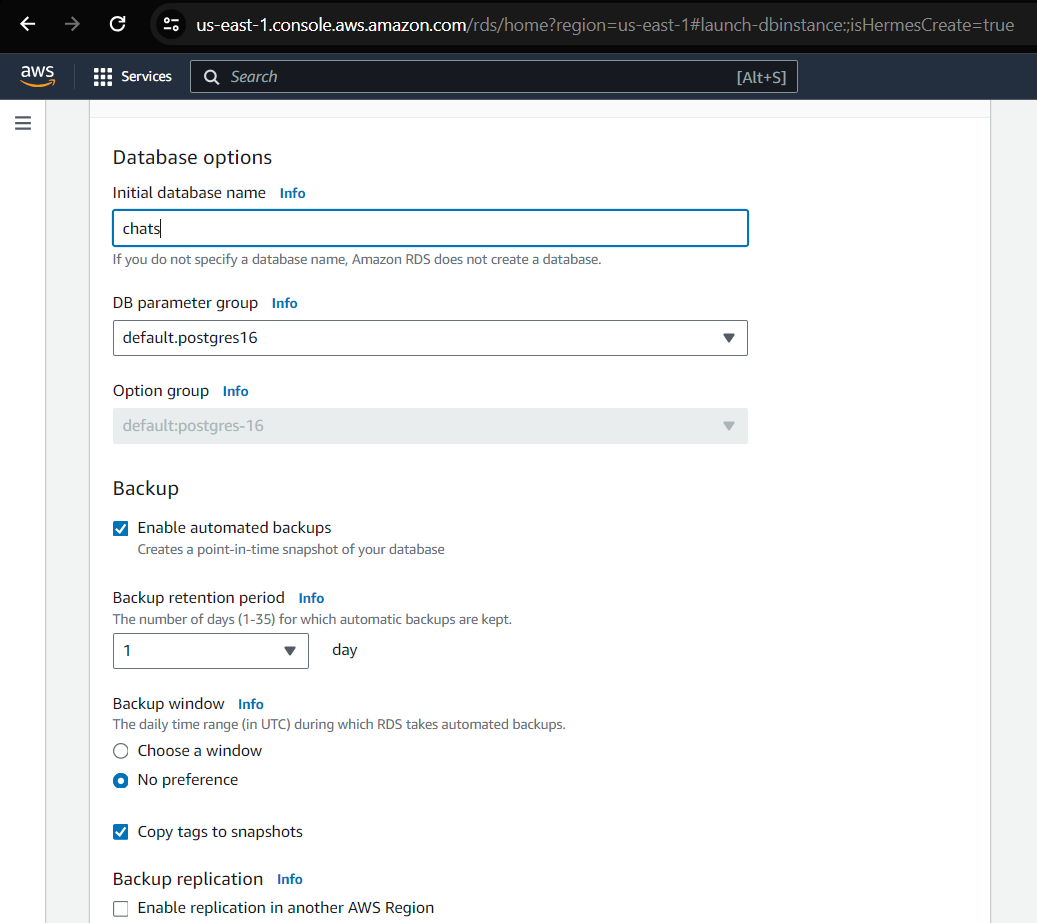
Insert database name, username and password to access database.



Select Public access and database authentication method as shown below.

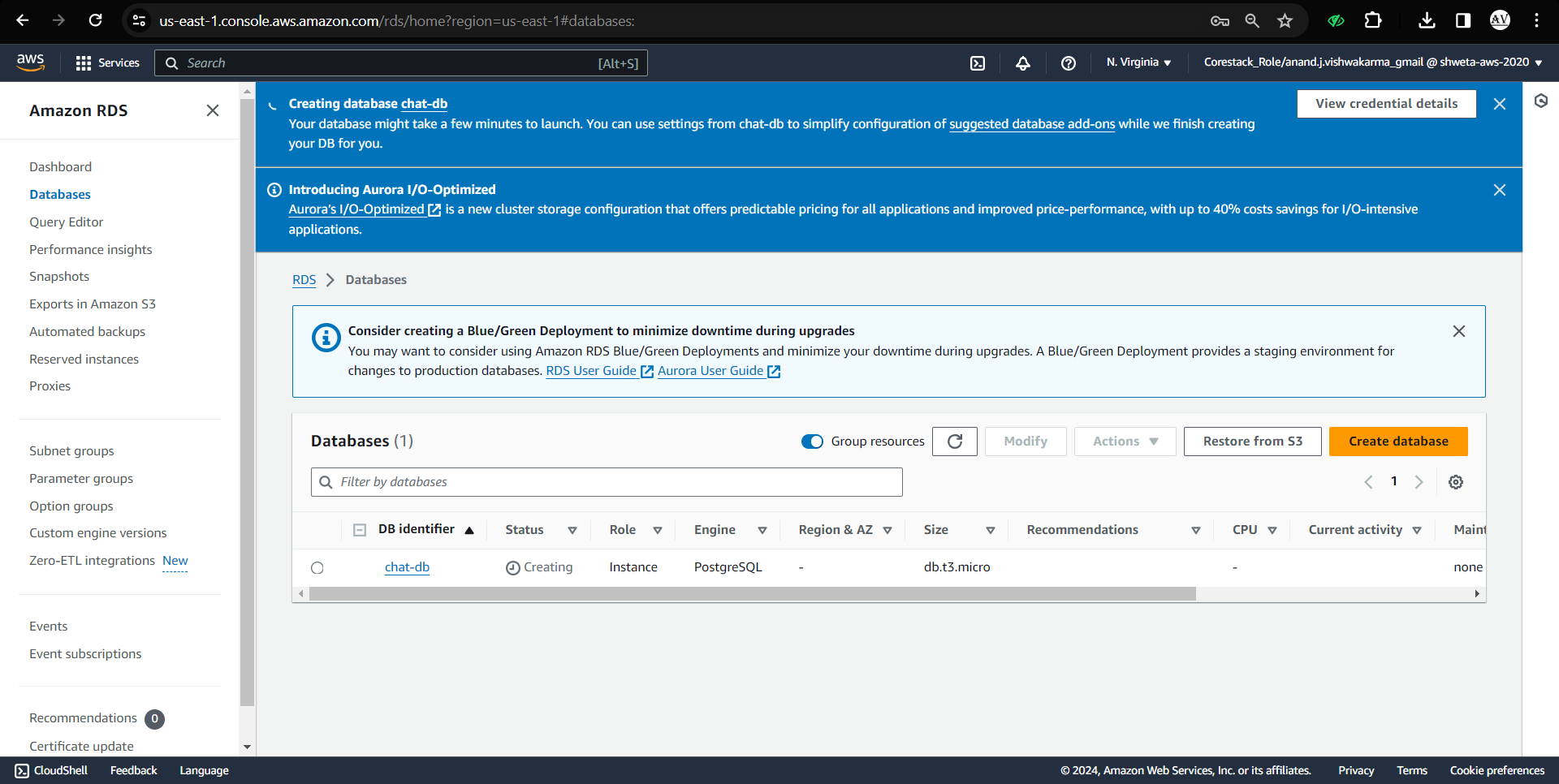




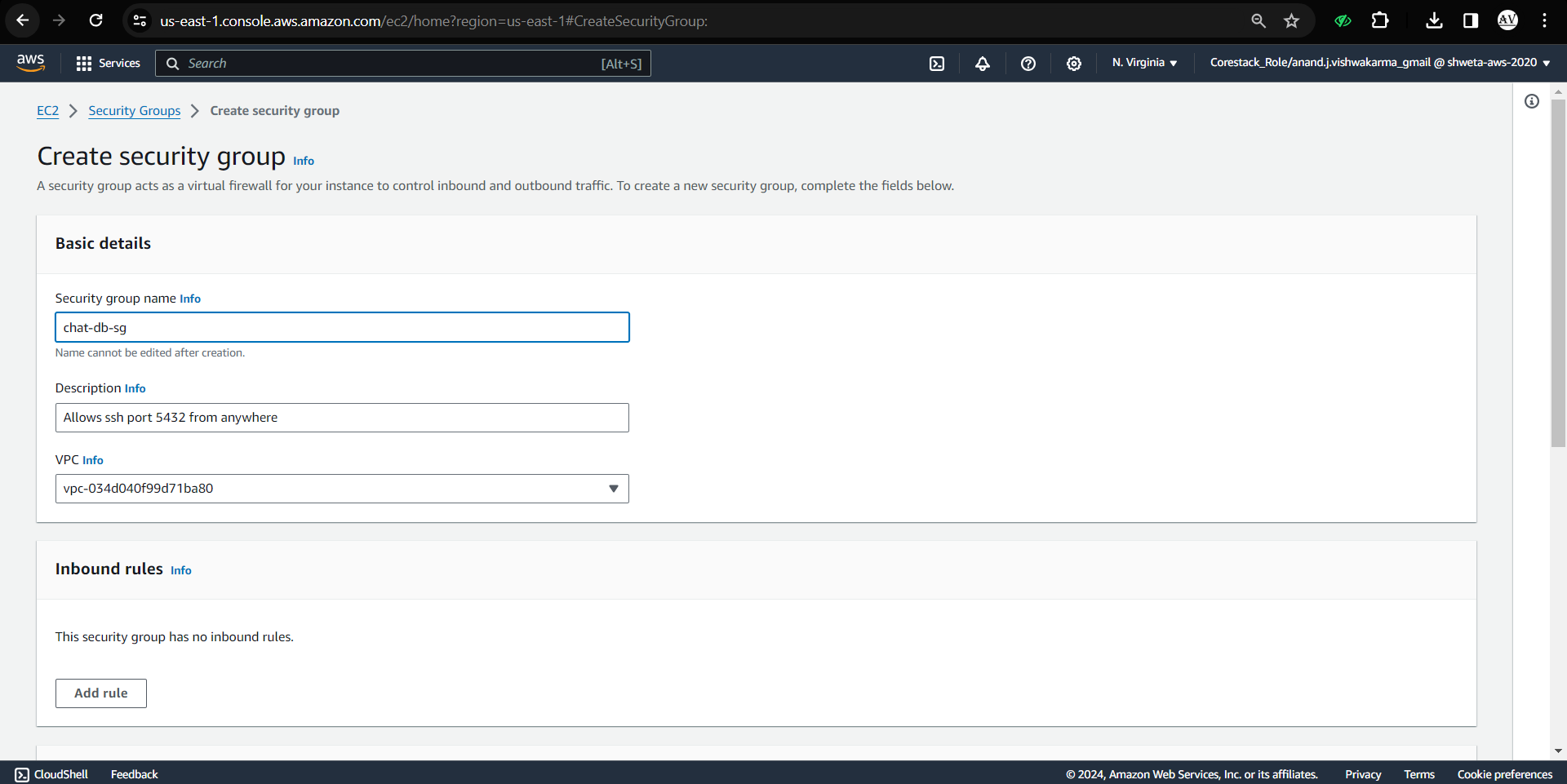


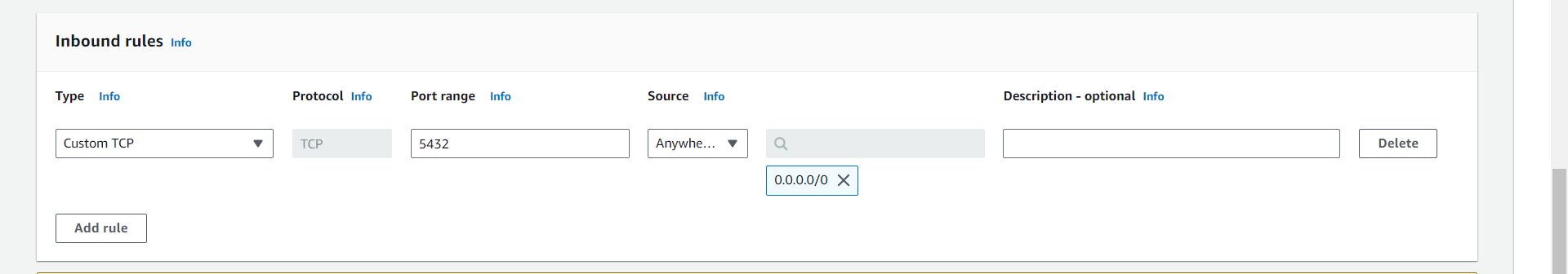
Once the above details are inserted and selected, click on create database button.

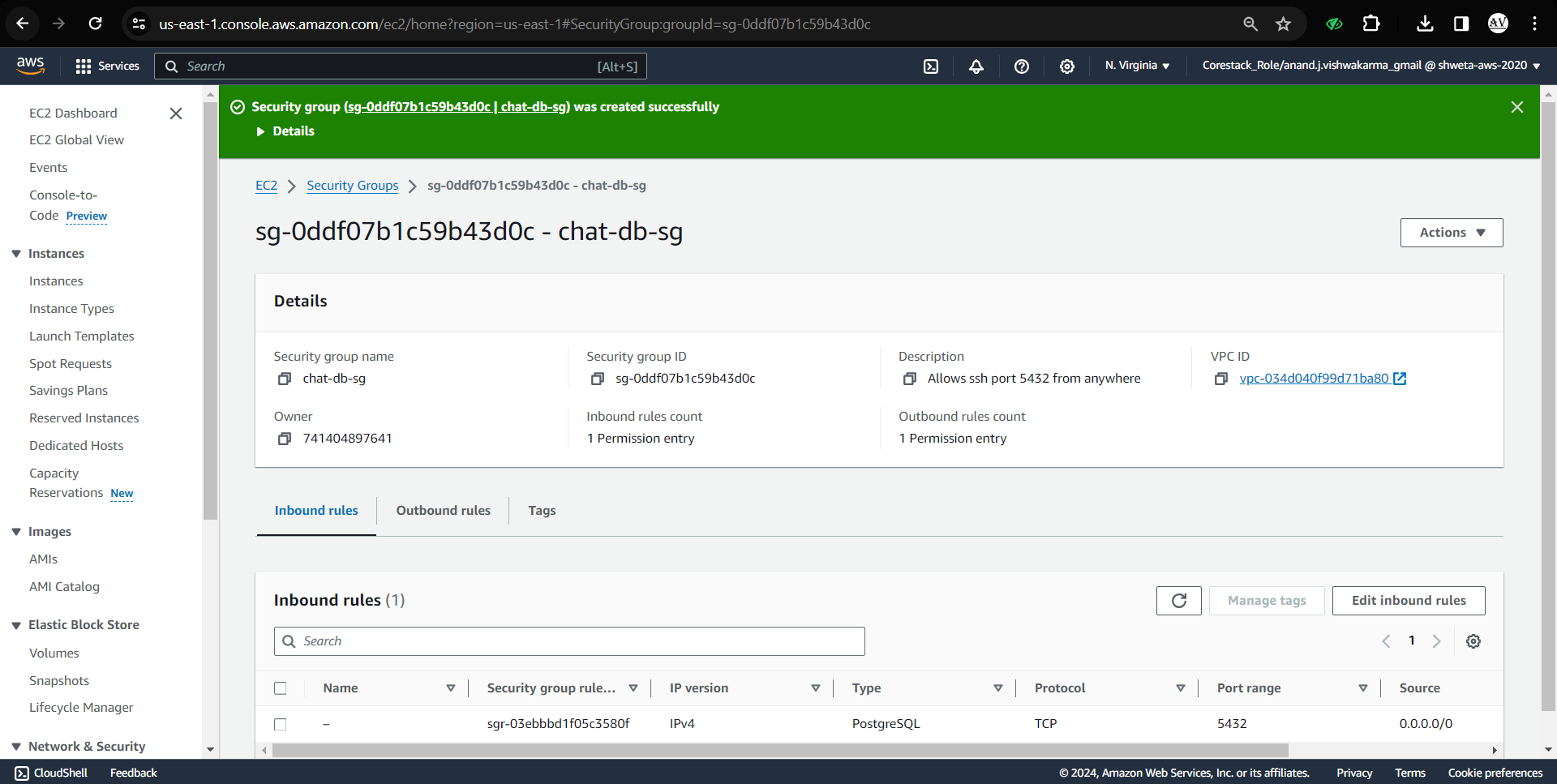
It will take few seconds to get the instance up.



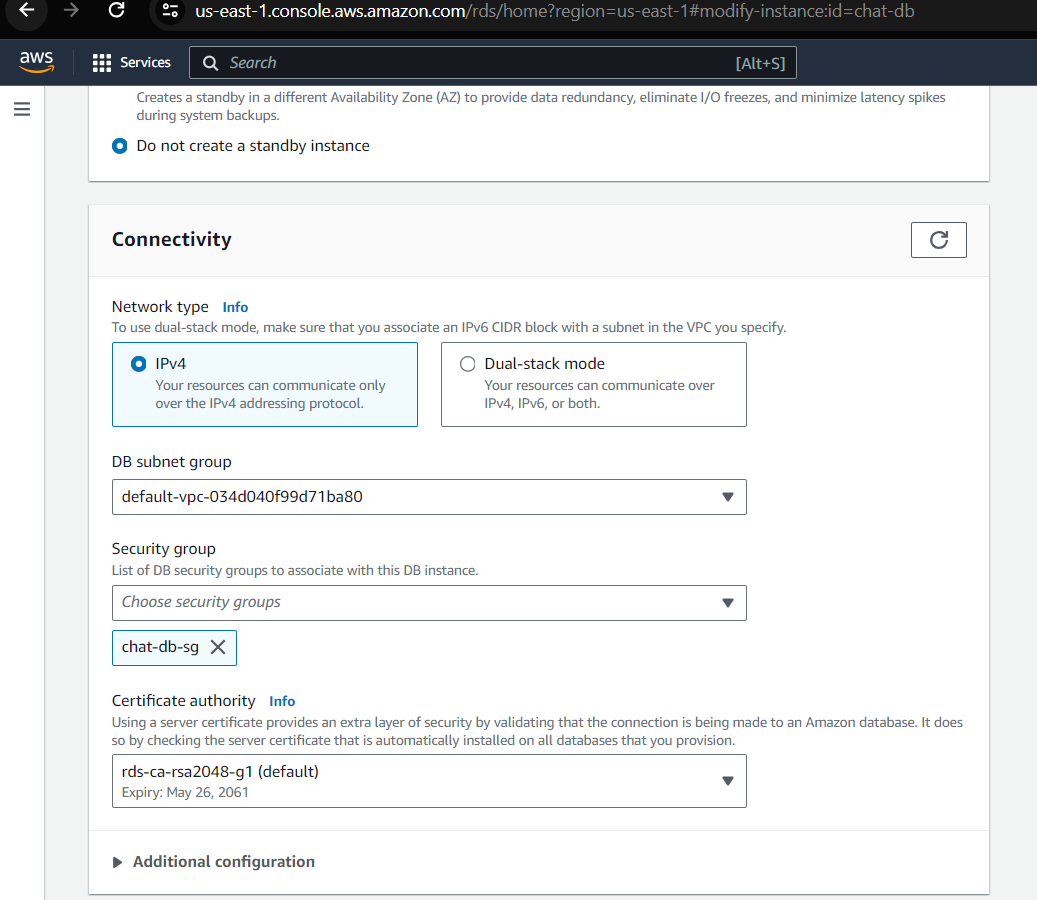
Create new SG group for AWS RDS Postgres DB.







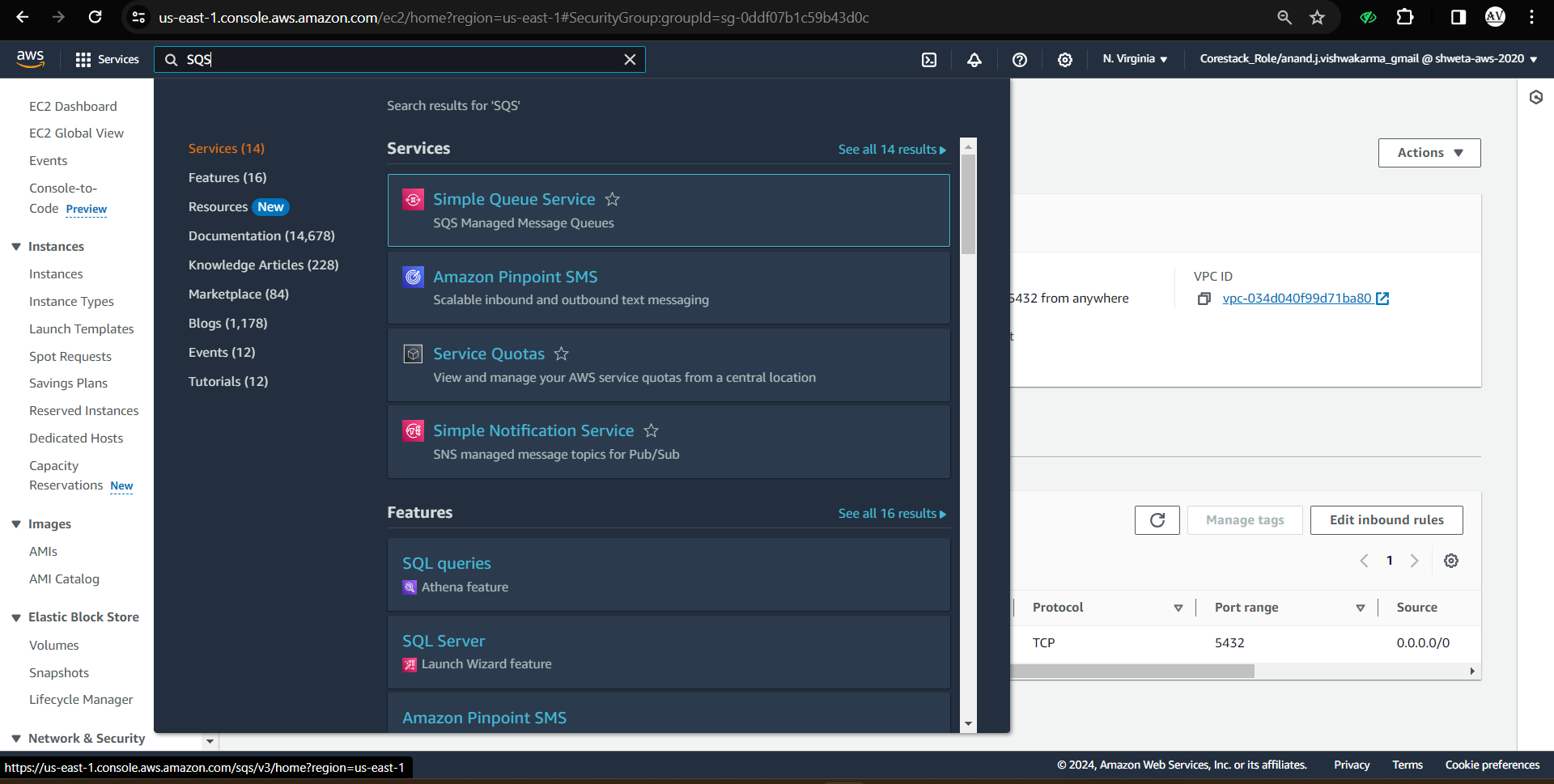
Now, go to the RDS, modify it and change the VPC Security group to the new one created.



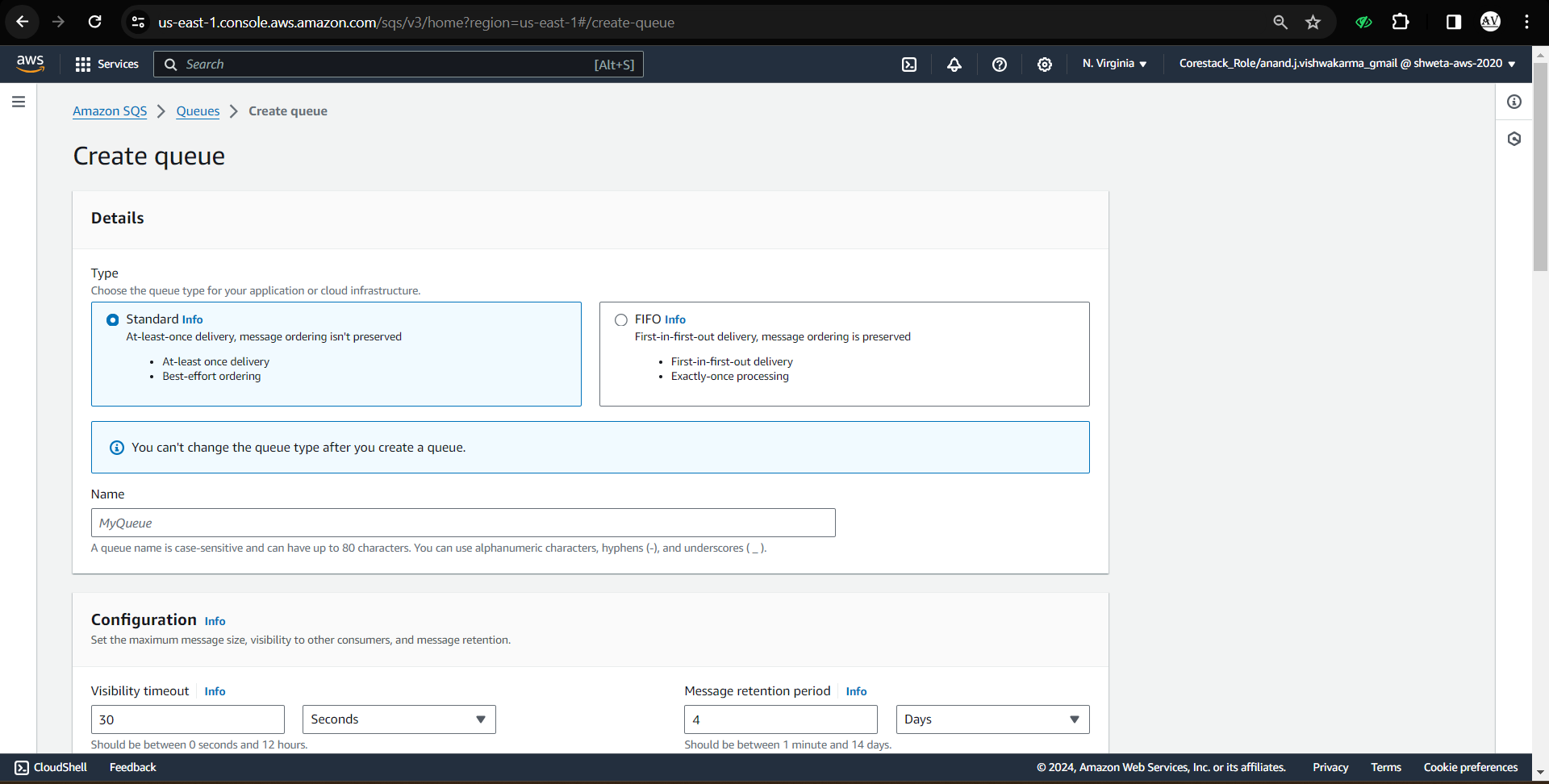
Now, the first step is completed successfully.

**Step 2:** Creating and configuring AWS SQS instance

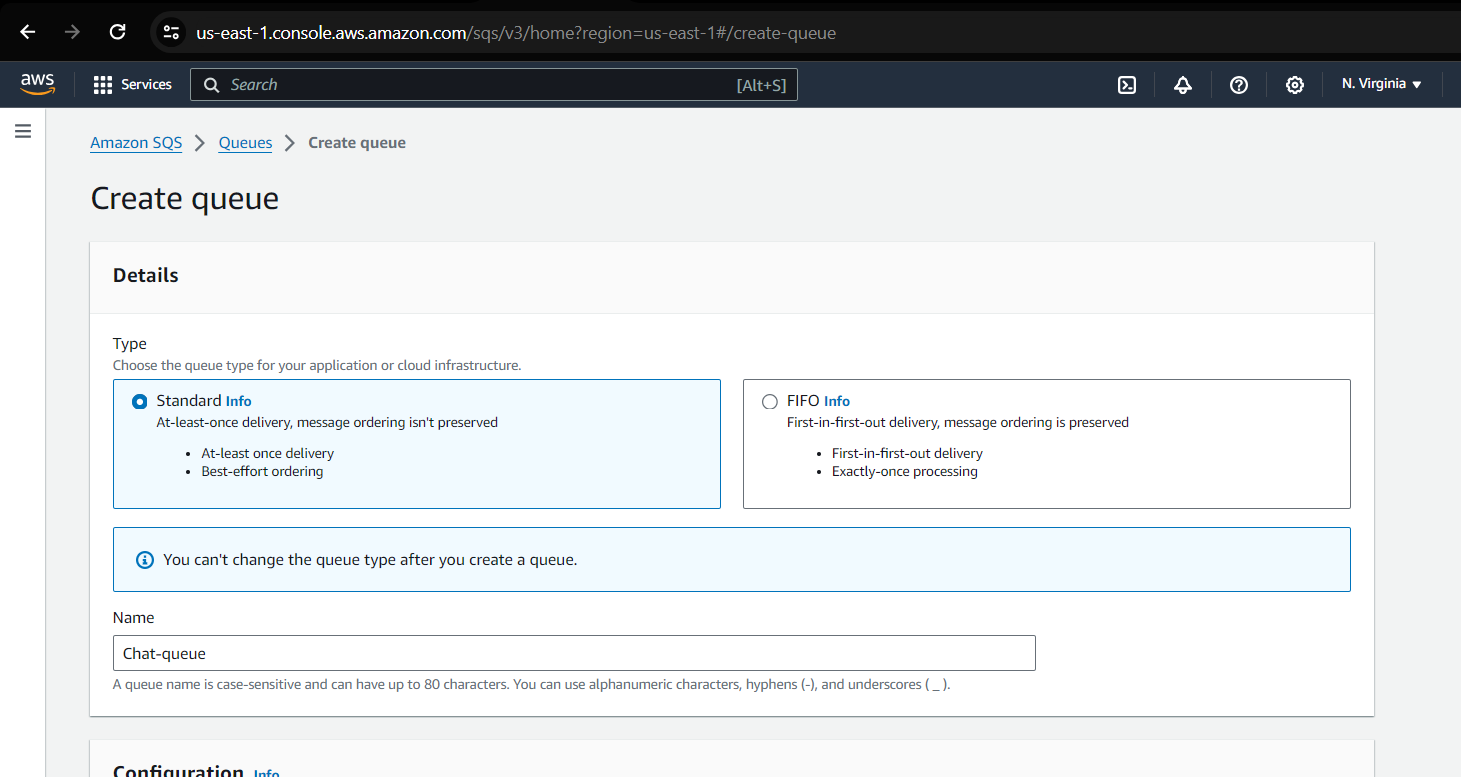
Go to AWS SQS and create new queue.



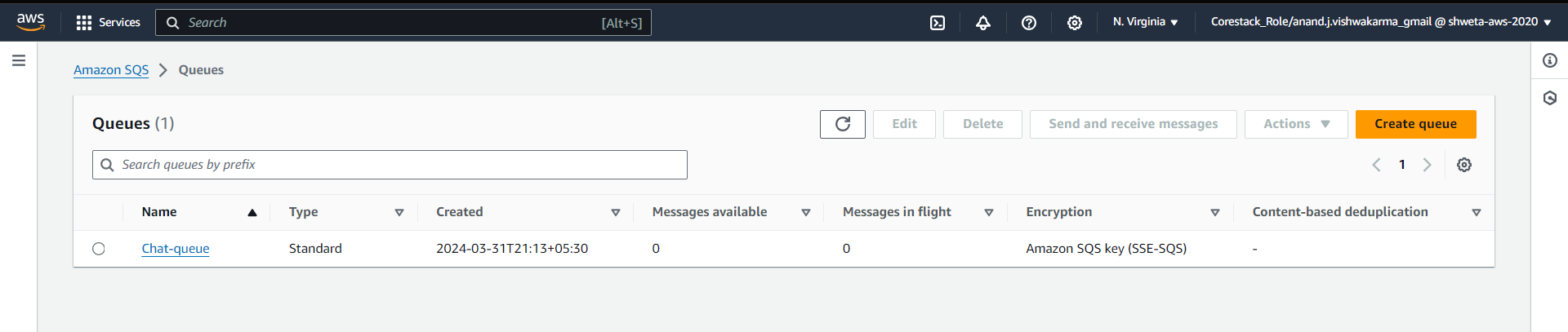
Select Standard method.



Enter Queue name.



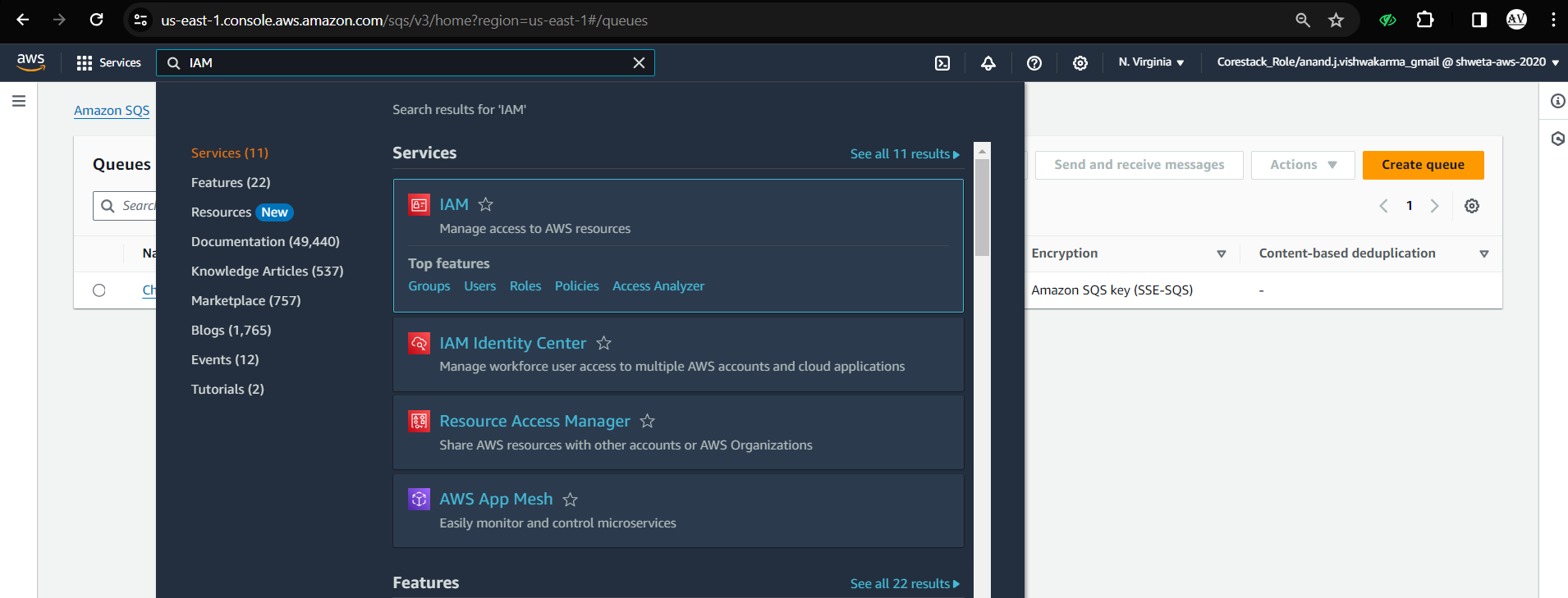
Leave rest of the options default and click on create queue button.



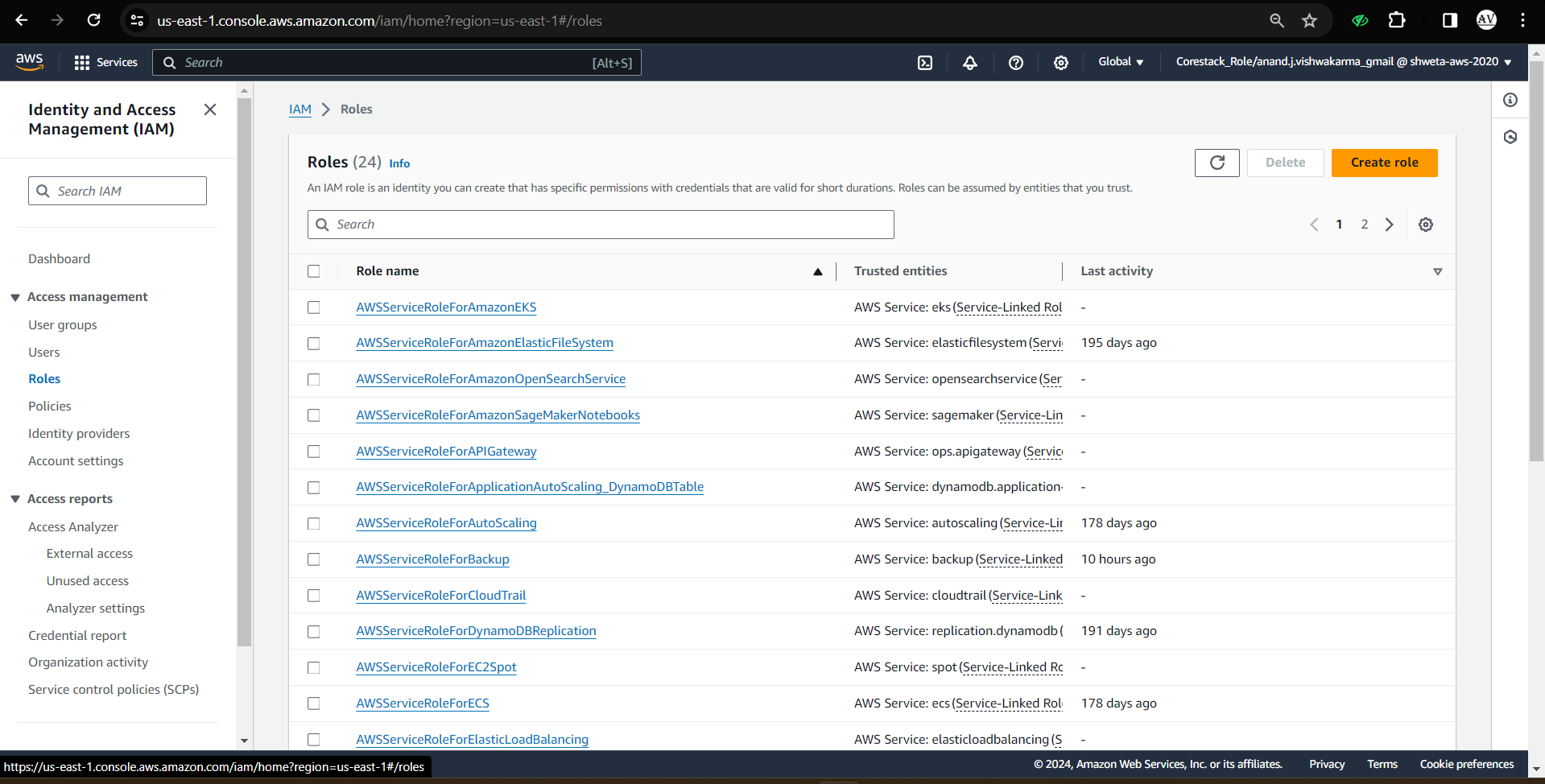
Here, we have successfully completed step 2.

**Step 3:** Creating and configuring IAM Role.

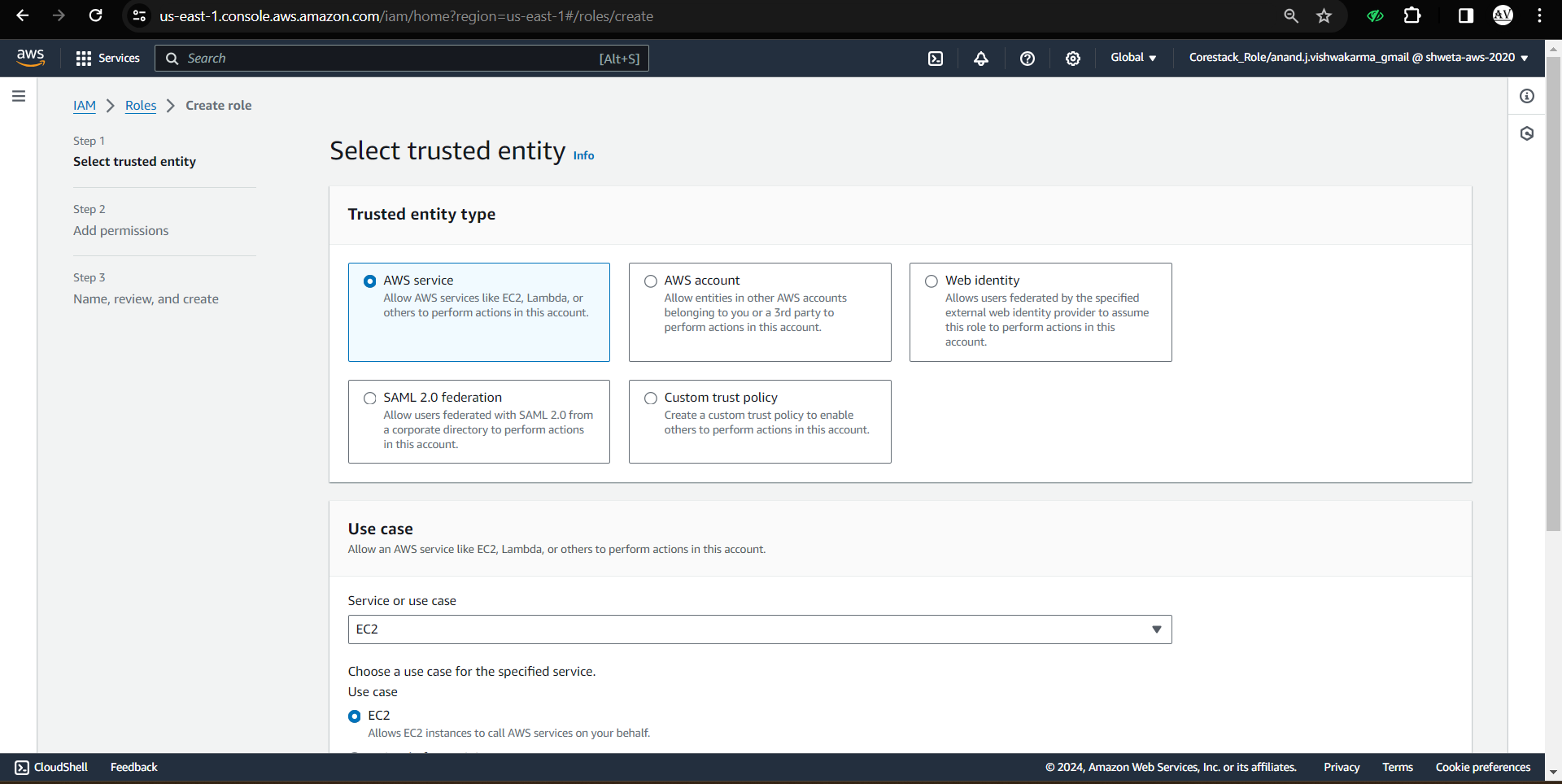
Go to IAM console.



Click on Role.

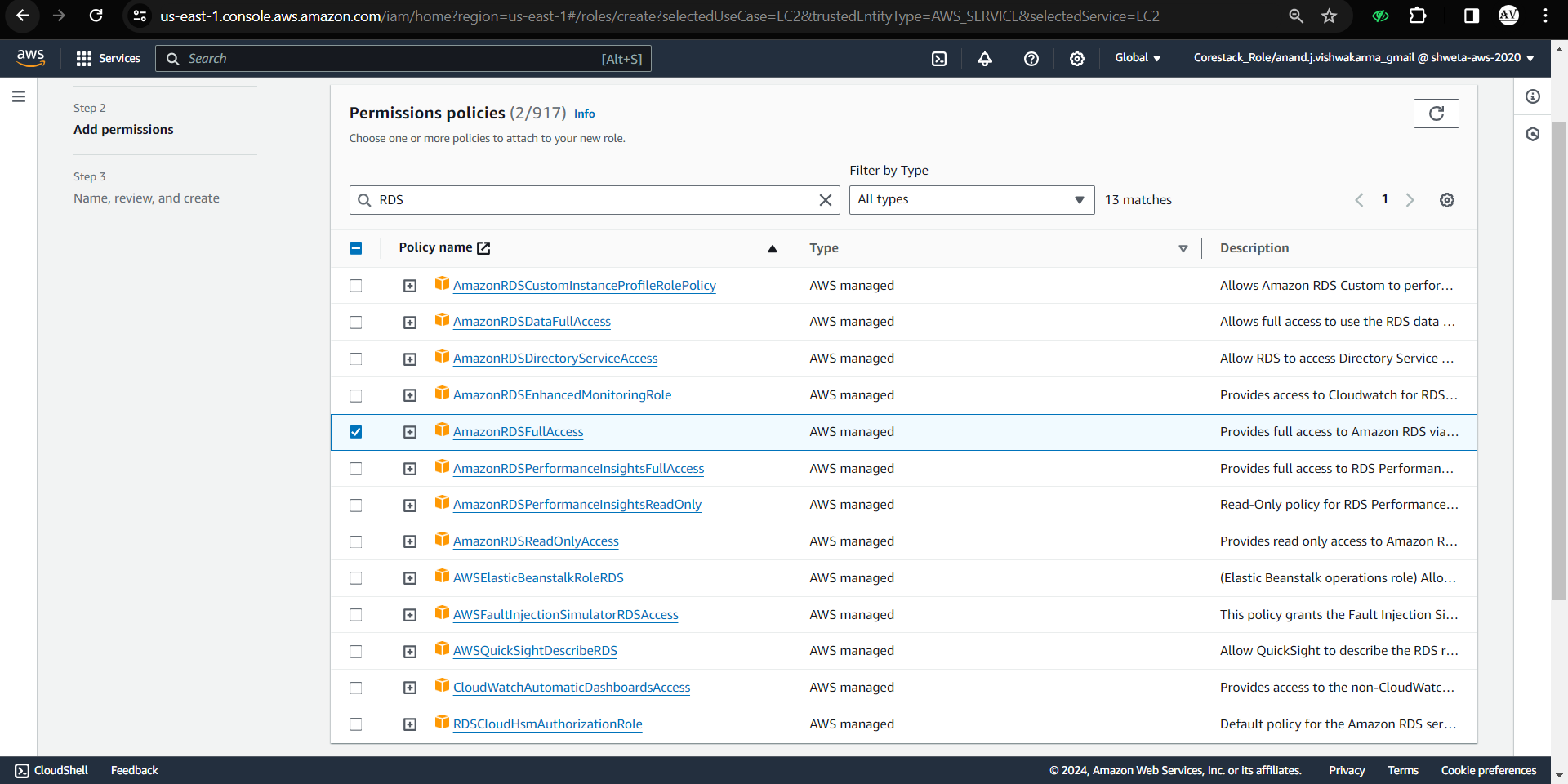


Select EC2 as use-case and keep rest default

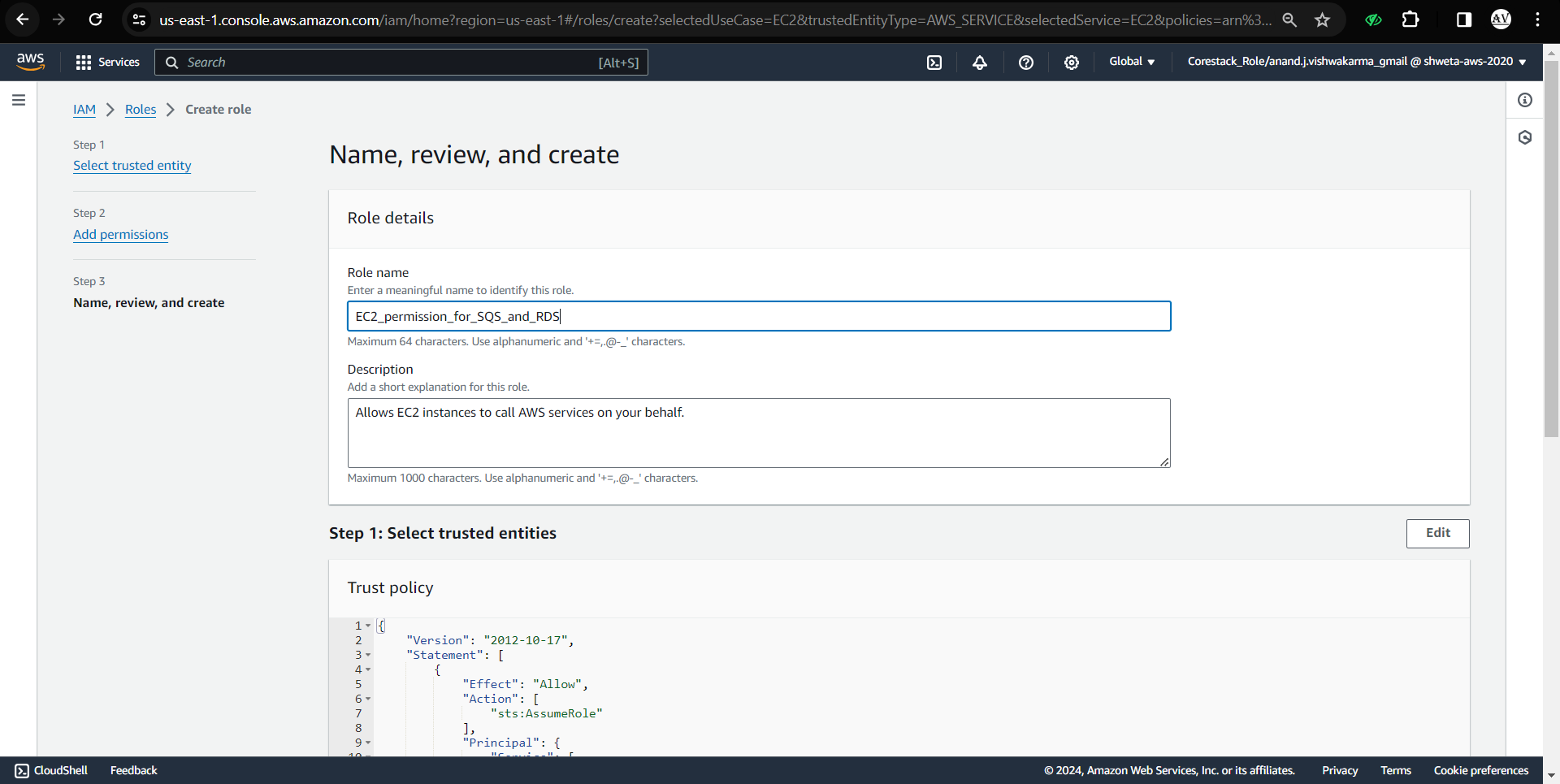


Now, Click on next and select SQS and RDS full access permissions as shown below.

Note: It's good to create separate roles and give only the permissions required (not full).



Enter the role name and click on create button.

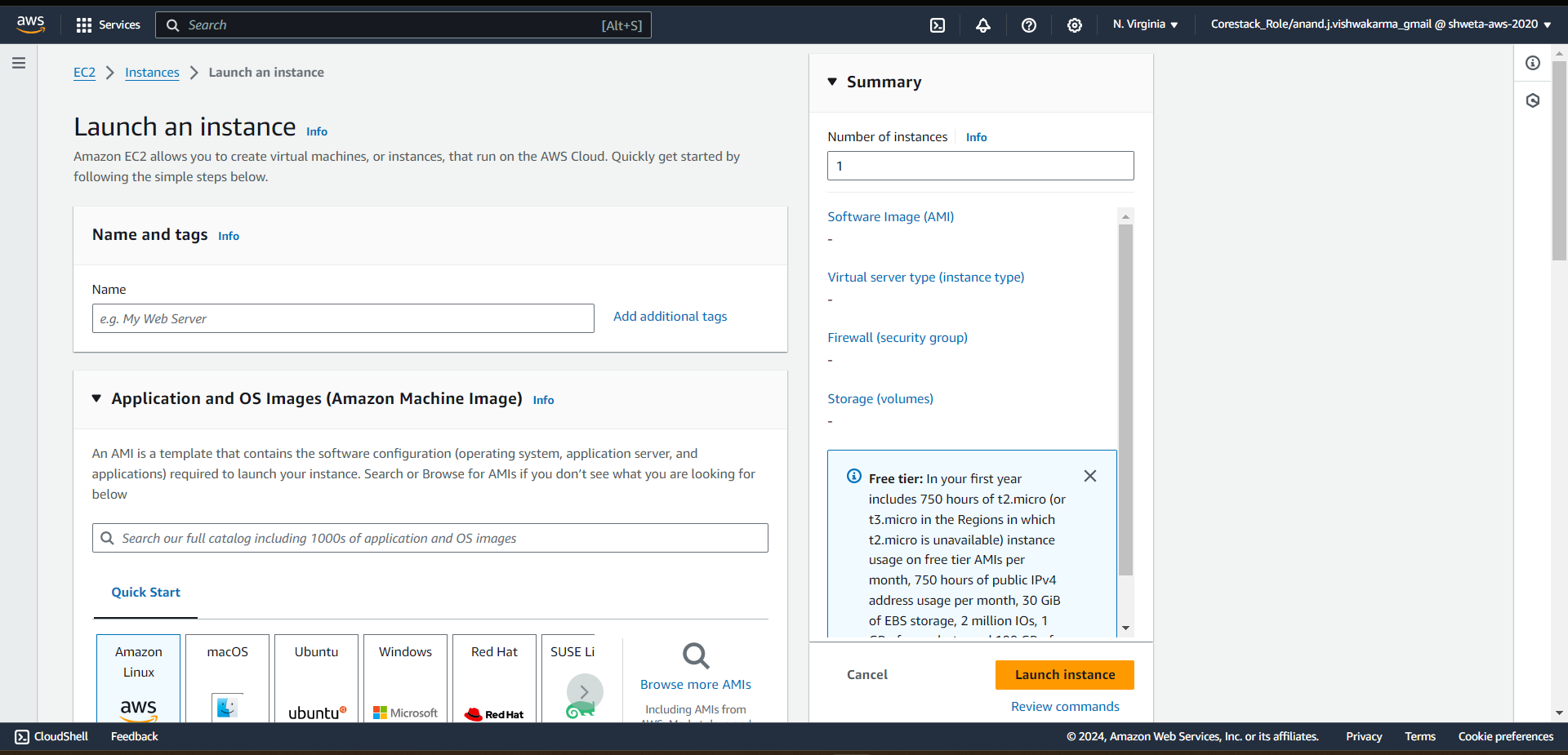


Third step is successfully configured.

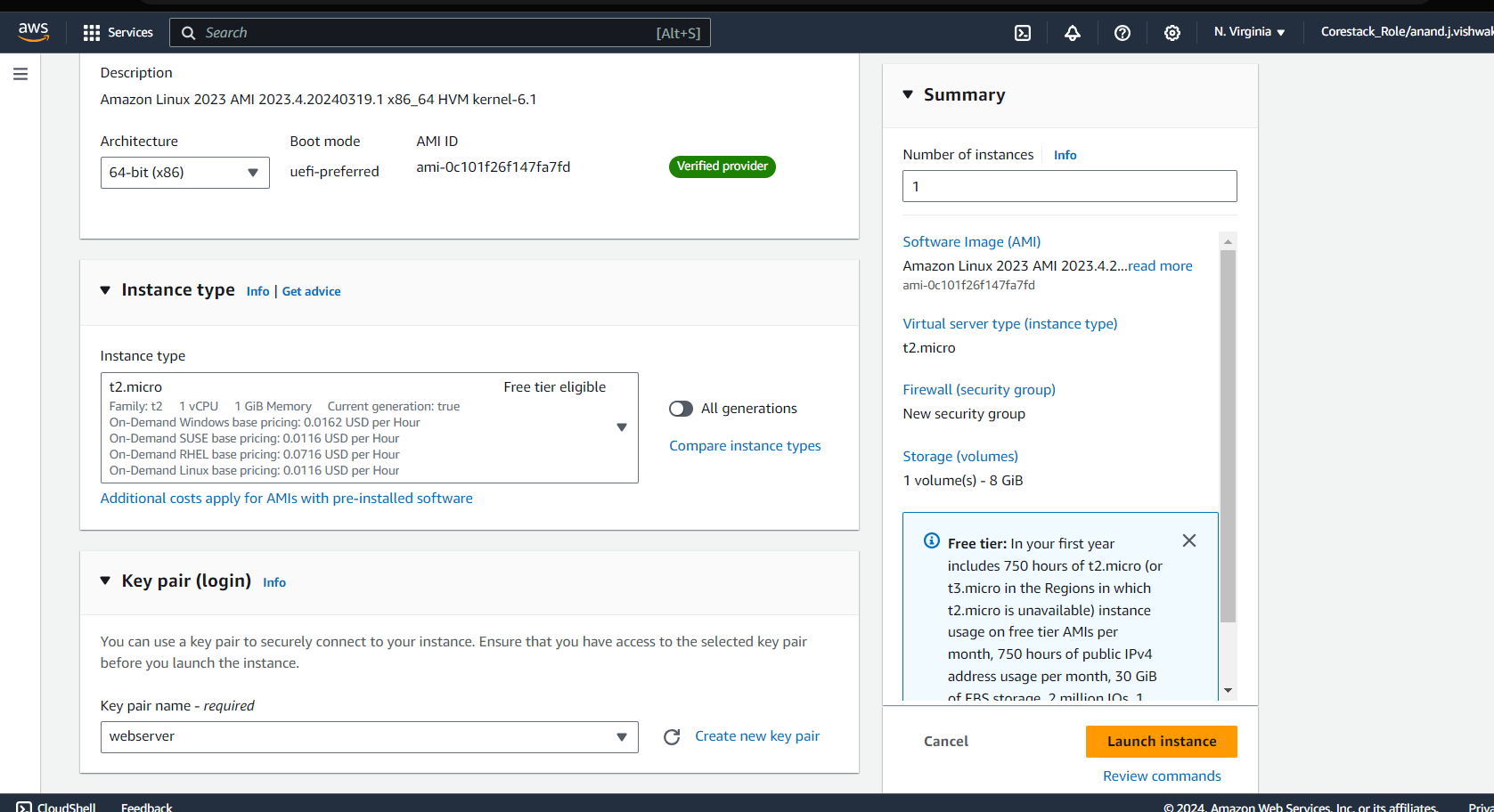
**Step 4:** Lets create and configure two EC2 instances.

Create two EC2 instances.

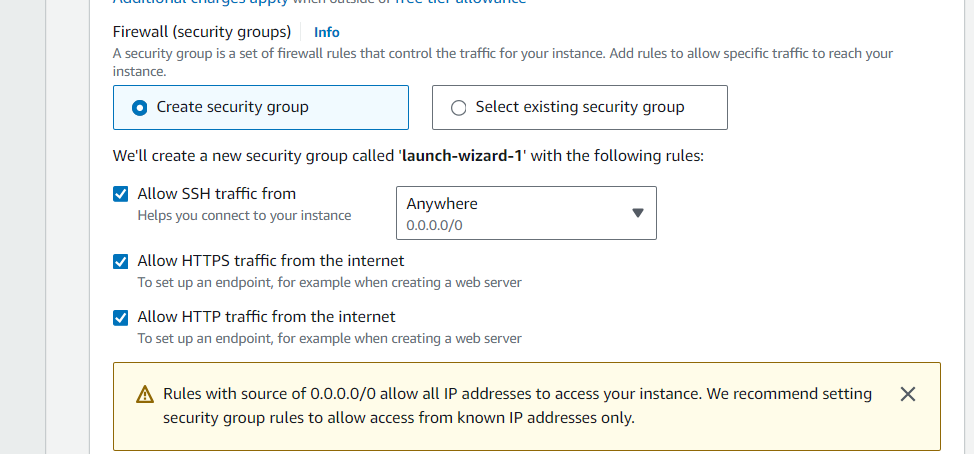
One for web-server and second for python application that will look to the SQS queue and if there is any message, it will send to RDS table and delete the message from the SQS queue.



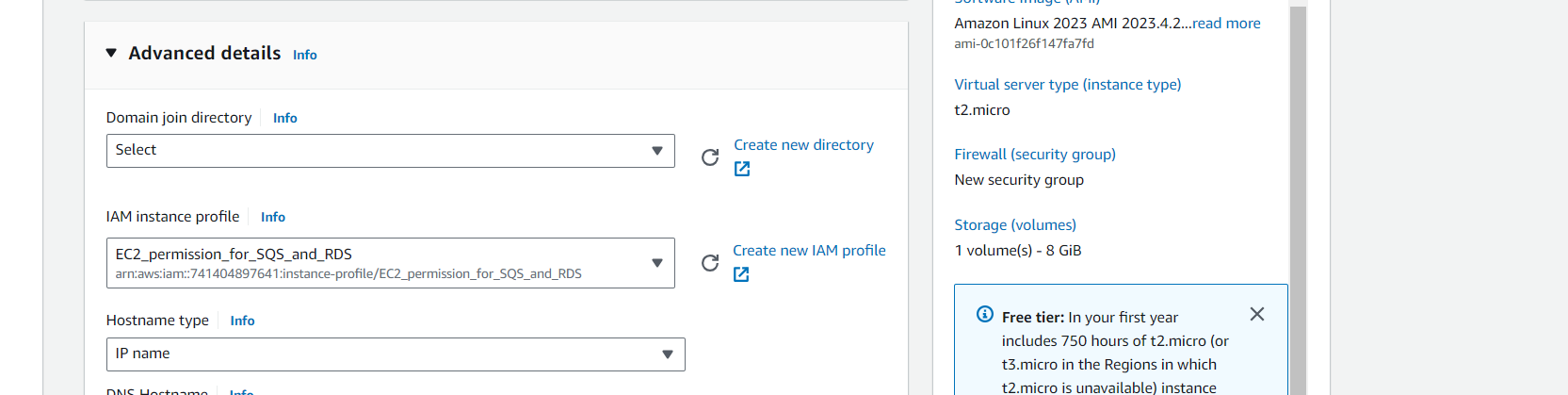
Create new key-pair and keep rest default.



Allow HTTP and HTTPS traffic.



In advance section, select IAM profile which we have created in step 3.



In user data, add below script to automatically install web-server and start the service.

#!/bin/bash

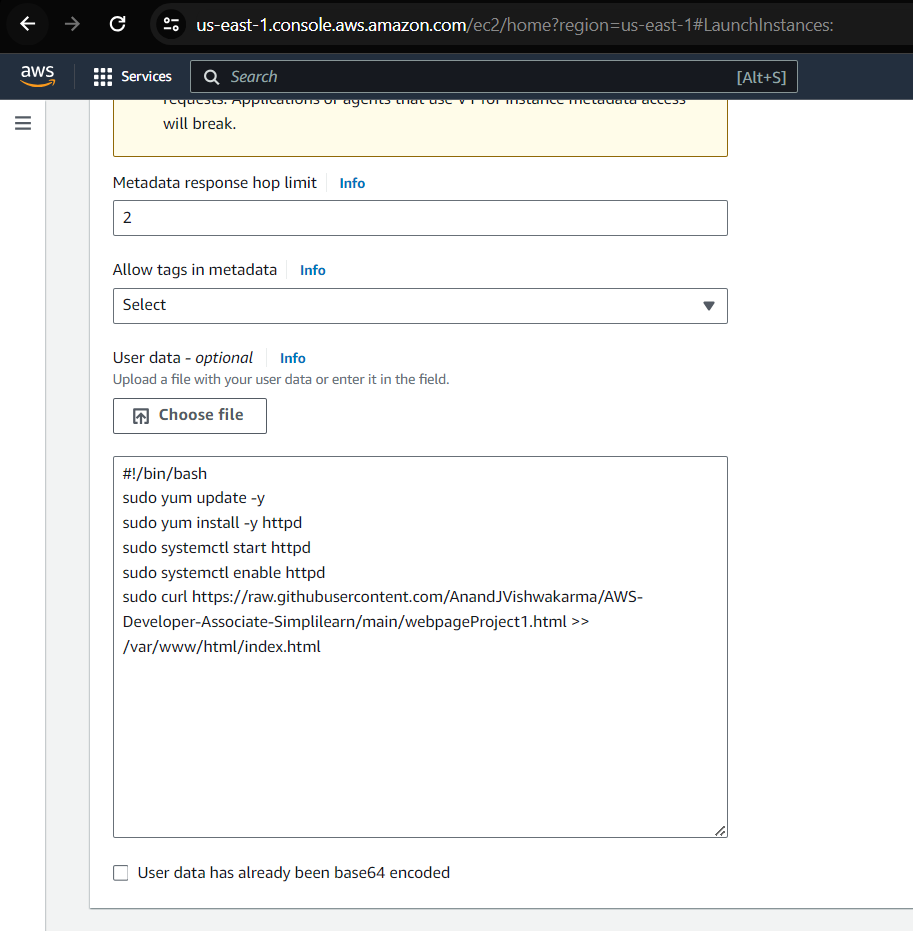
sudo yum update -y

sudo yum install -y httpd

sudo systemctl start httpd

sudo systemctl enable httpd

sudo curl https://raw.githubusercontent.com/AnandJVishwakarma/AWS-Developer-Associate-Simplilearn/main/webpageProject1.html >> /var/www/html/index.html



Once the configuration is done, click on create button.

Similar to above instance, configure second EC2 machine and add below script in user data.

#!/bin/bash

sudo yum update -y

sudo yum install -y httpd

sudo systemctl start httpd

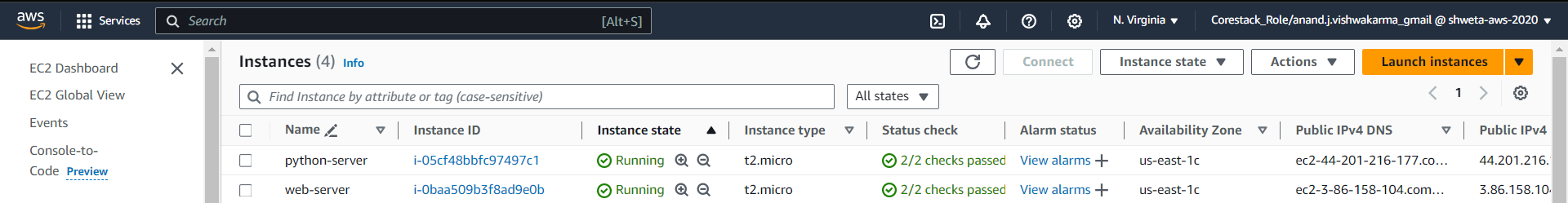
sudo systemctl enable httpd

sudo pip3 install boto3

sudo pip3 install pyscopg2-binary

sudo curl https://raw.githubusercontent.com/AnandJVishwakarma/AWS-Developer-Associate-Simplilearn/main/code.py >> /code.py

Once both machine are up and running, validate the web page using public IP of web-server machine.



Now you can access the website and submit a contact message. It'll send to SQS and the python

application take the message and add to the RDS table.

If the DataBase is offline the message will still send and when the DataBase will be online all the

Messages will be send to RDS table.

**Kudos!!! We have successfully implemented the “AWS SQS to manage a tightly coupled architecture”.**