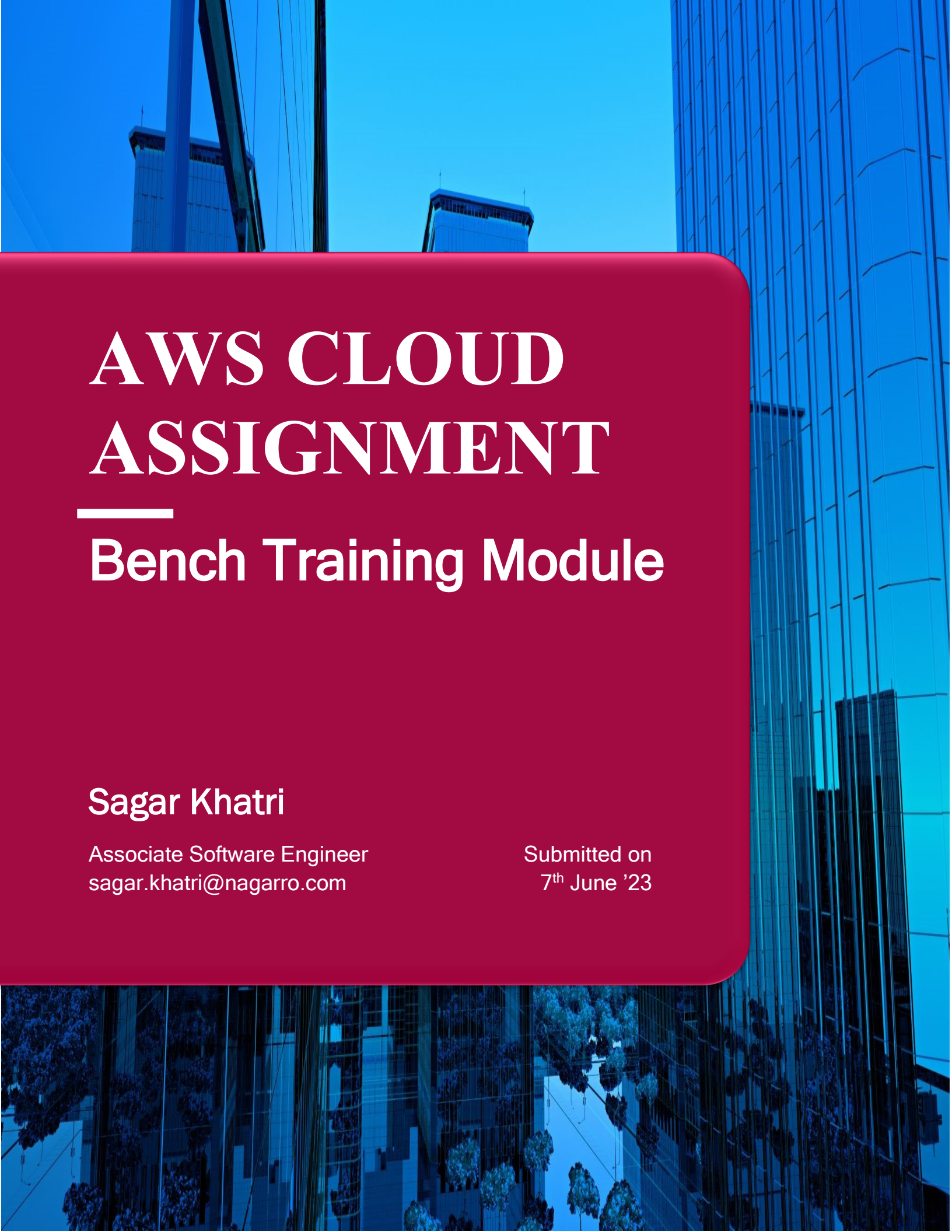


AWS CLOUD ASSIGNMENT

Bench Training Module

Sagar Khatri

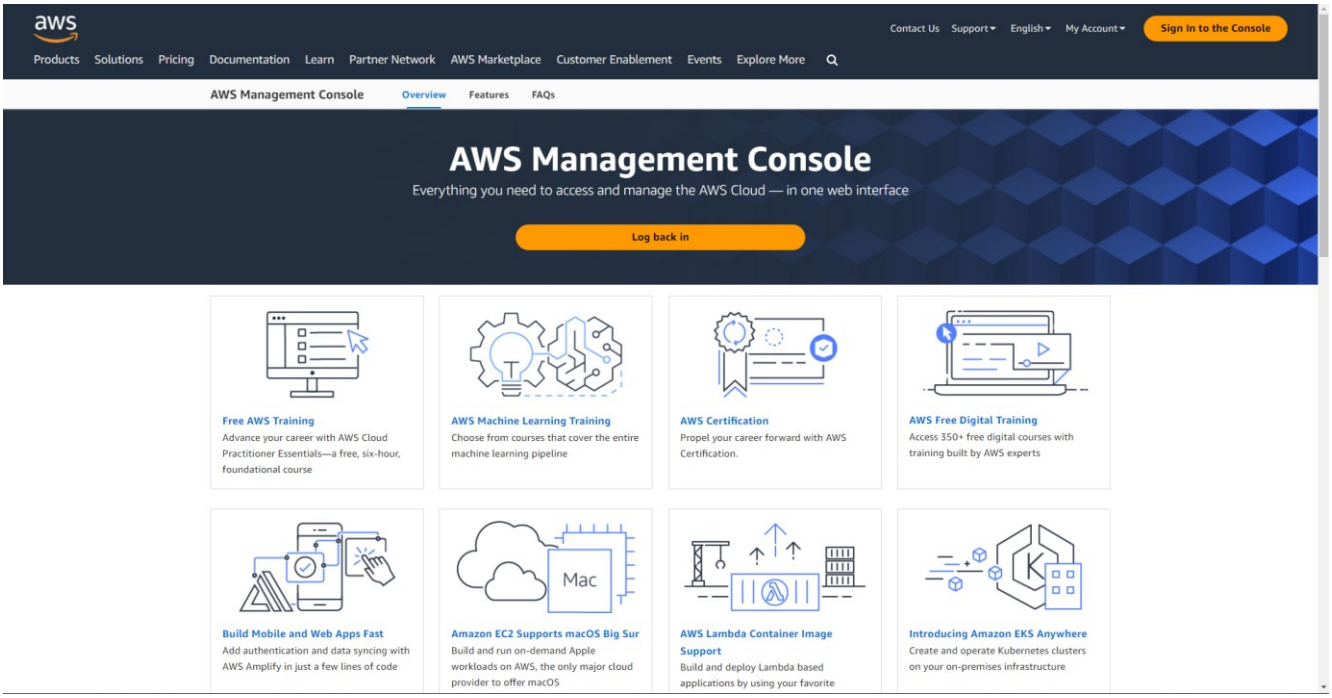
Associate Software Engineer
sagar.khatri@nagarro.com

Submitted on
7th June '23

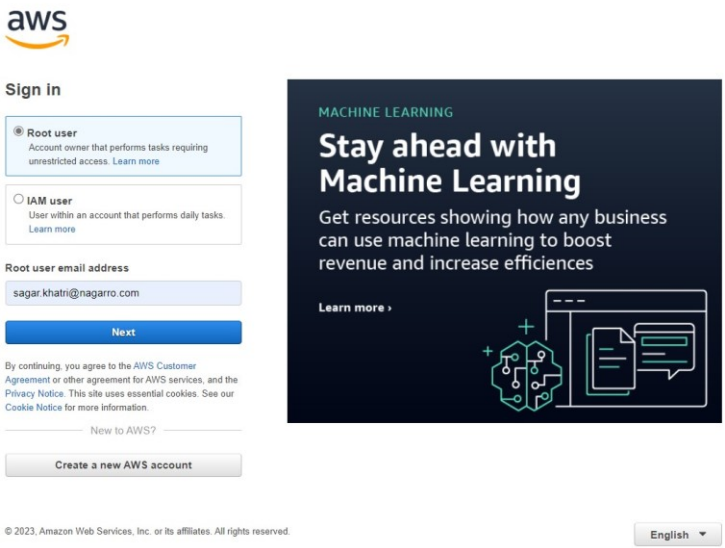
Assignment: AWS Cloud Assignment

Step-by-step Procedure

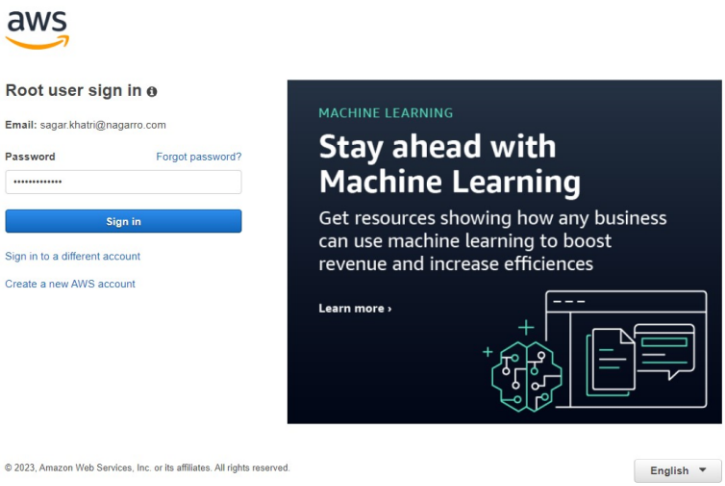
Open the link for AWS Management Console: https://aws.amazon.com/console/



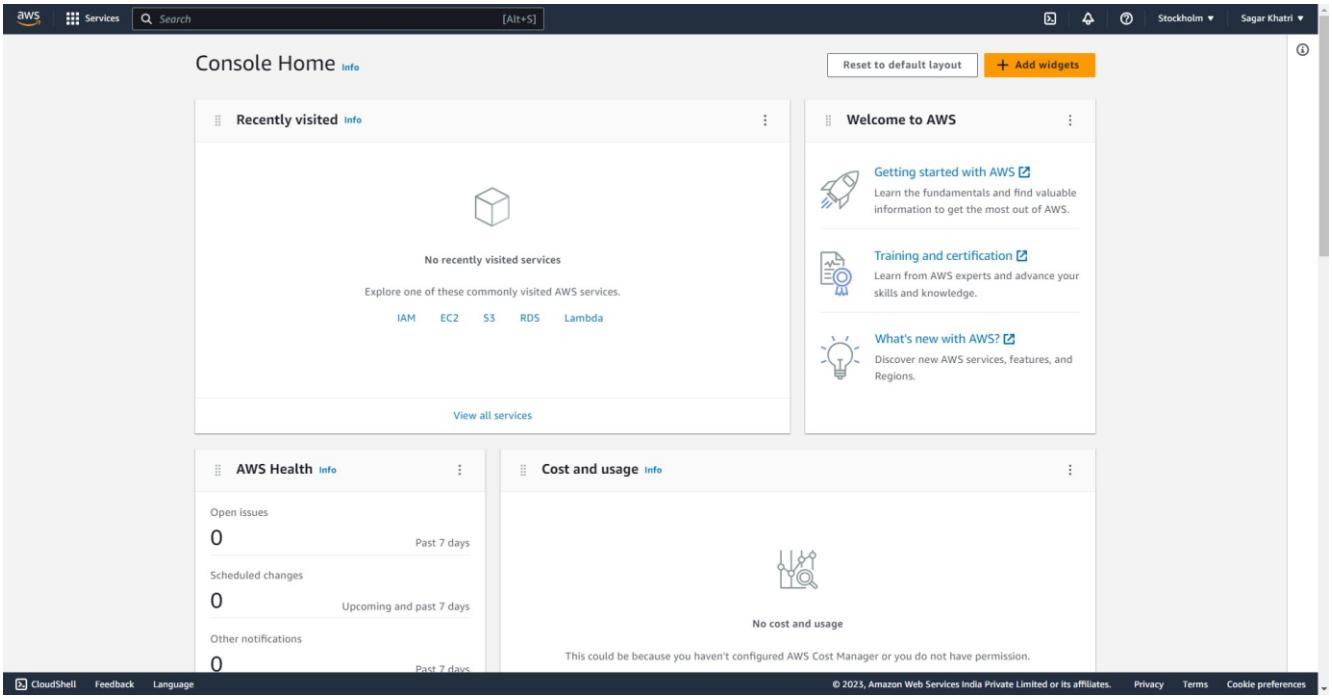
Click on login button which will take you to login page.



Login as root user, and enter your email address, then enter your password :



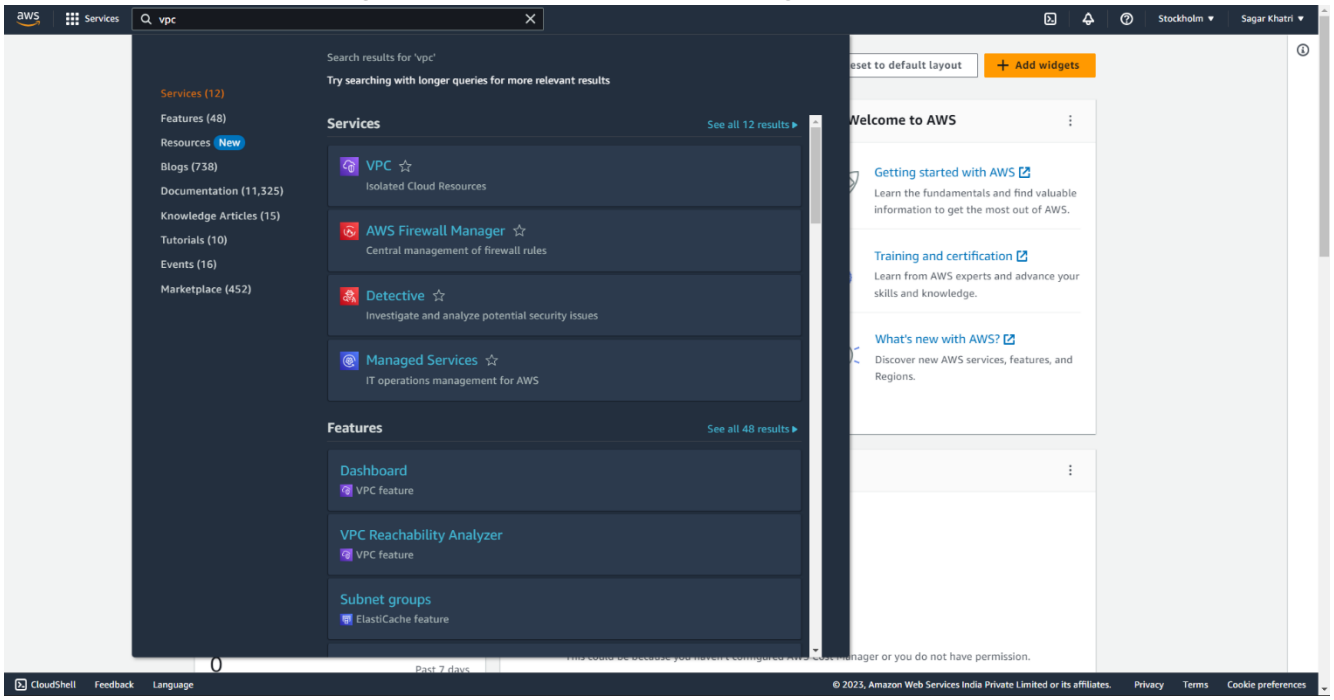
This will take you to AWS Console Home



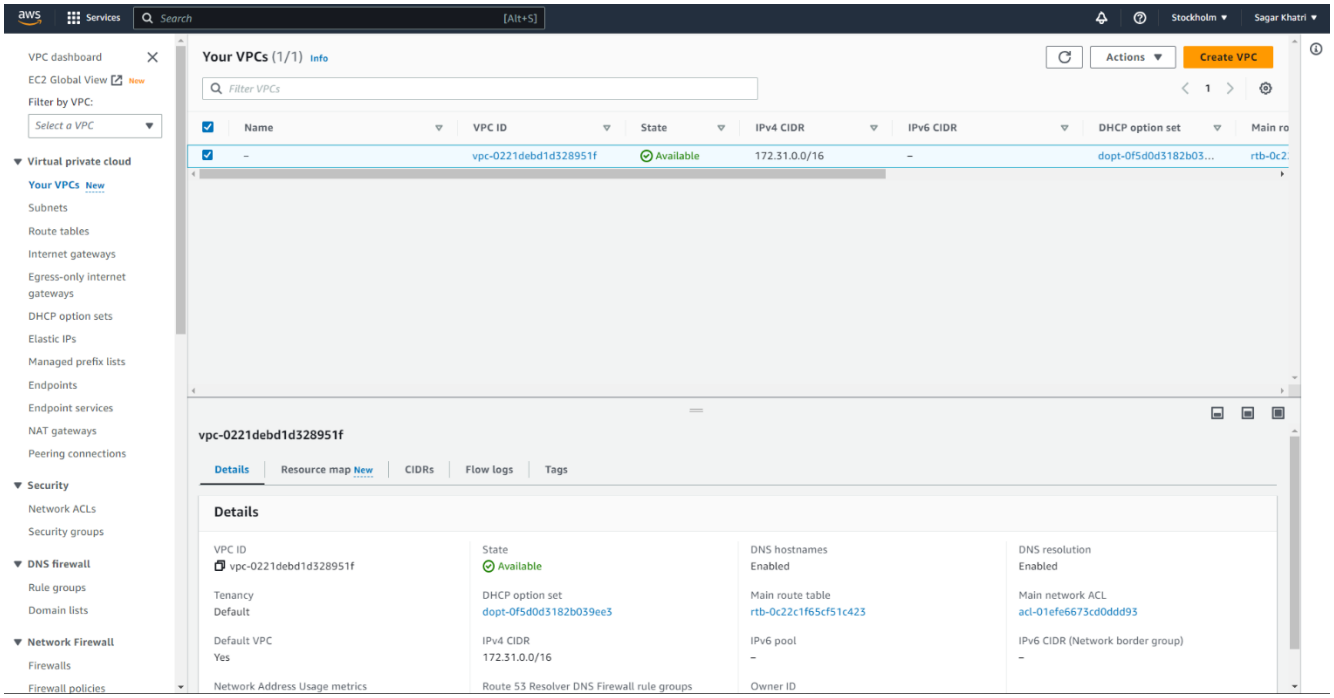
Now we can proceed with the assignment instructions from next page.

Creating a Virtual Network (VPC) with 2 Subnets:

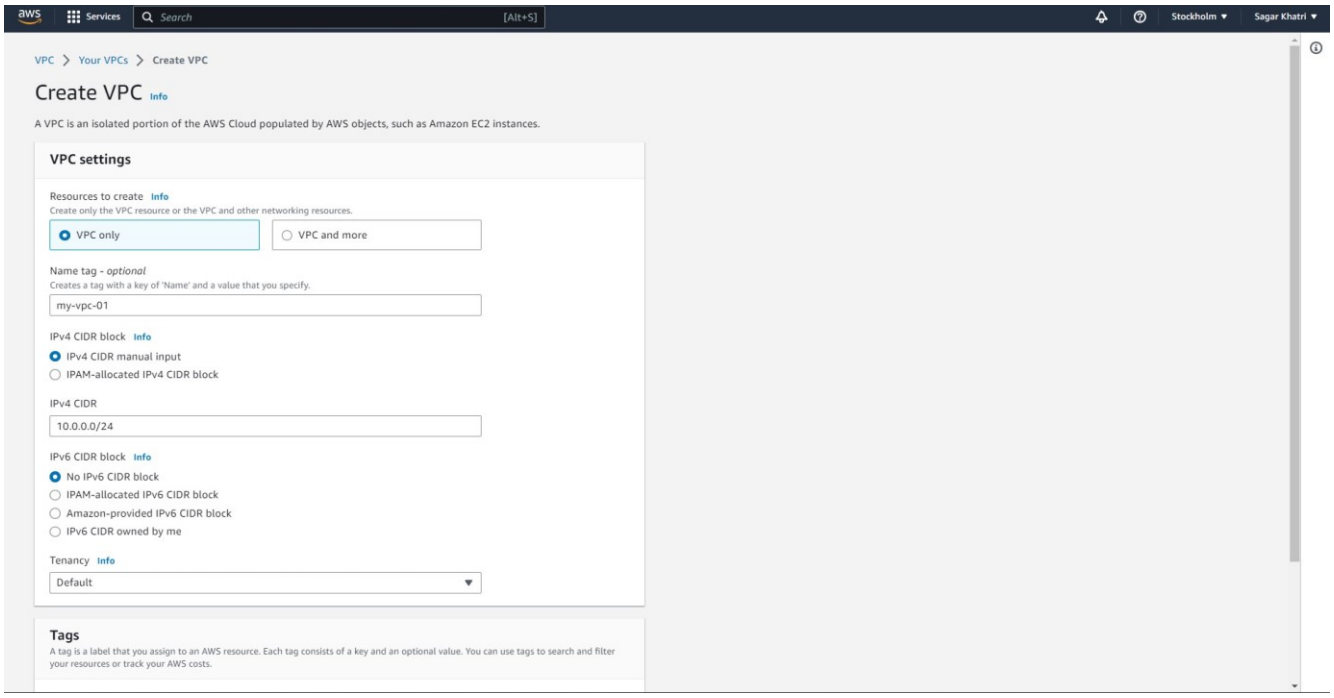
a) Open the AWS Management Console and navigate to the VPC service.



b) Click on "Your VPCs" in the left-hand menu.



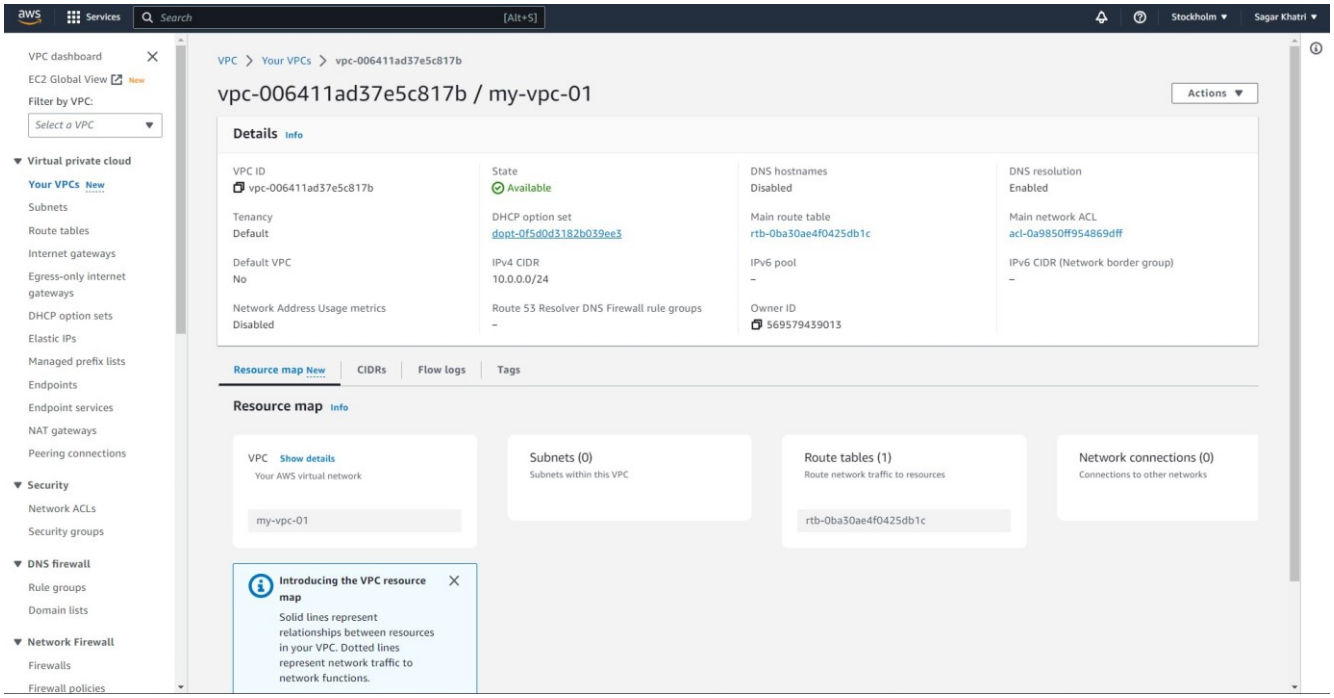
c) Click on the "Create VPC" button.



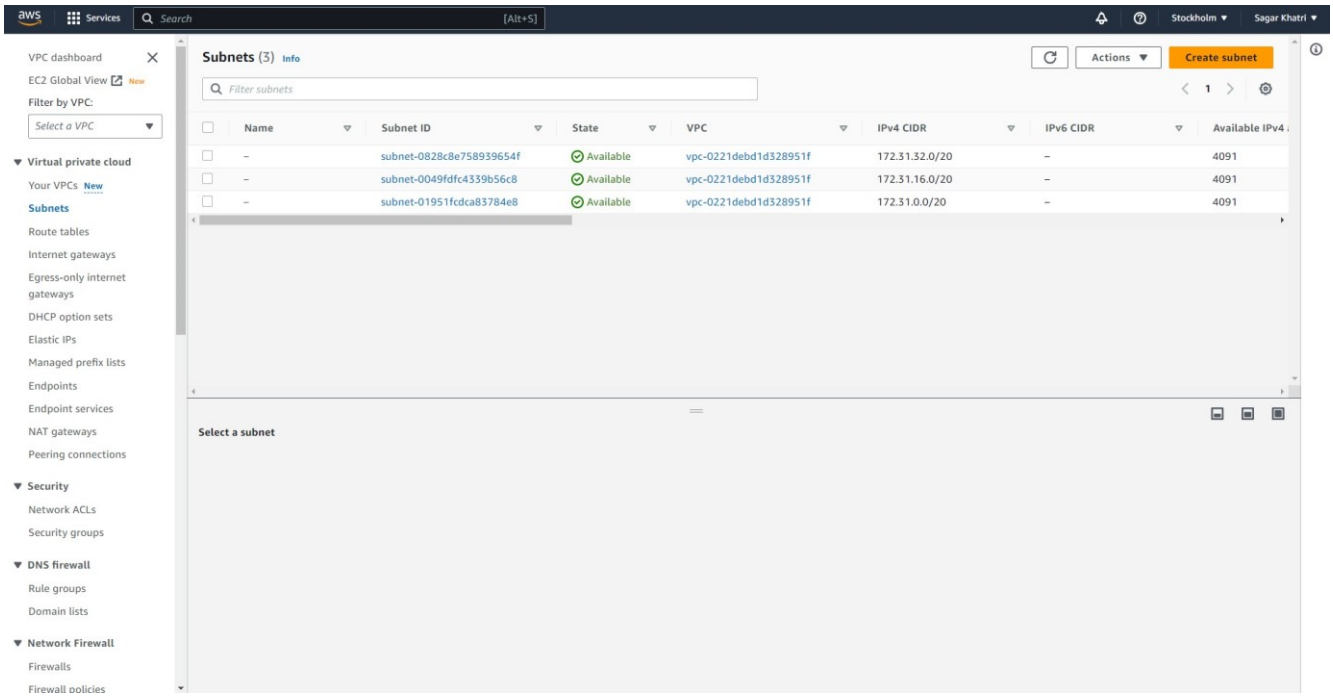
d) Enter the following details:

- VPC name: my-vpc-01
- IPv4 CIDR block: 10.0.0.0/24

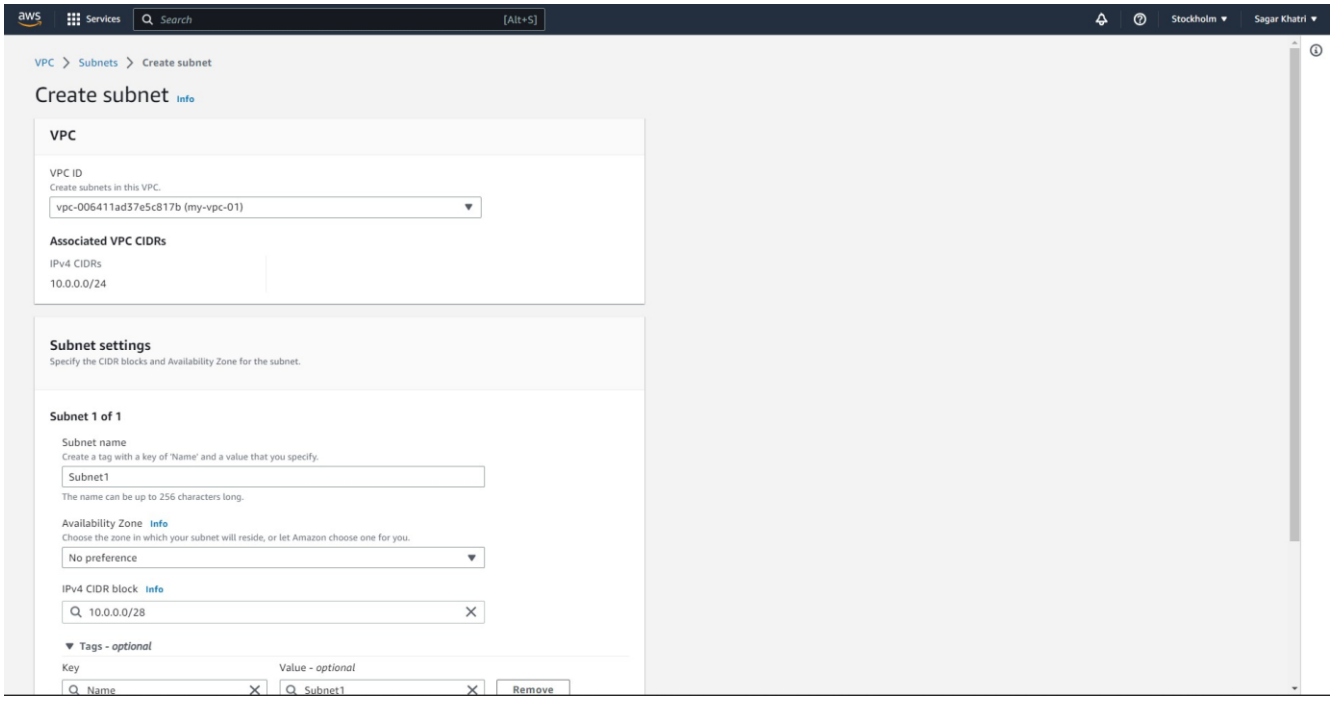
e) Click on the "Create" button to create the VPC.



f) Once the VPC is created, click on "Subnets" in the left-hand menu.



g) Click on the "Create subnet" button.

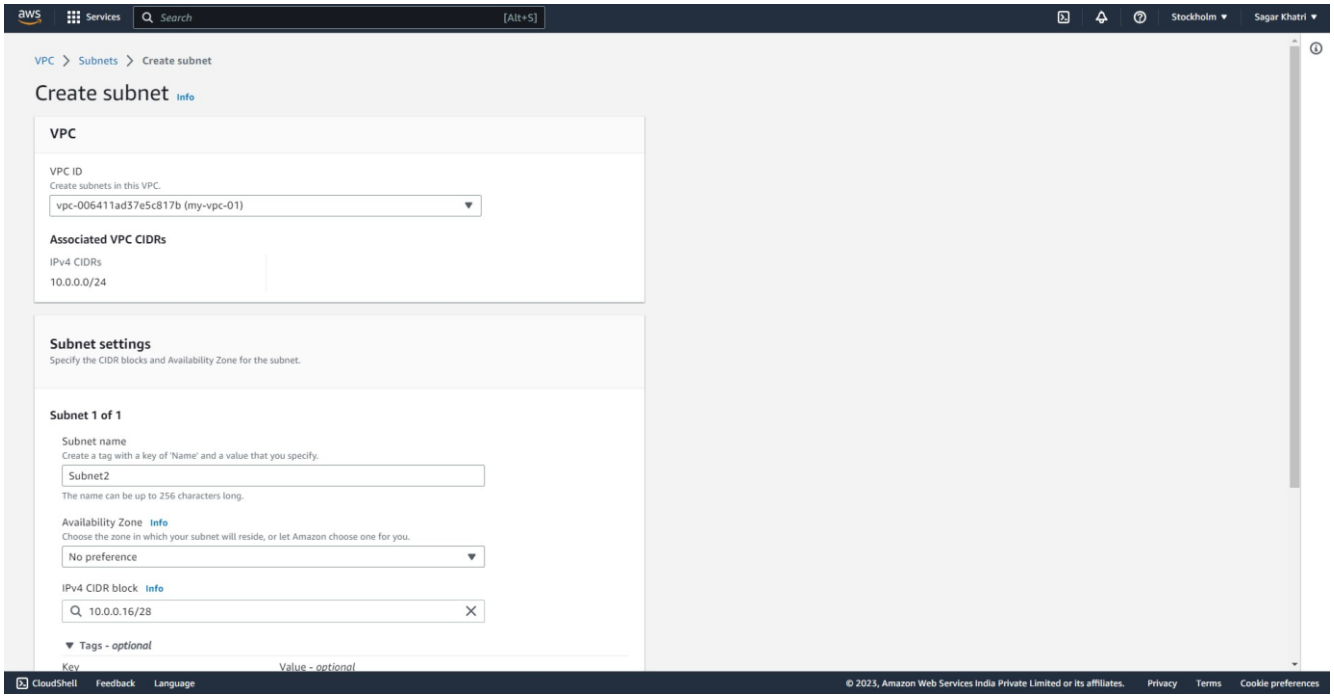


h) Enter the following details for the first subnet:

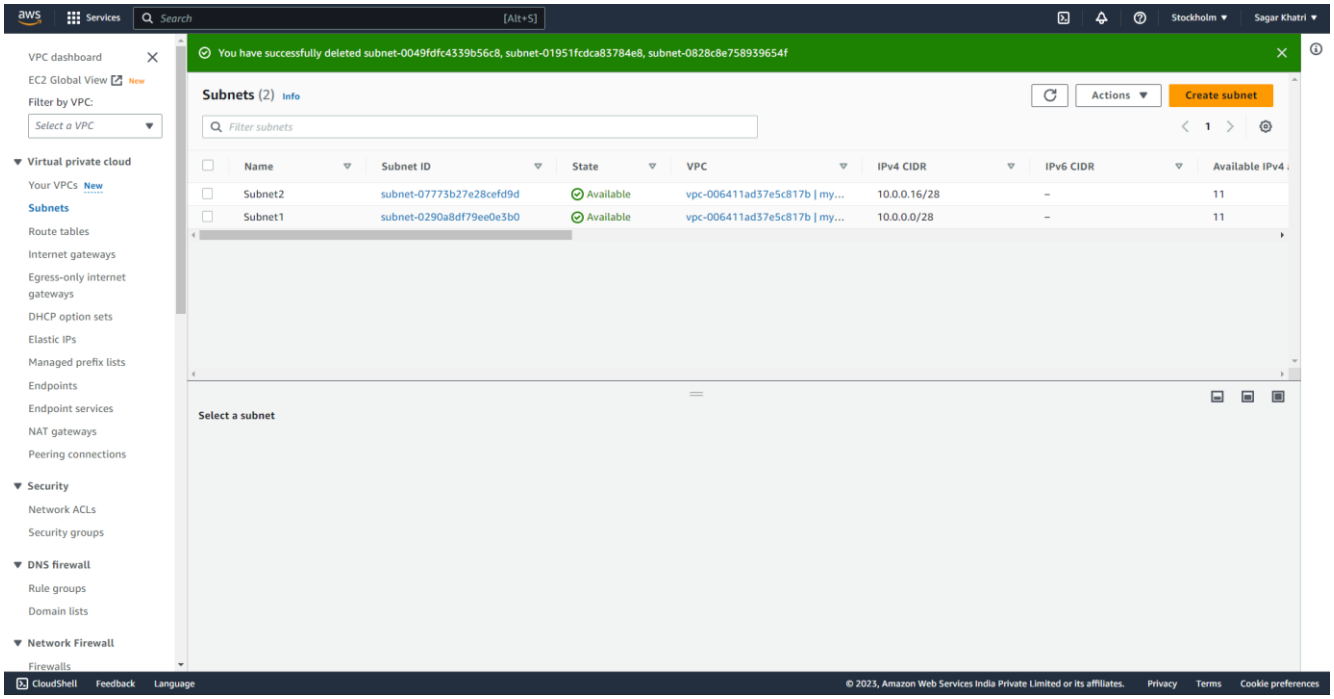
- Name tag: Subnet1
- VPC: my-vpc-01
- Availability Zone: Stockholm
- IPv4 CIDR block: [Enter the CIDR block for the subnet]

i) Click on the "Create" button to create the first subnet.

j) Repeat steps h-i to create the second subnet, using a different CIDR block within the VPC.



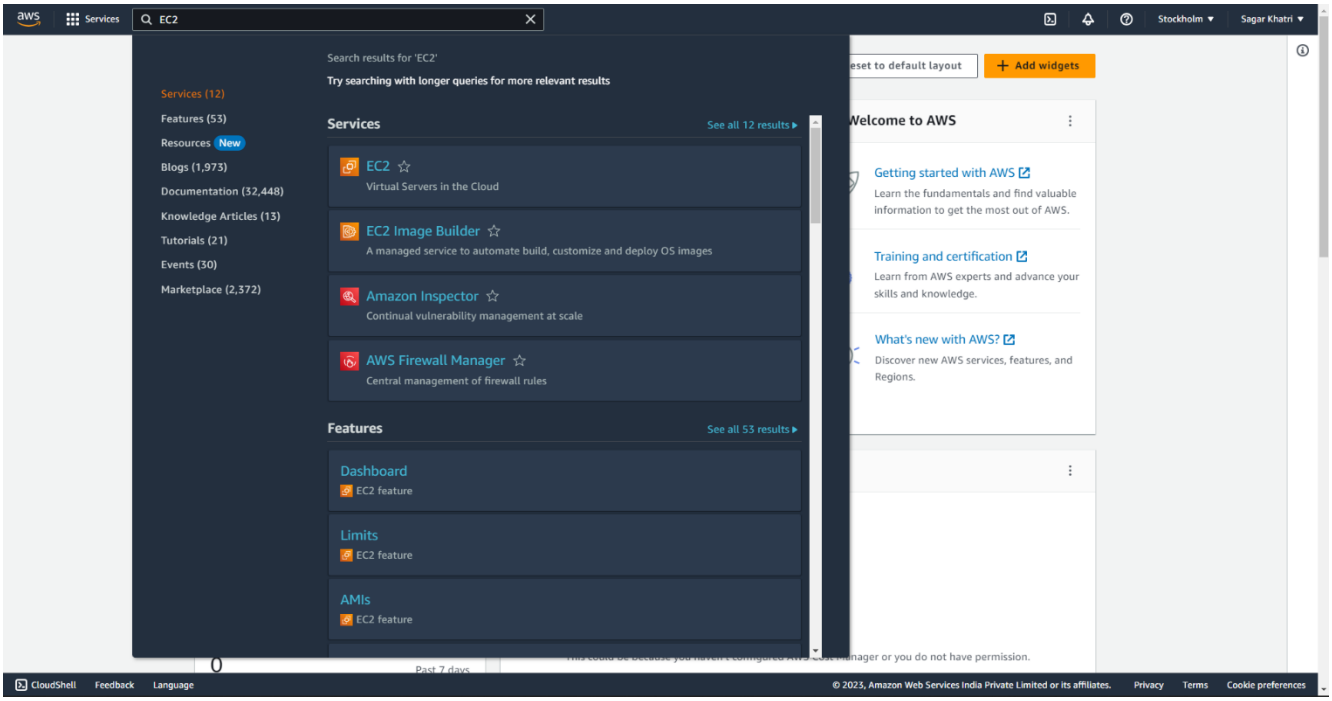
k) View the created subnets



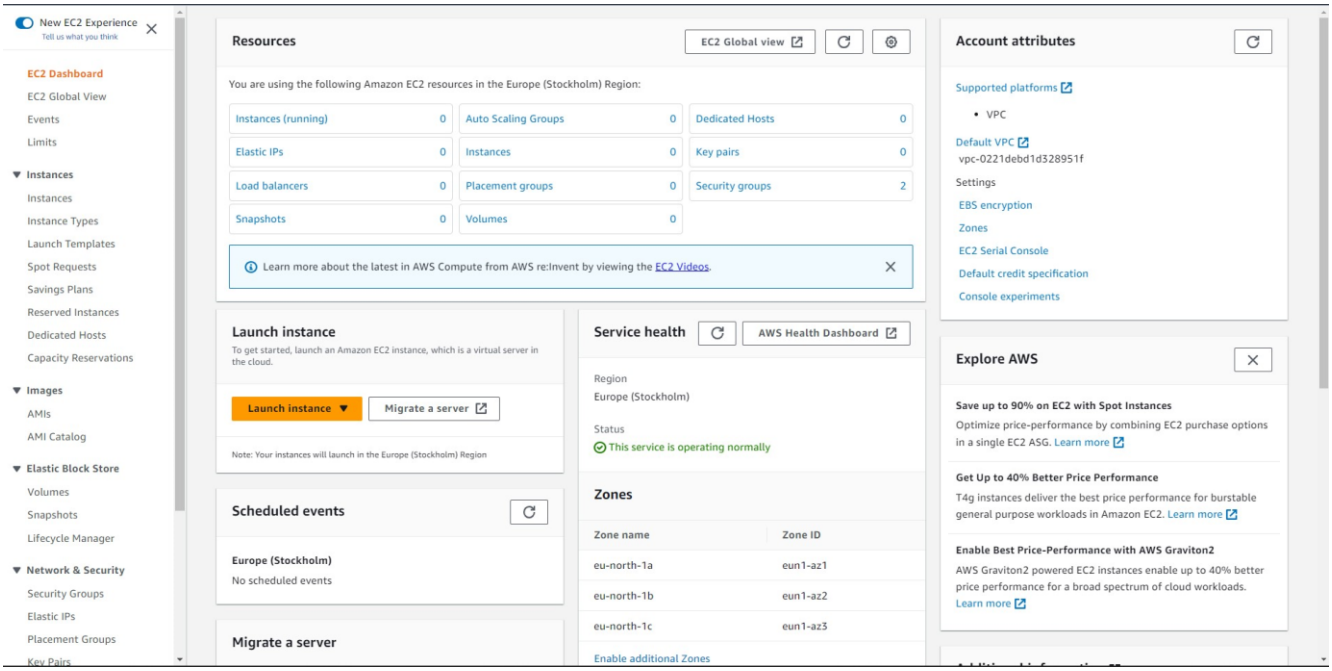
Creating a VM and Deploying Application Code in one of the Subnets:

a) Launch an EC2 Instance in the Desired Subnet:

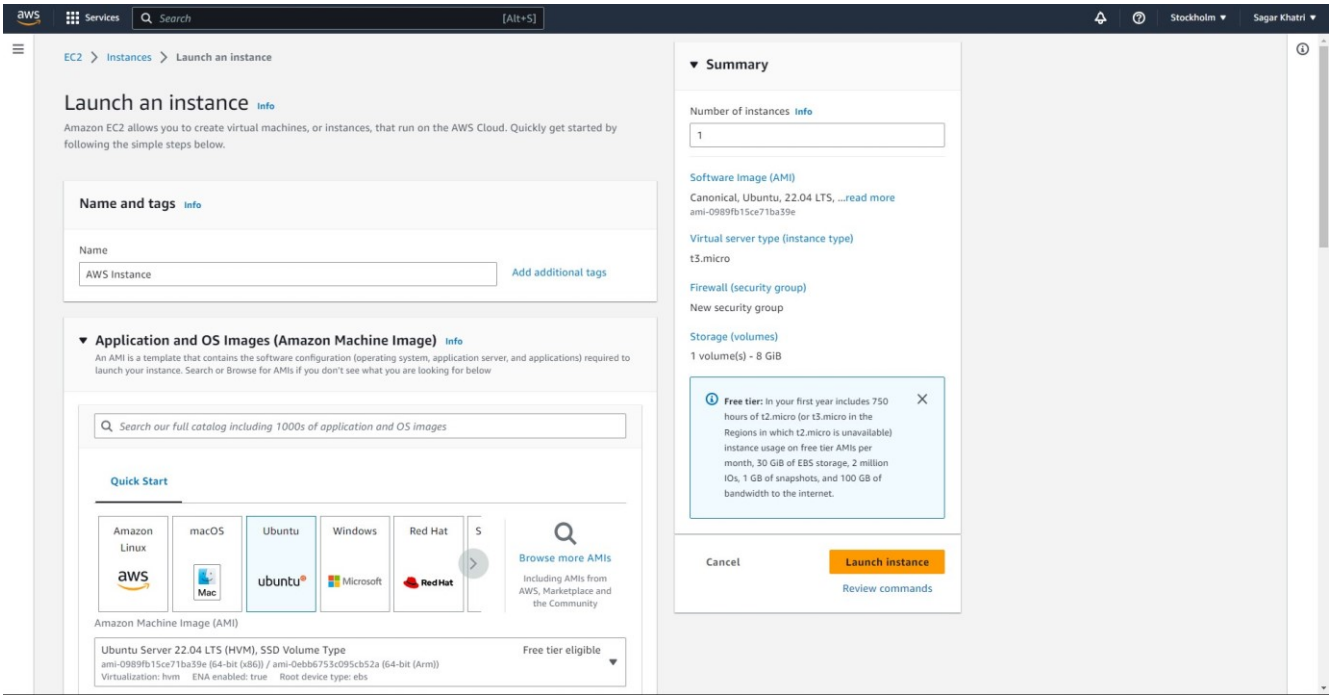
- Open the AWS Management Console and navigate to the EC2 service.



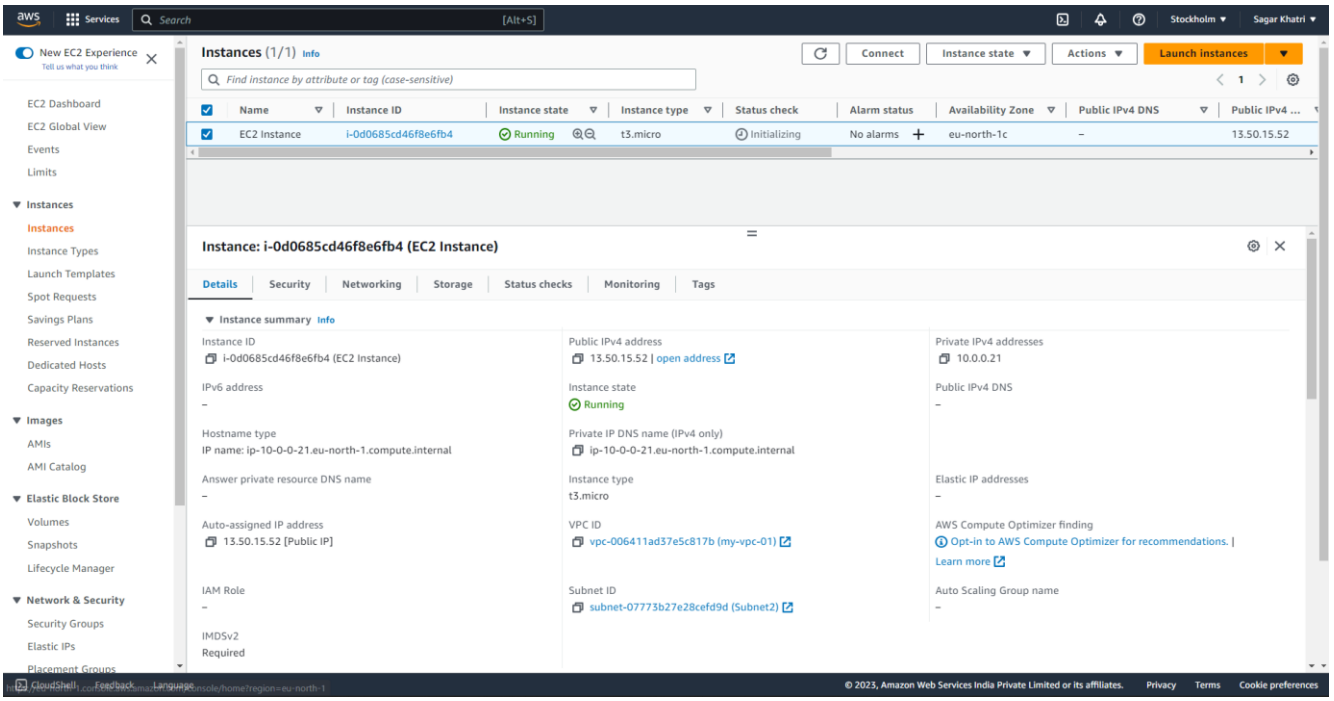
- Click on "Instances" in the left-hand menu.



- Click on the "Launch Instances" button.

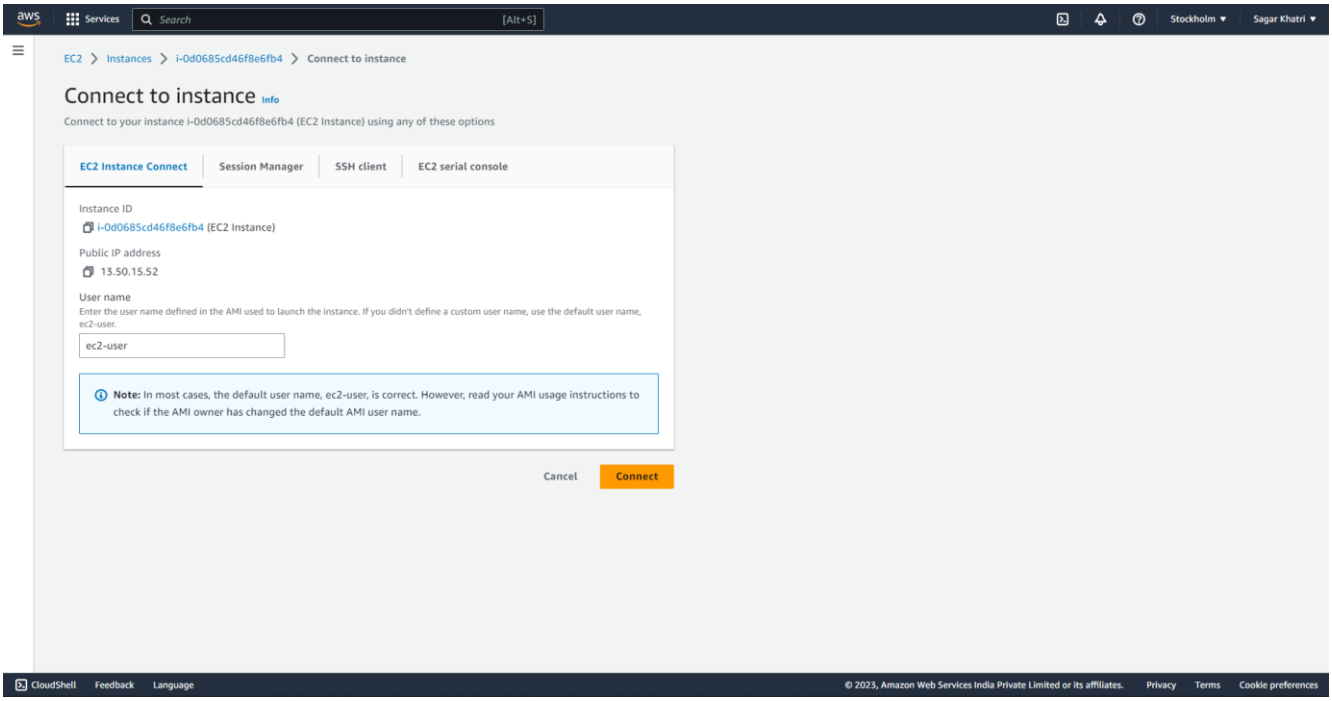


- Select an Amazon Machine Image (AMI) that matches your application requirements.
- Choose an instance type based on your application's resource needs.
- Configure the instance details, such as the subnet, security group, and any additional settings.
- Review the instance details and click on "Launch" to start the instance.



b) Connect to the Instance:

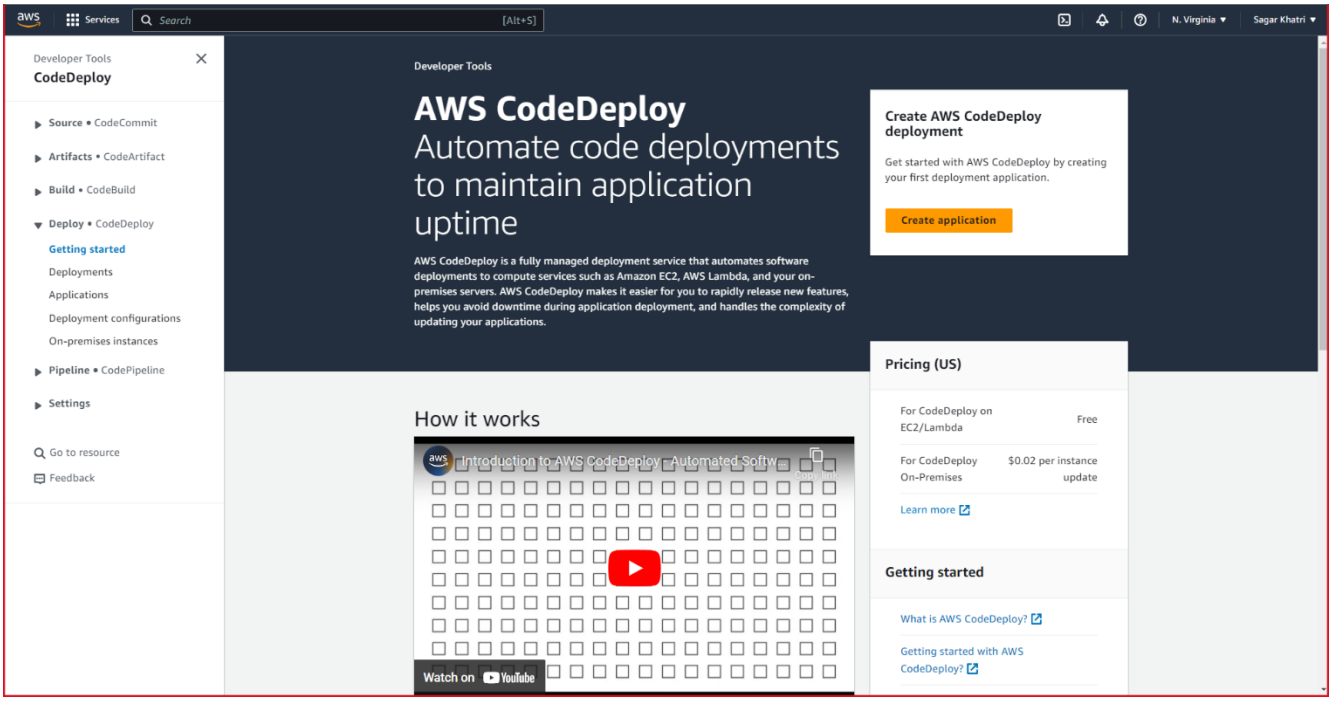
- Once the instance is running, select the instance from the EC2 dashboard.
- Click on the "Connect" button and follow the instructions to connect to the instance using SSH or Remote Desktop, depending on the operating system.
- Use the appropriate credentials to log in to the instance.



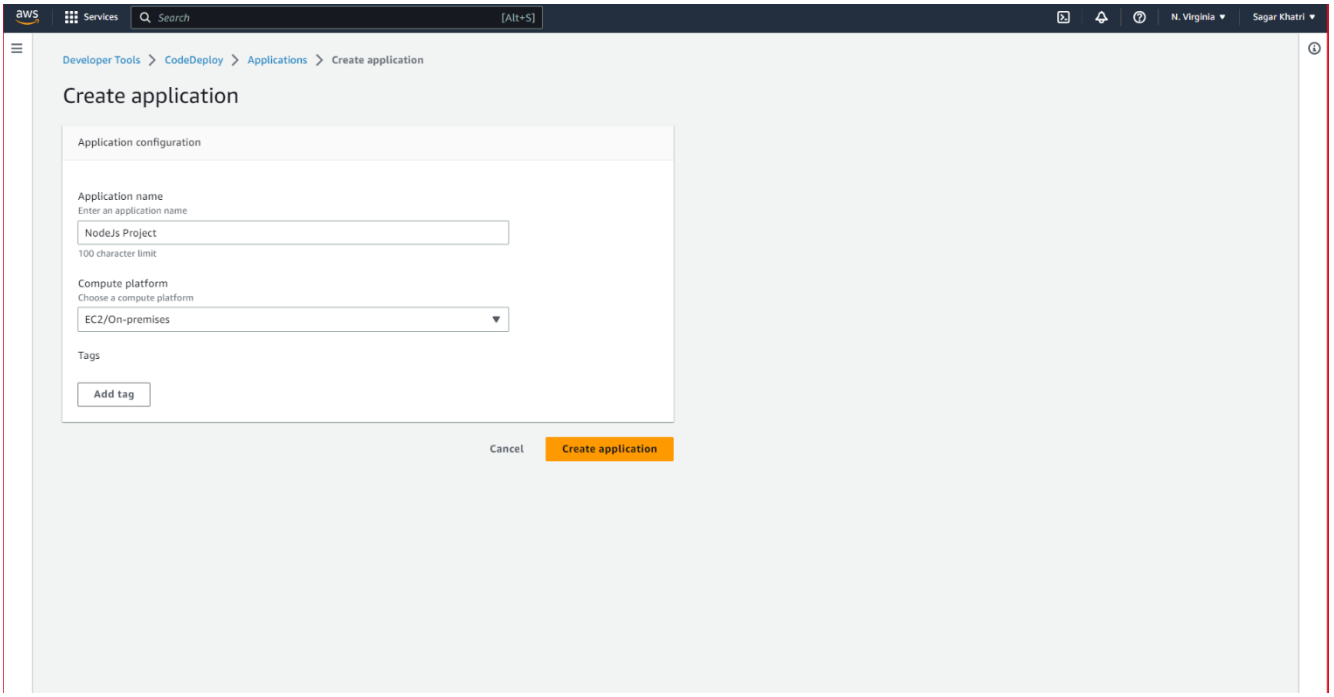
b) Deploy the Application Code:

- Transfer the application code to the instance using secure file transfer methods like SCP or SFTP.
- Install any necessary dependencies or libraries required by your application.
- Configure the application environment and any relevant settings.

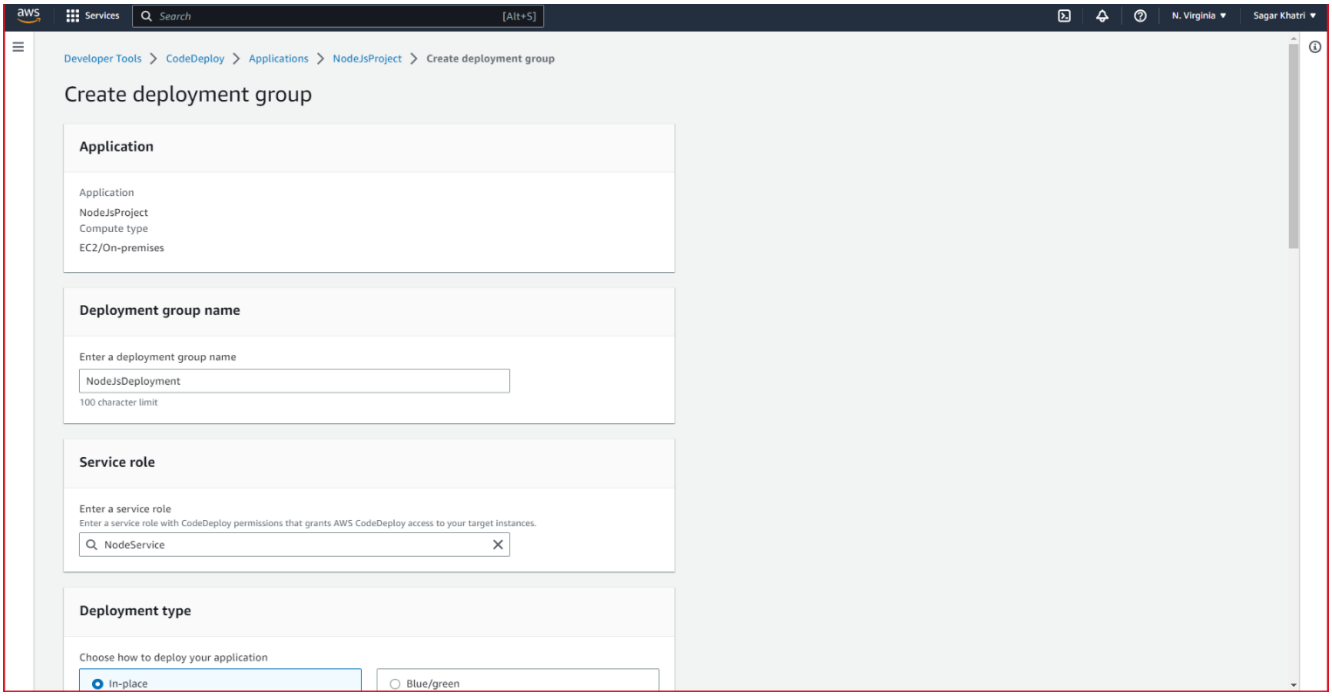
- Start the application or web server to make your application accessible.



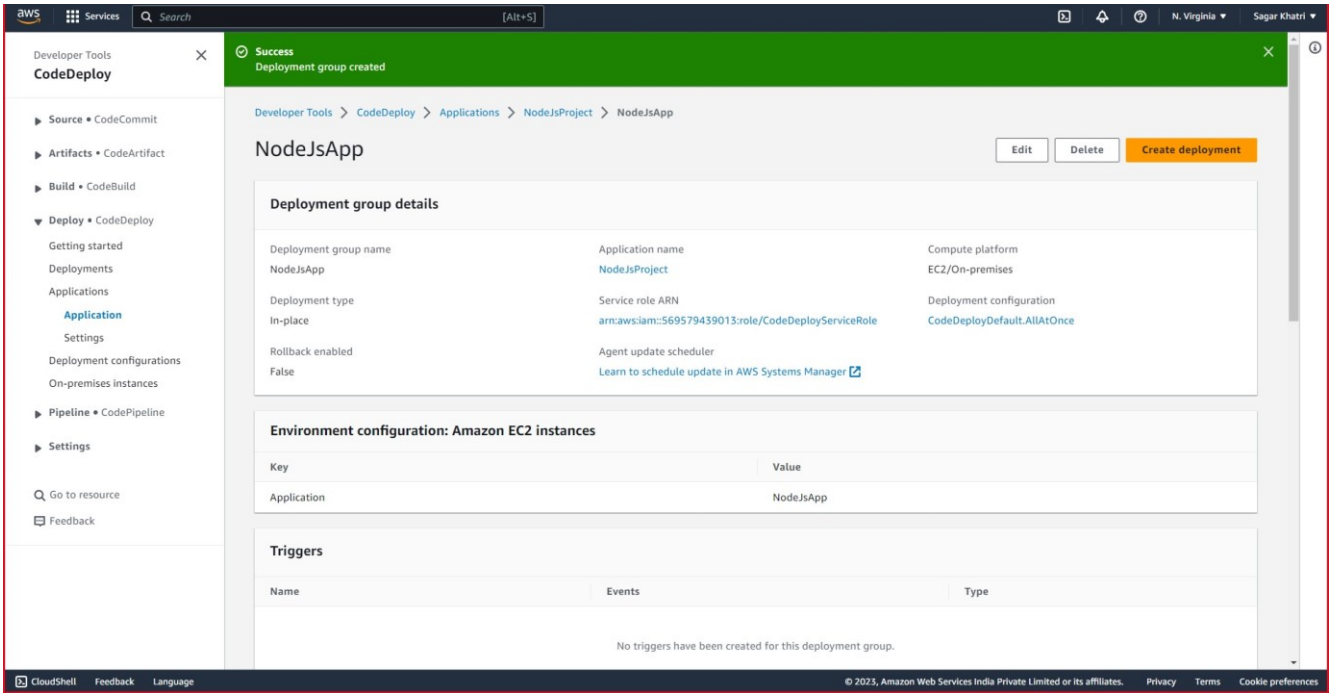
- Create the application by clicking on “Create Application”.



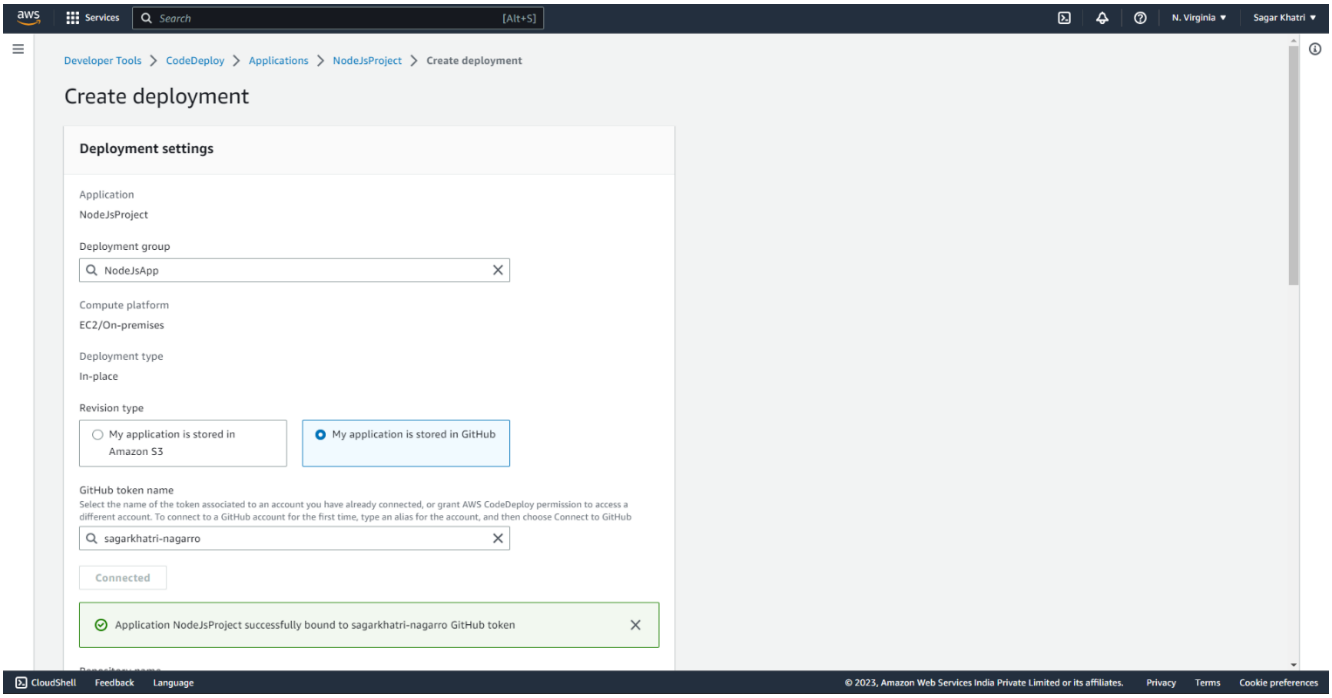
- After creating the application, we next have to create a deployment group for initiating our deployments.



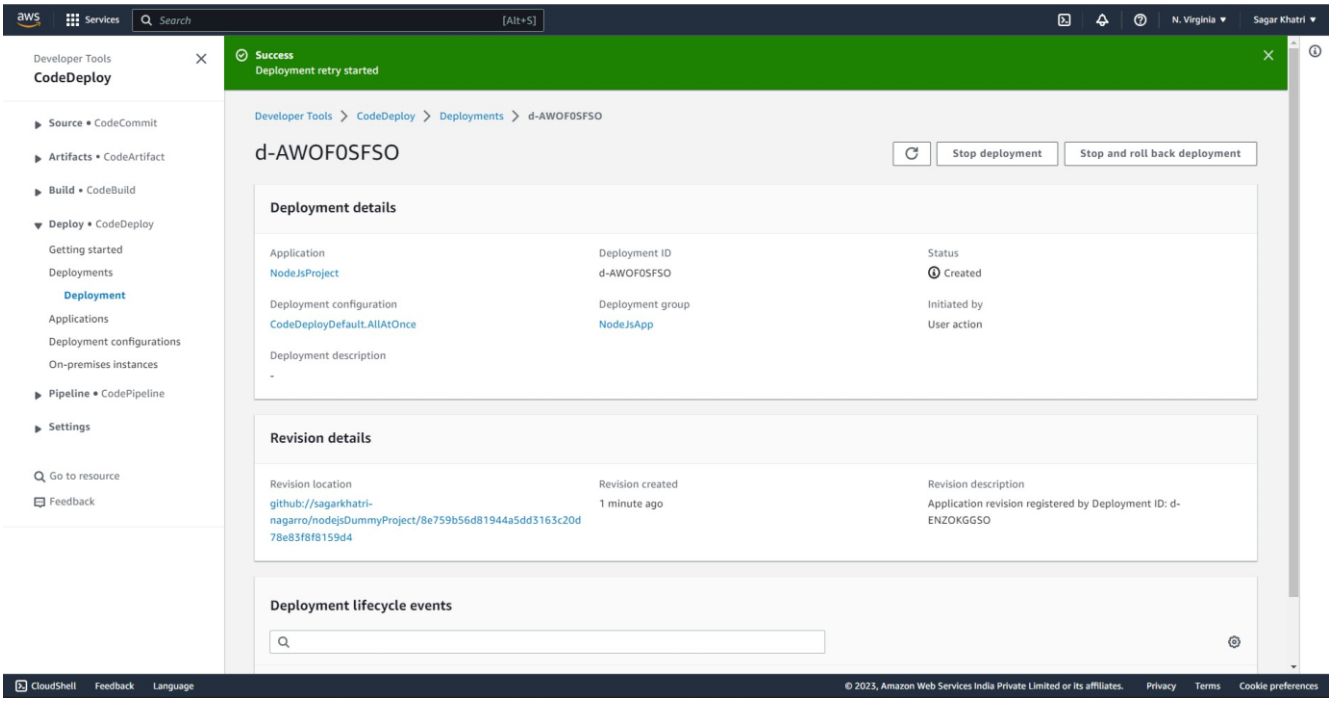
- Once the deployment group is created, we can now see it visible on the screen where we can depoloy our app into it.



- To create application deployment click on “Create deployment”.



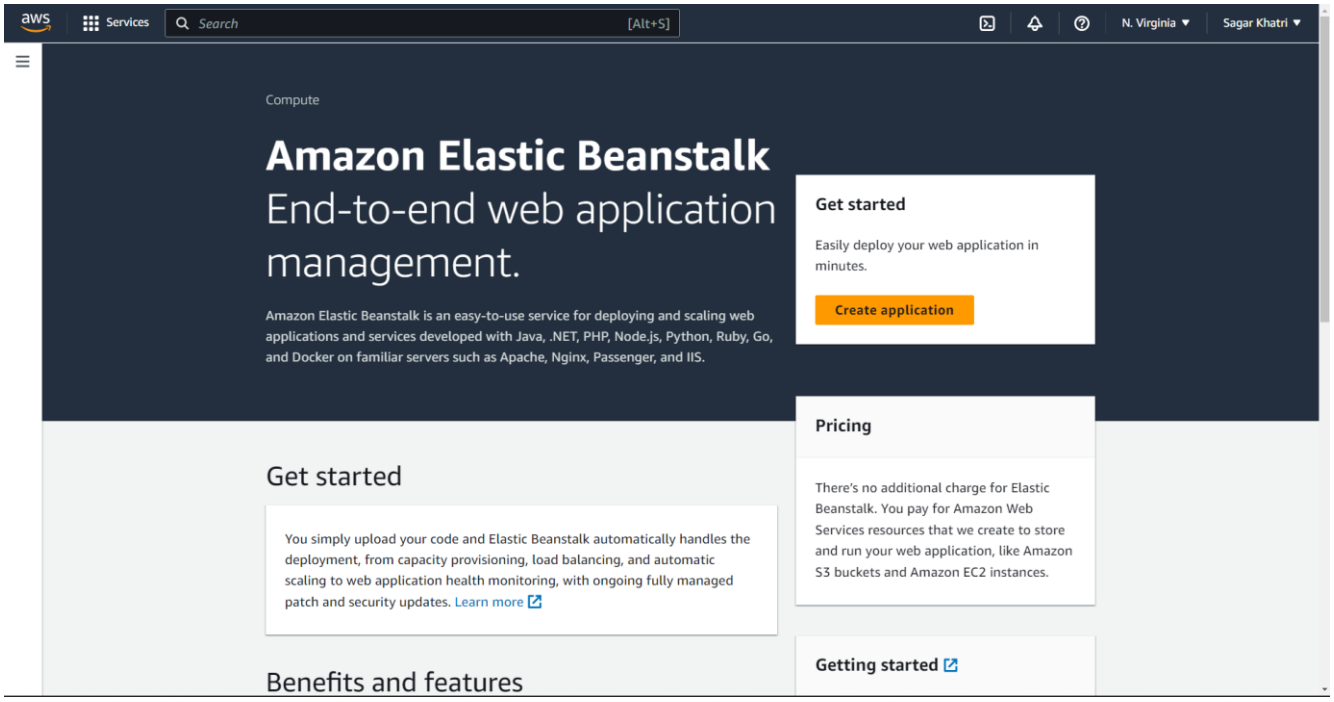
- Here, select your application stored in GitHub, where you will be asked your github token name.
- Once you are done with the filling of required information, click on “create deployment” button to finish the deployment.



- It will look like this once the deployment is complete.

Deploy the same application to Elastic beanstalk Service:

a) Open the AWS Management Console and navigate to the Elastic Beanstalk service.

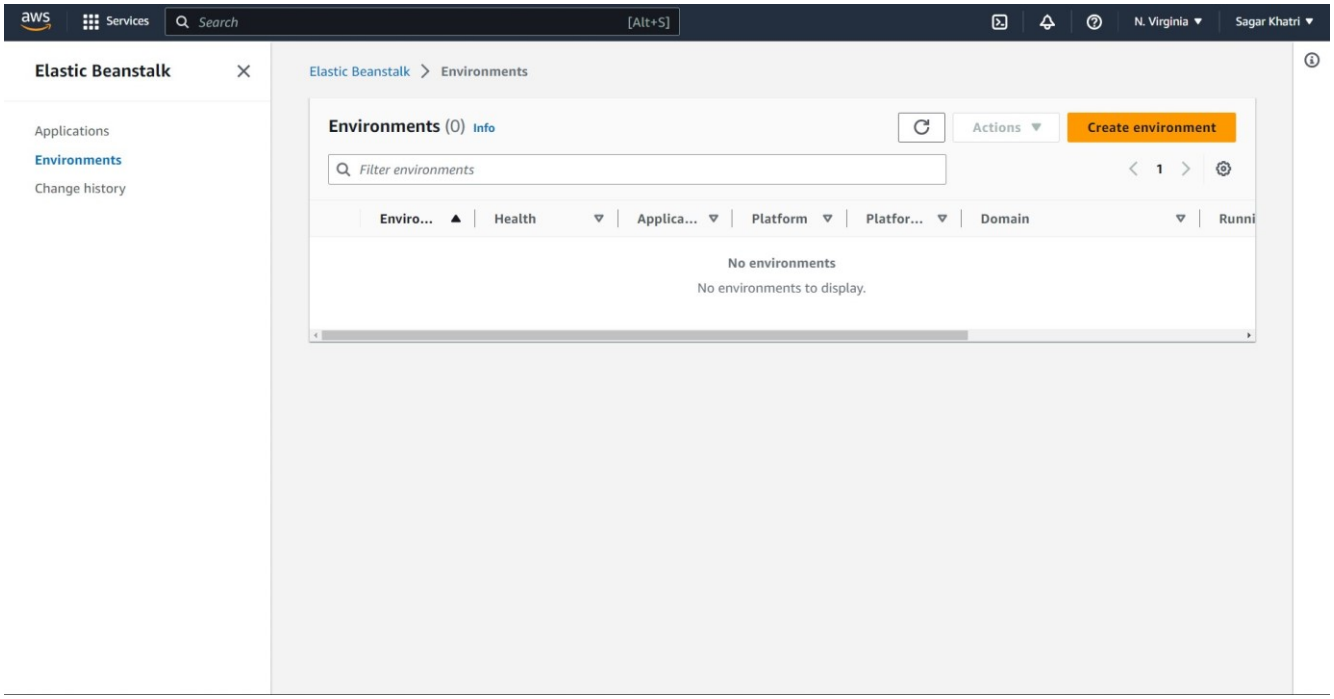


b) Click on "Create Application" and provide the necessary details (application name, platform, etc.).

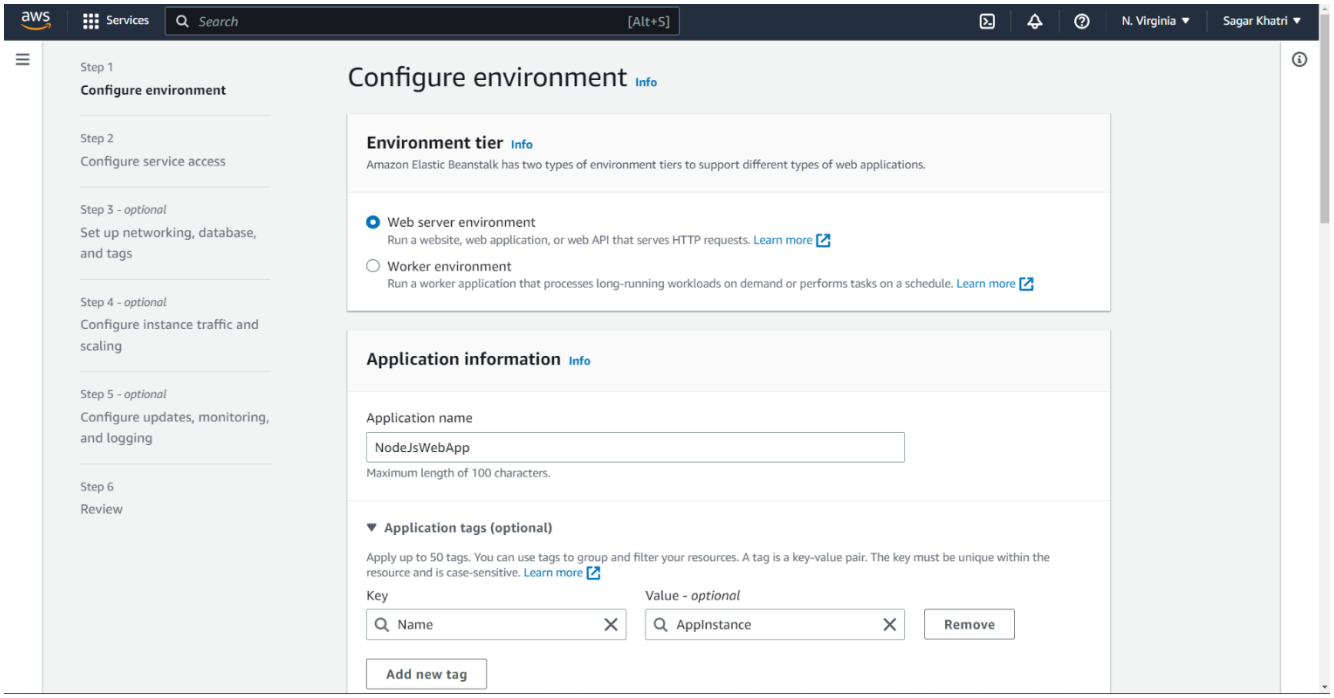
c) Configure the application settings, such as environment type, instance type, and scaling options.

d) Upload your application code or connect to a version control system.

e) From the left panel click environments.

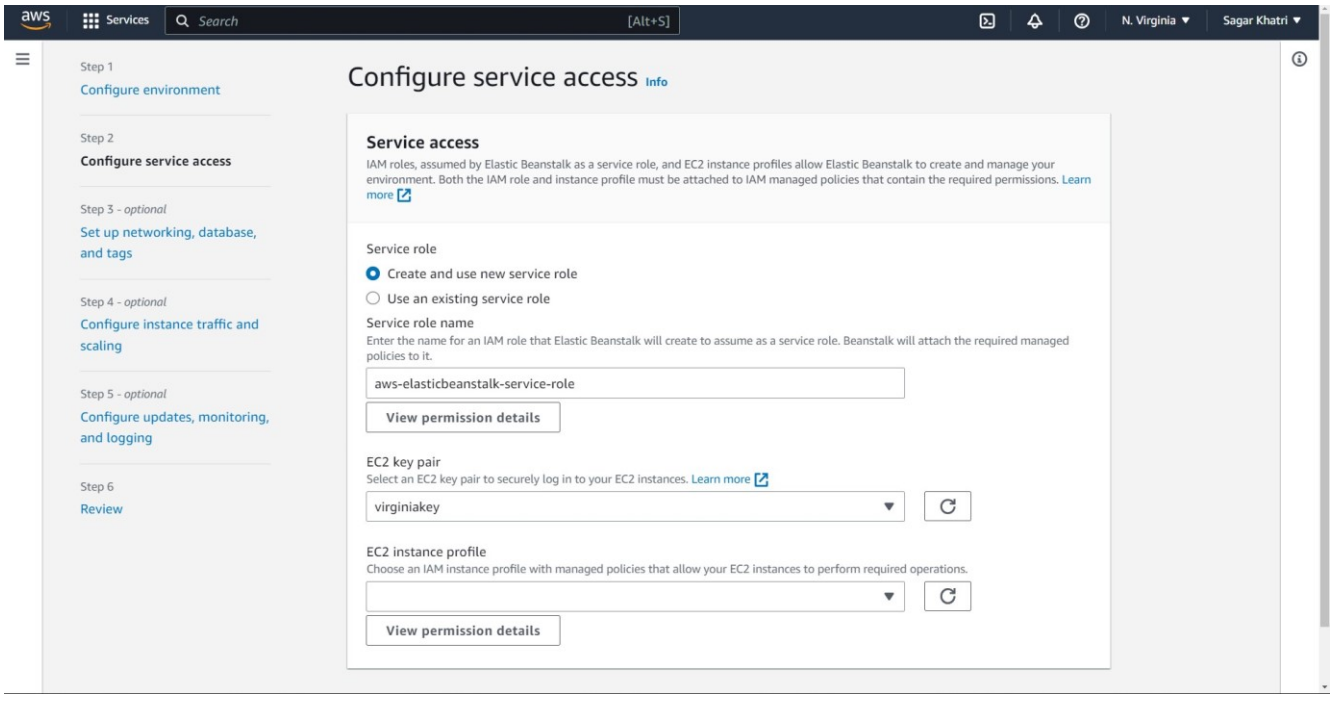


f) Click on create environment button to create a new beanstalk environment.

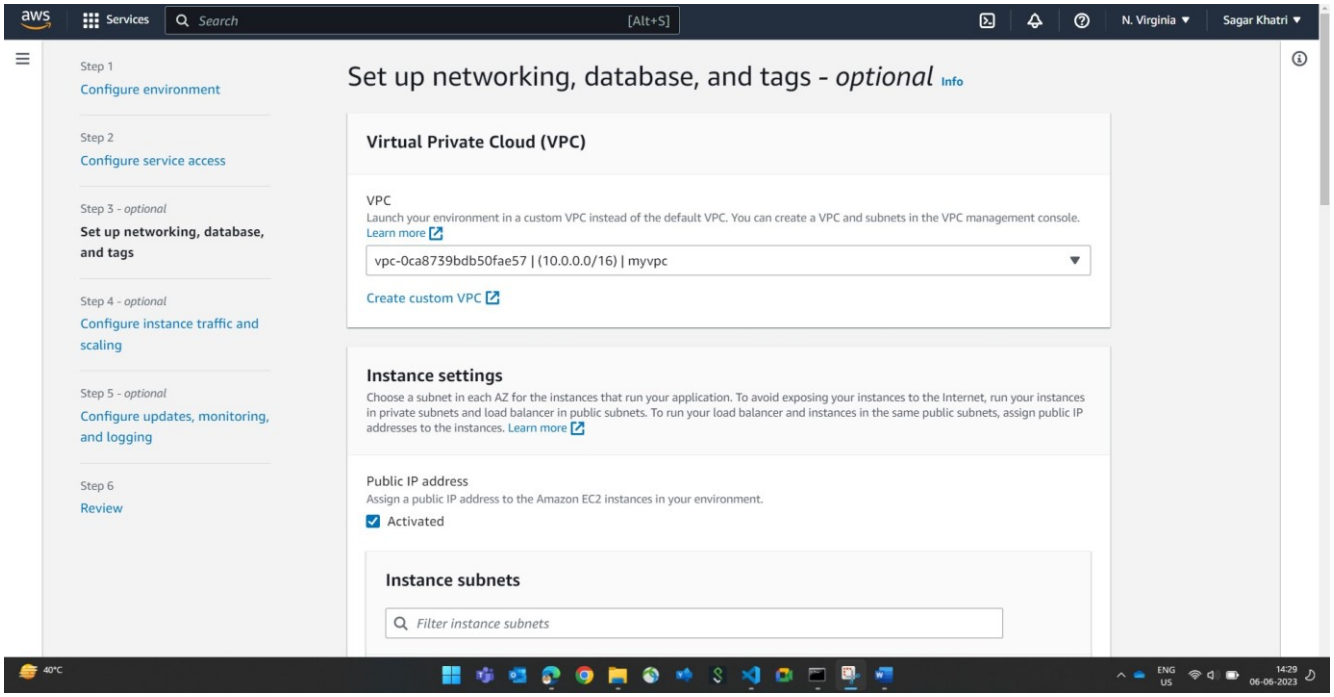


Add your name and respective keys like Name and their values. Fill in the values to configure your environment.

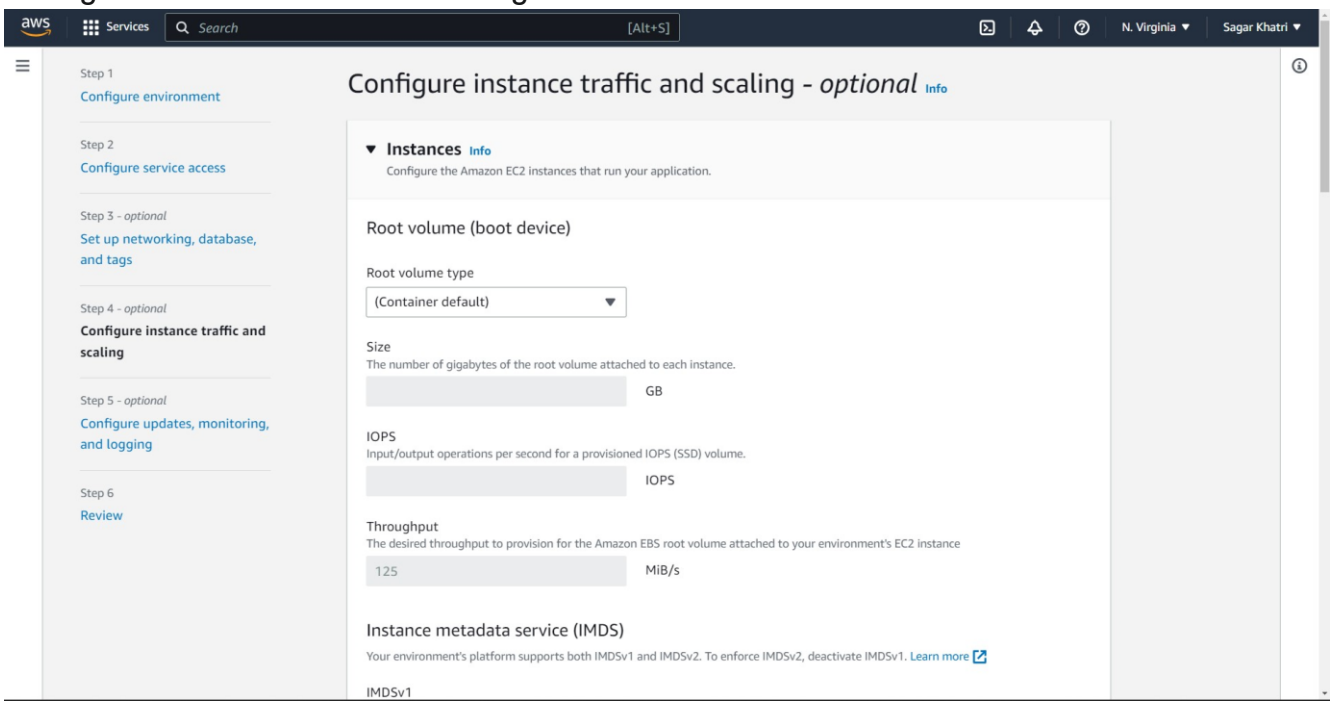
g) Now proceed to second step of this process and configure the service access by defineing the existing service role or creating a new one. Also Choose an EC2 pair key created before.



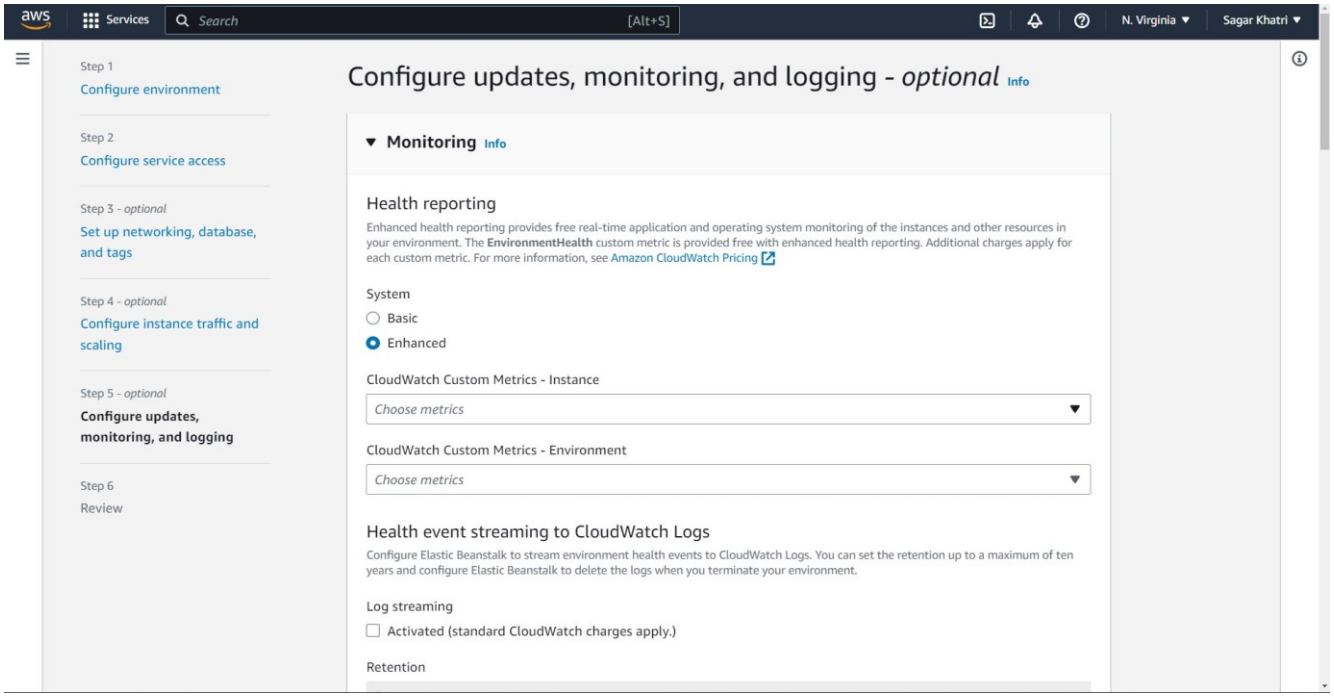
h) Now setup networking Database and tags.



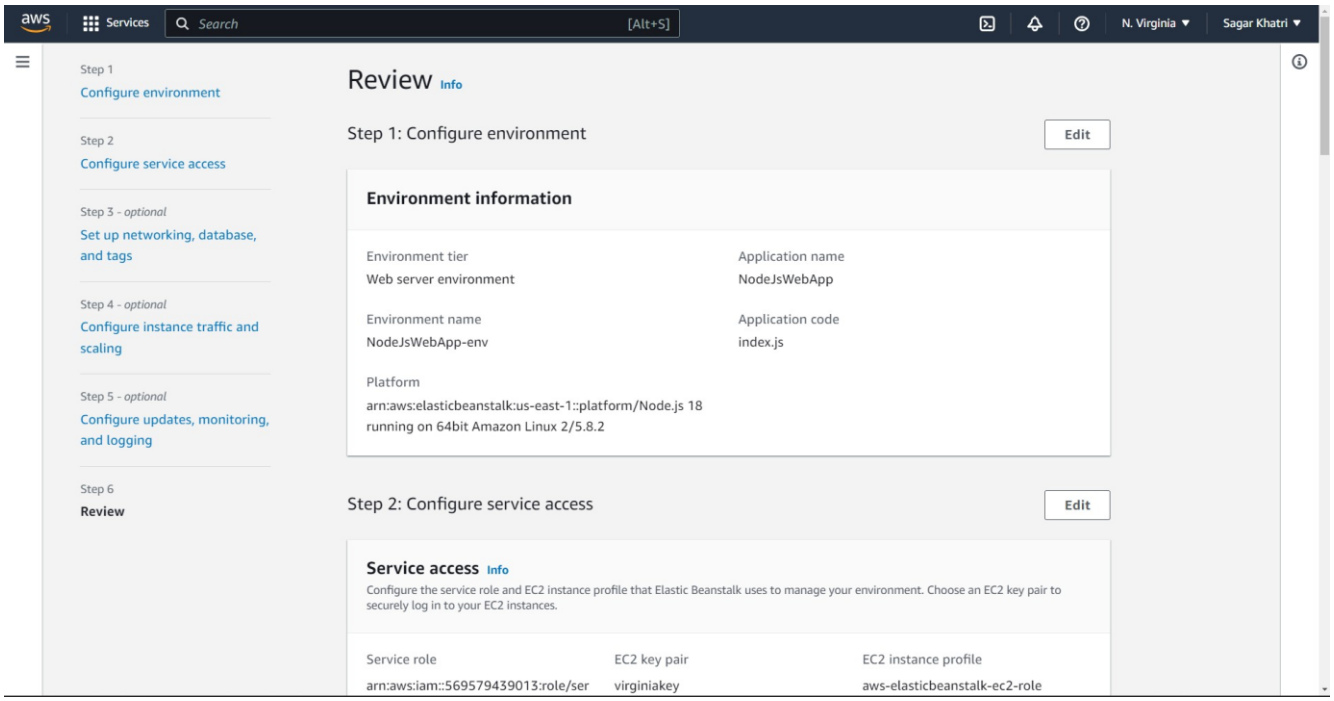
i) Configure instance traffic and scaling



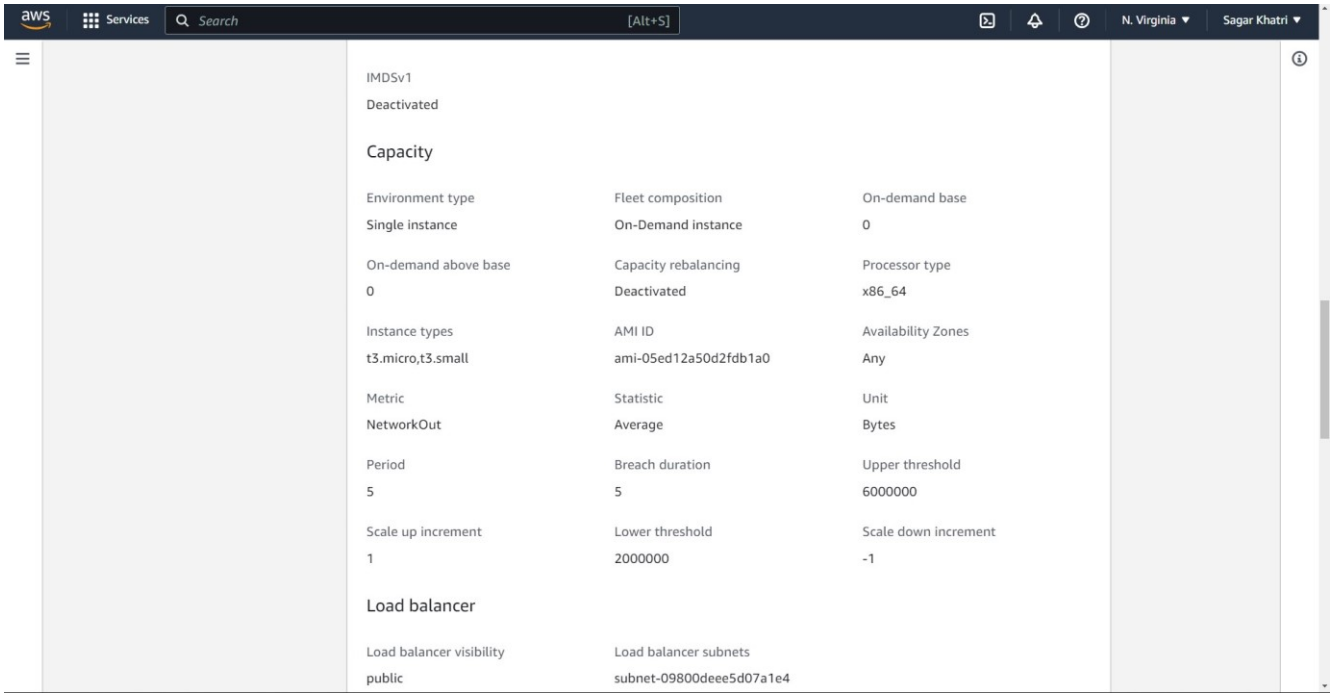
j) Setup updates ,monitoring and logging



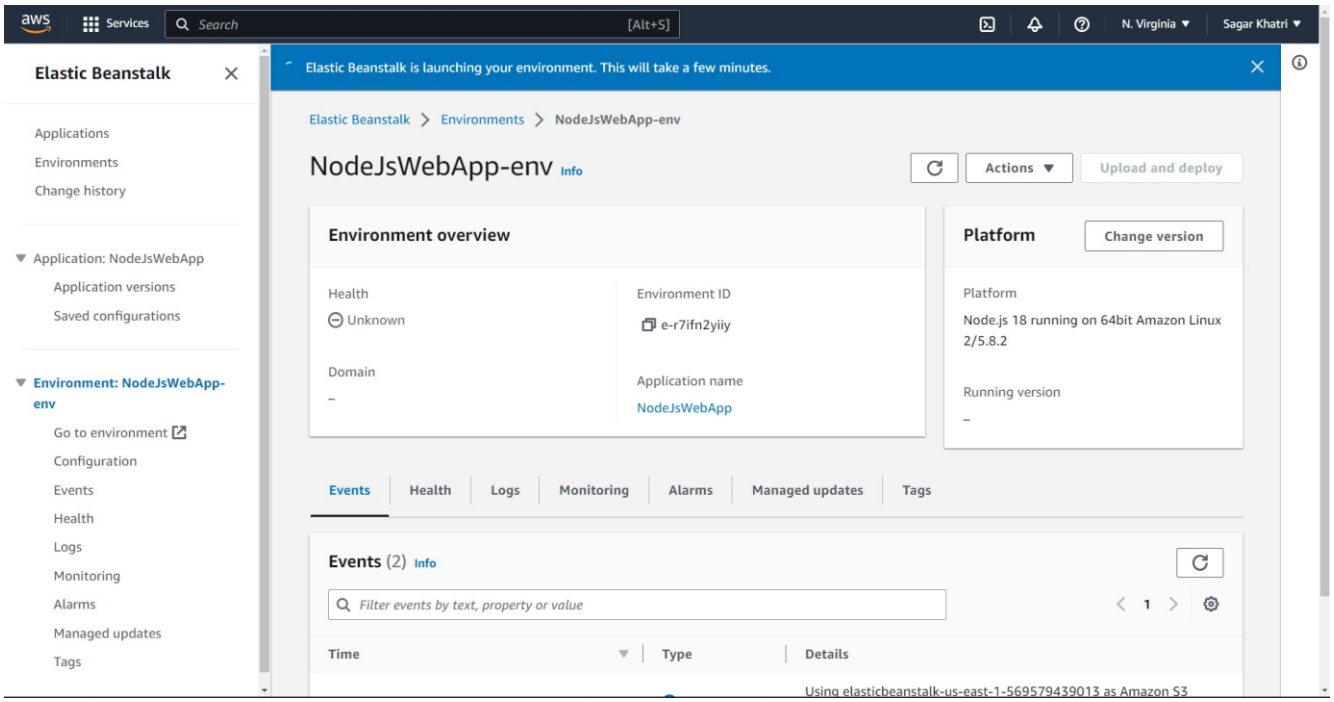
k) Now click on the review button to review your selected settings before creating the environment.



l) Review you capacity and load balancer metrics to avoid additional cost.



m) Now click Submit to create the new environment.

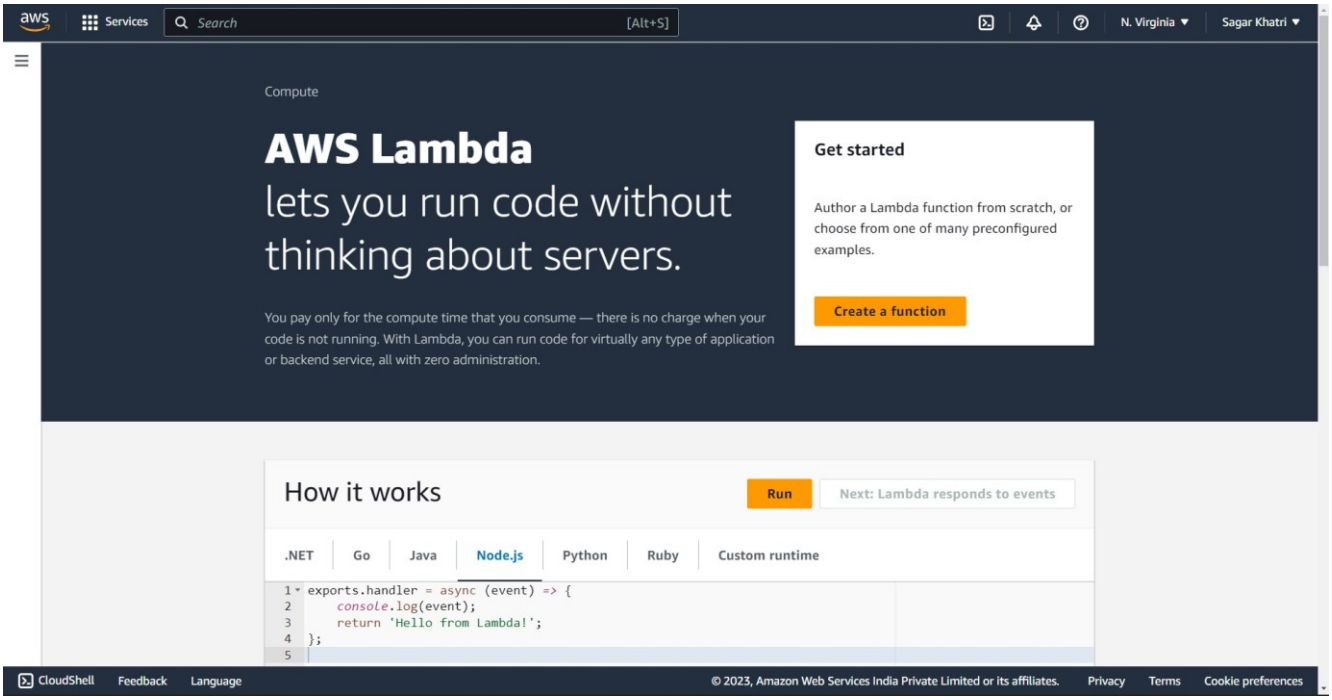


you can now deploy your application in the created beanstalk environment now.

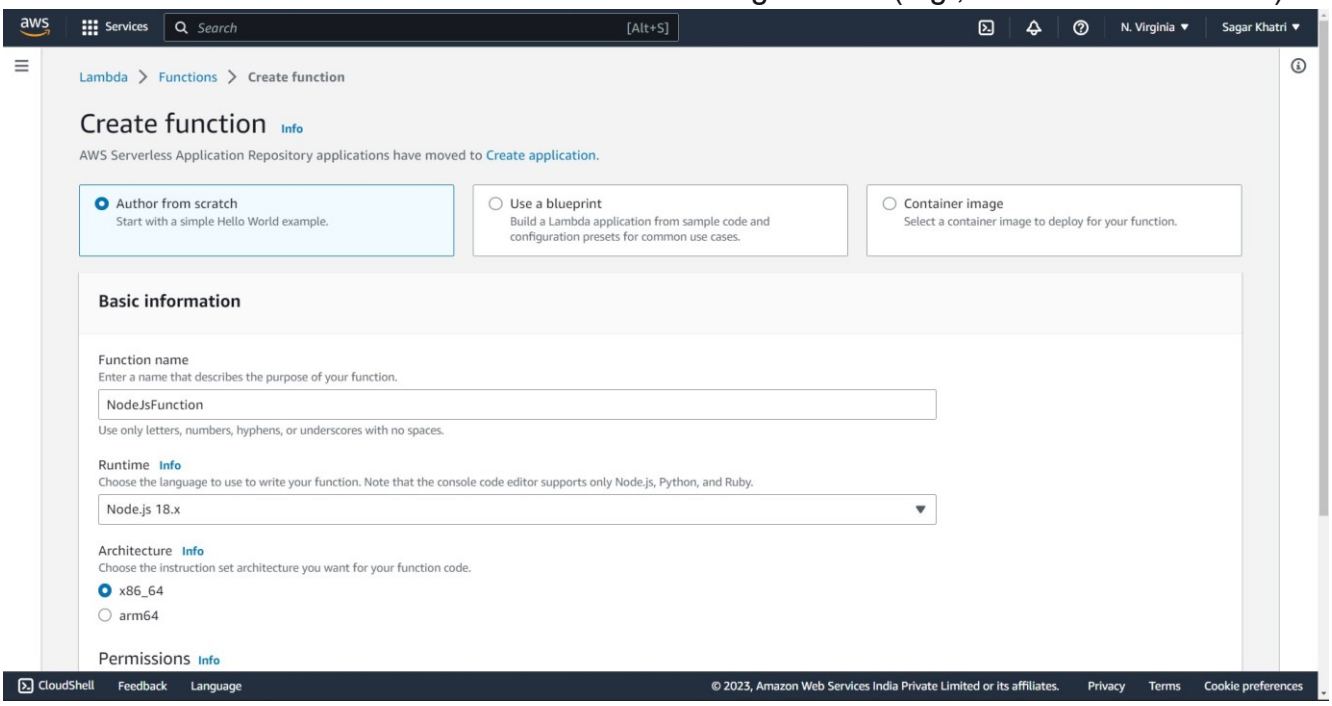
Once the deployment is successful, you can access your application using the provided URL or the Elastic Beanstalk environment URL.
Test your application to ensure it is functioning as expected.

Creating a Lambda Function Triggered by S3 Bucket Upload:

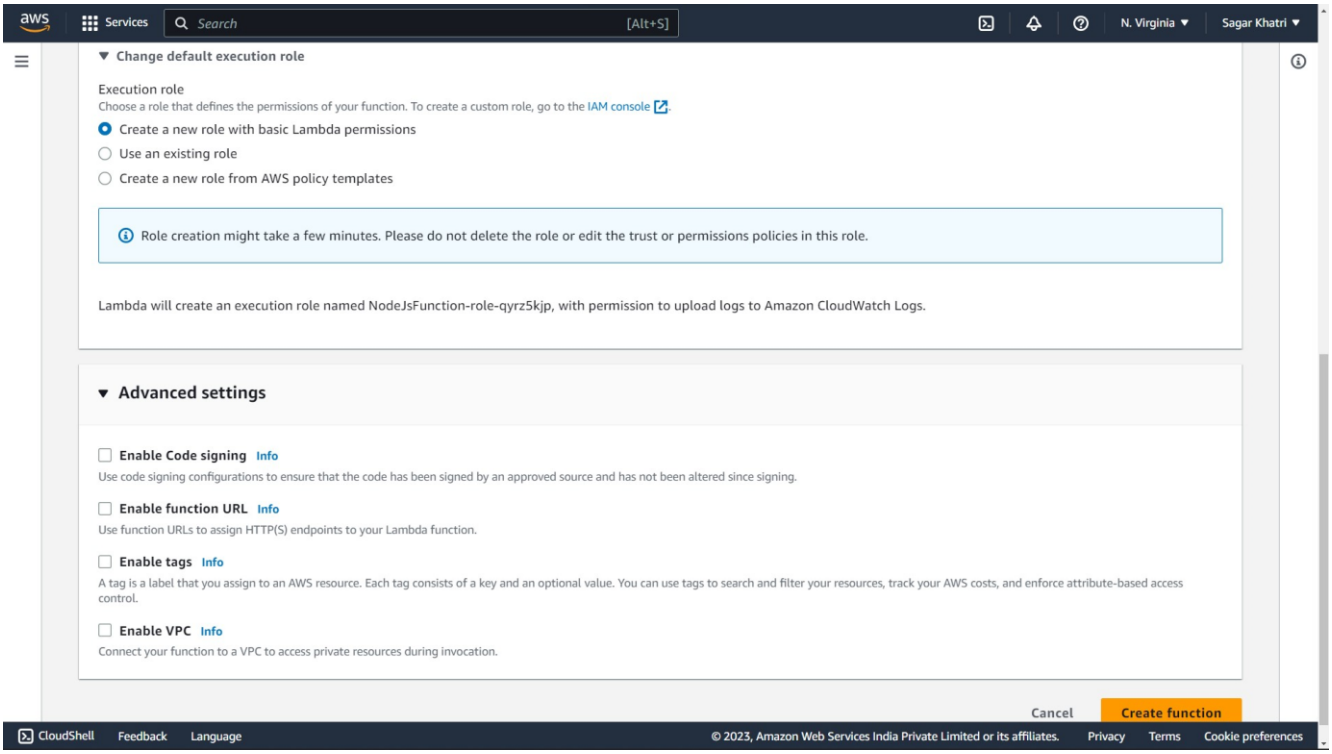
a. Open the AWS Management Console and navigate to the Lambda service.



b. Click on "Create function" and choose the authoring method (e.g., "Author from scratch").

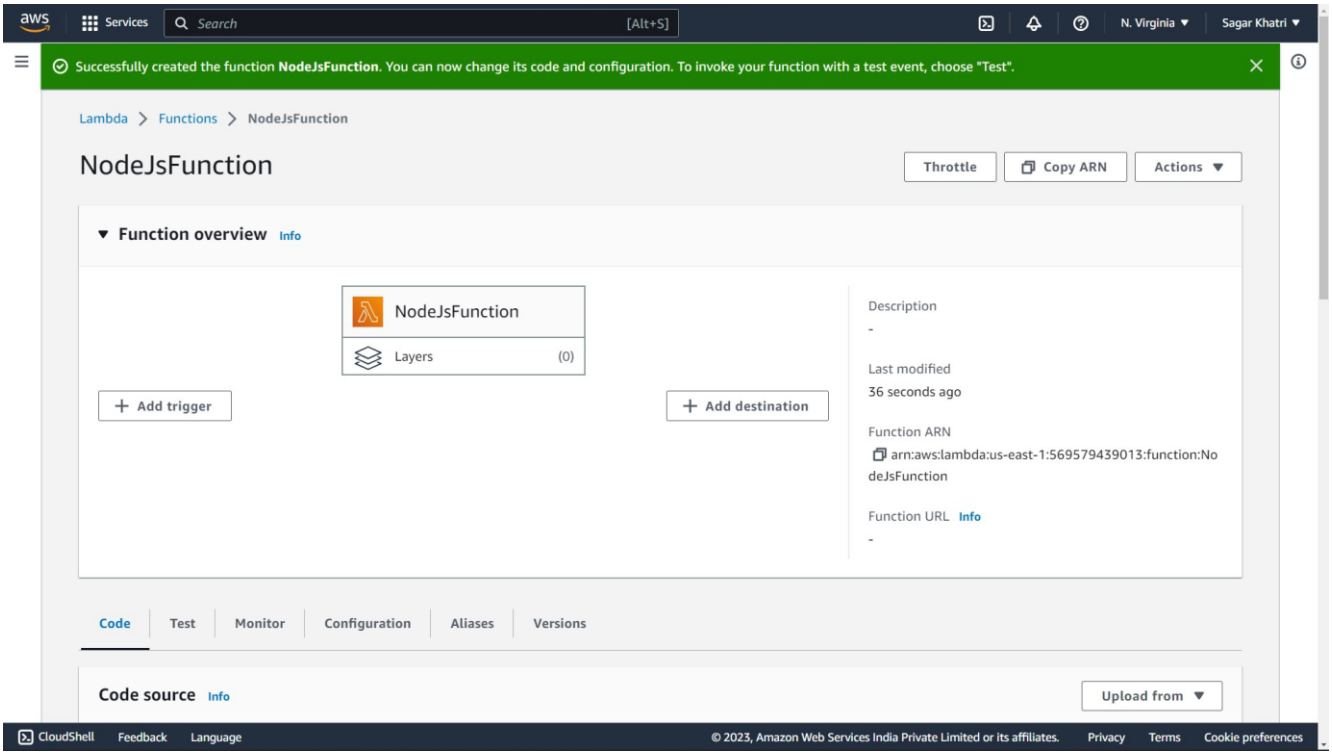


c. Provide the necessary details for the function (function name, runtime, etc.).

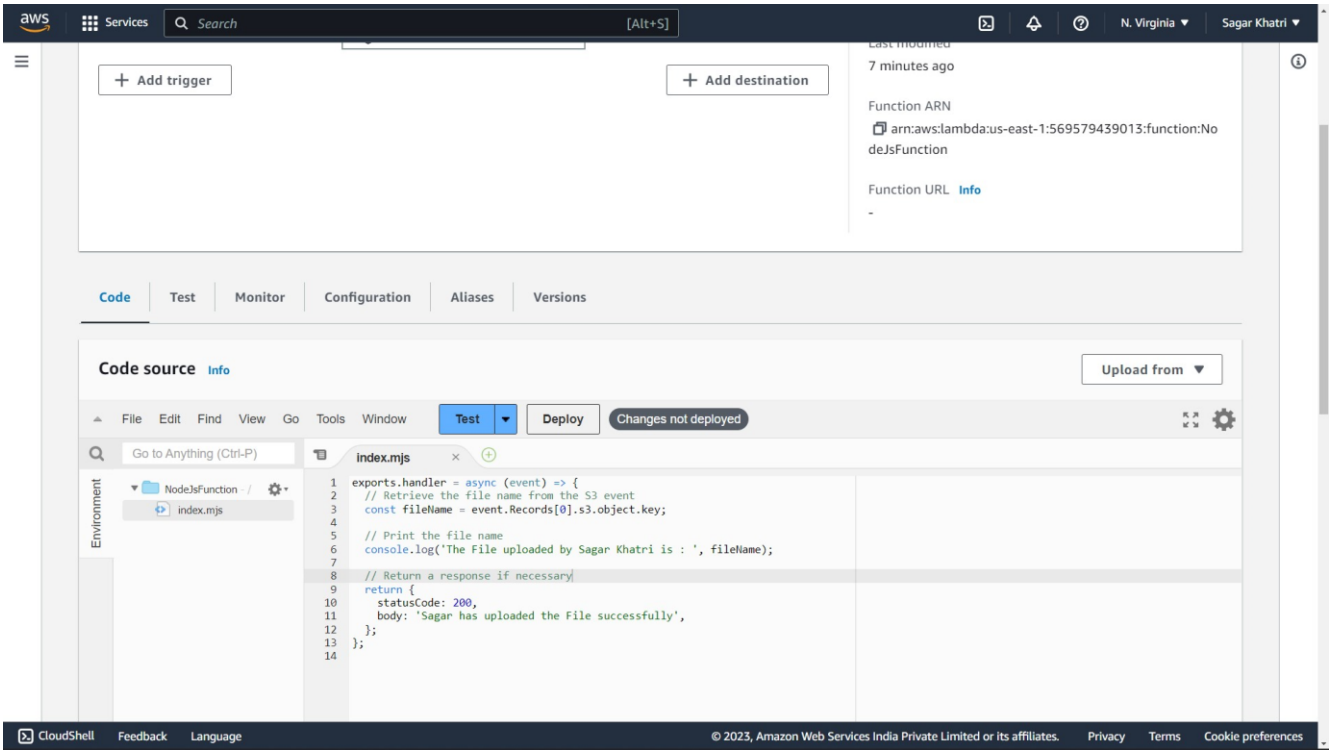


d. Under "Permissions," choose an existing or create a new execution role with the necessary S3 permissions.

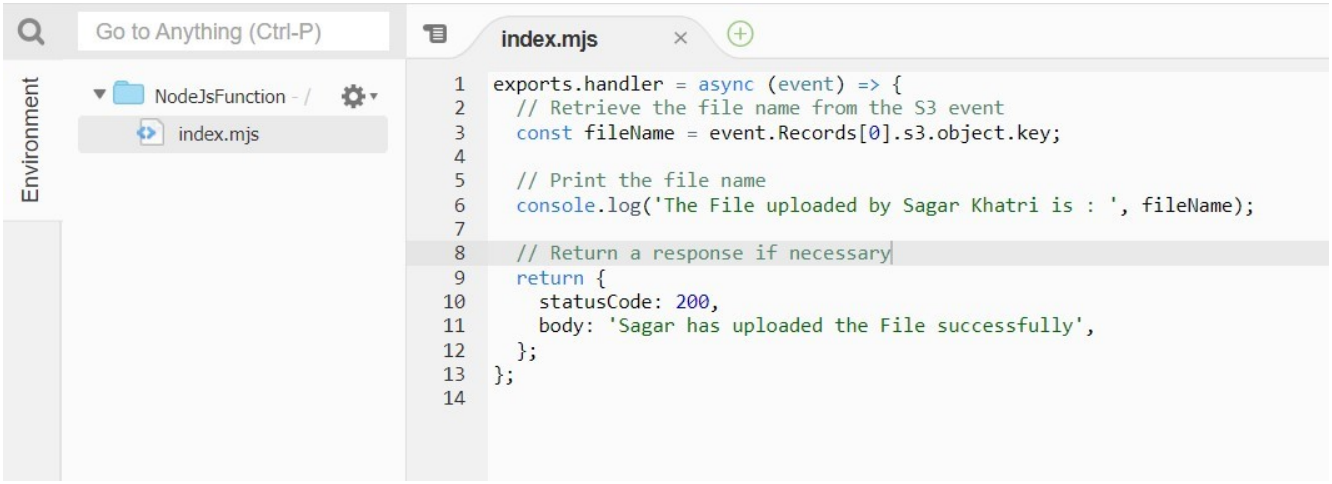
e. Click on the "Create function" button to create the Lambda function.



f. Scroll to the code source in the lambda function created.

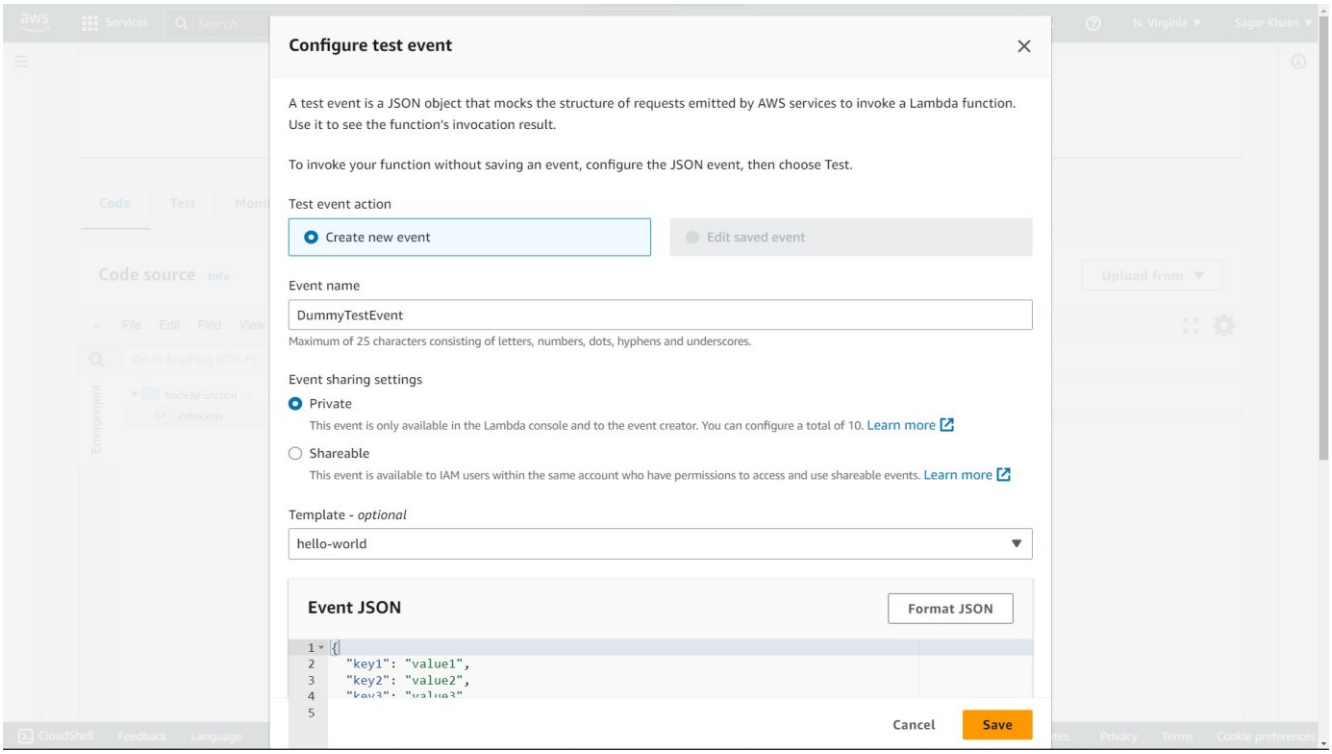


g. In the function's code editor, write the code to handle the S3 trigger event and print the uploaded file's name.

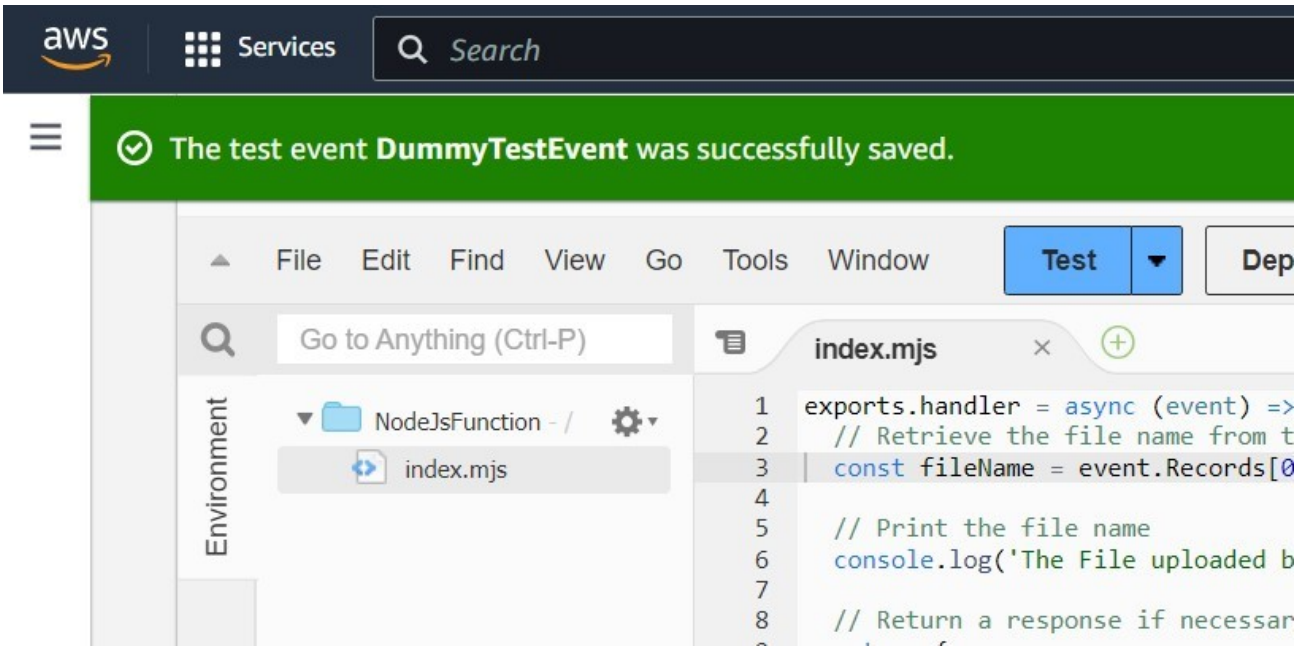


h. Save the Lambda function.

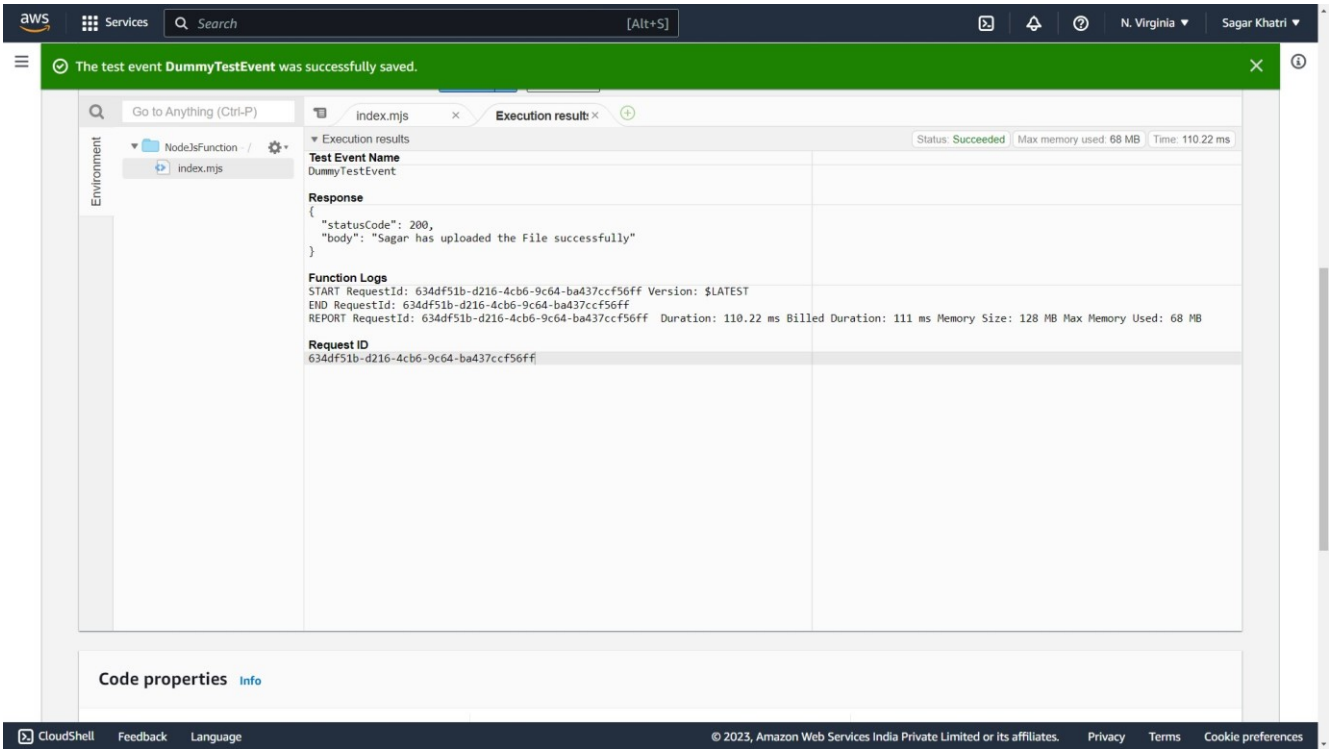
I. Click on “Test” and then cclick on confiure to open the dialog box



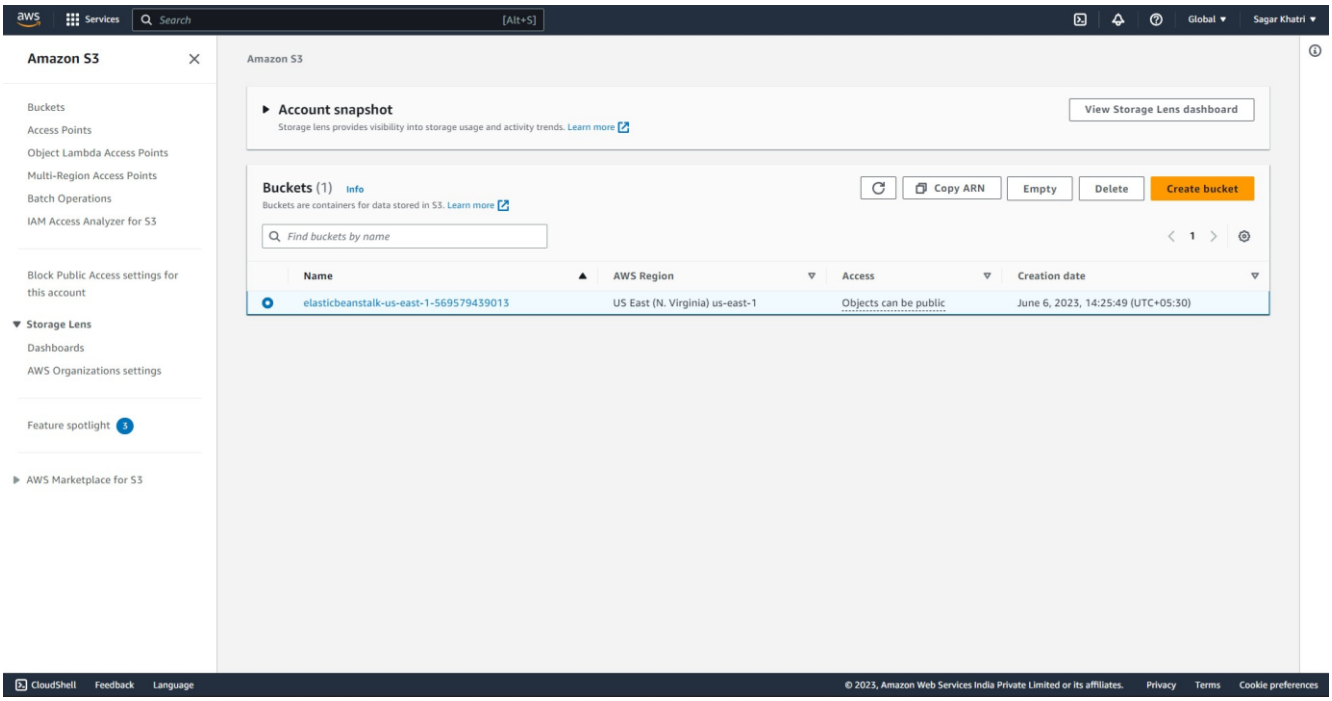
j. Fill the details of the test event to test the code which is already written above and provided the required json data and templates, if any. Click Save.



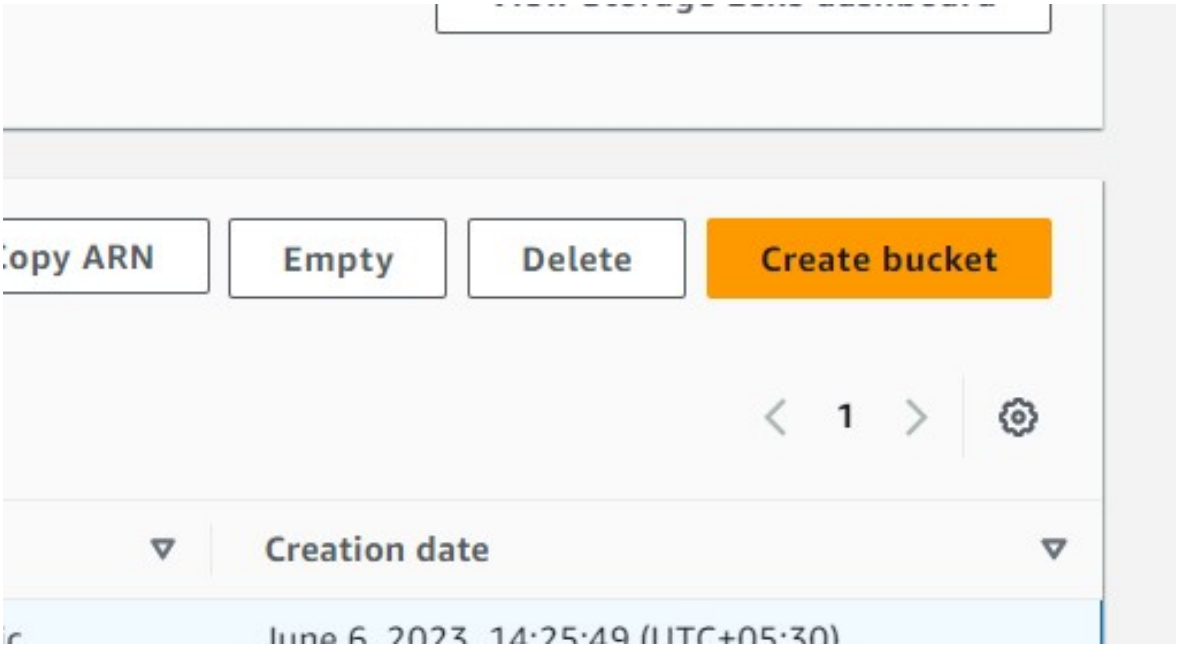
Now run the test event to check whether the code is working or not. Then click Deploy to deploy the function.



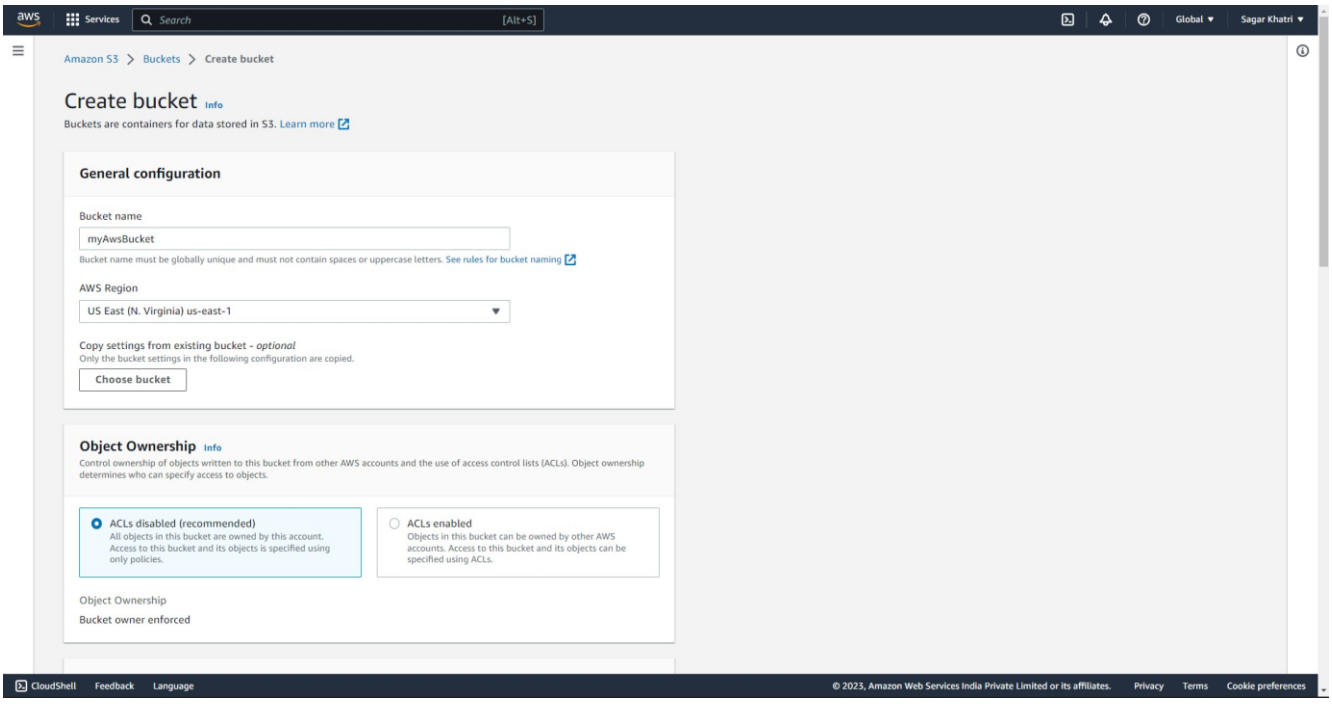
Now move to the search bar to search for S3 bucket. Click on S3 in the search suggestions.



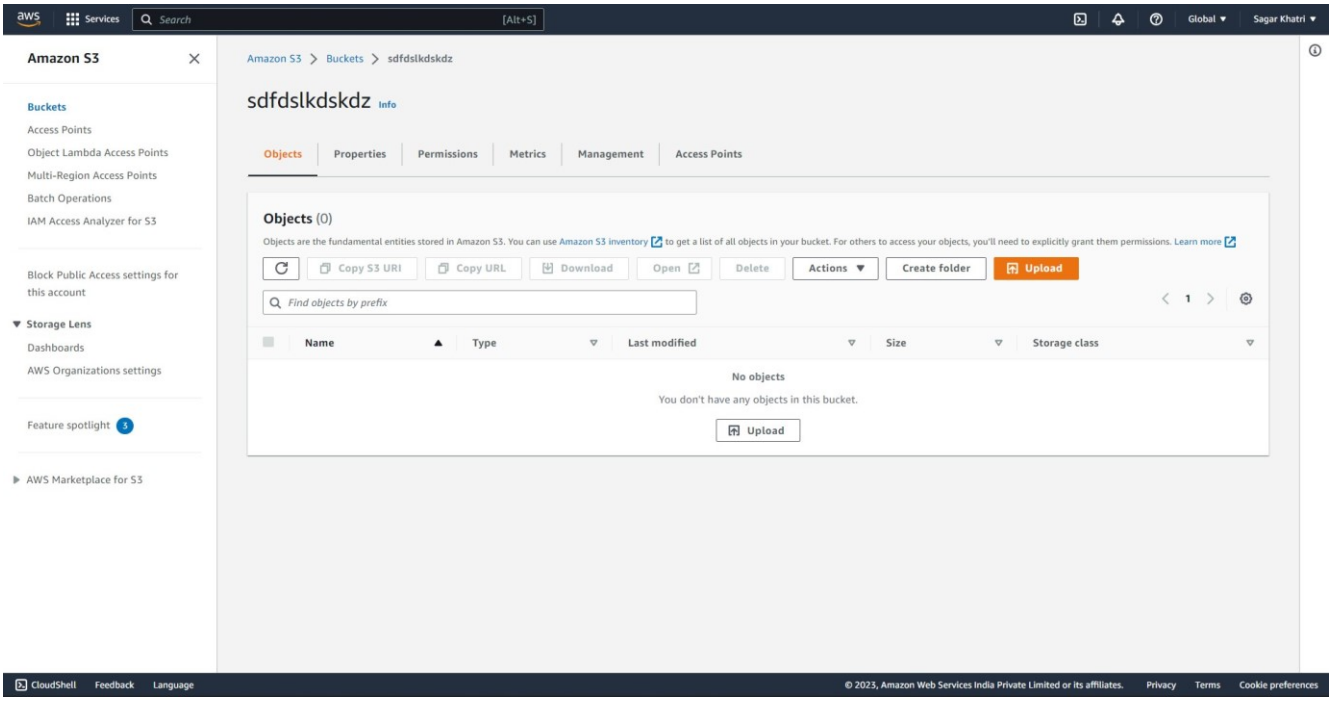
Click on the “Create bucket” button to start creatin a new bucket.



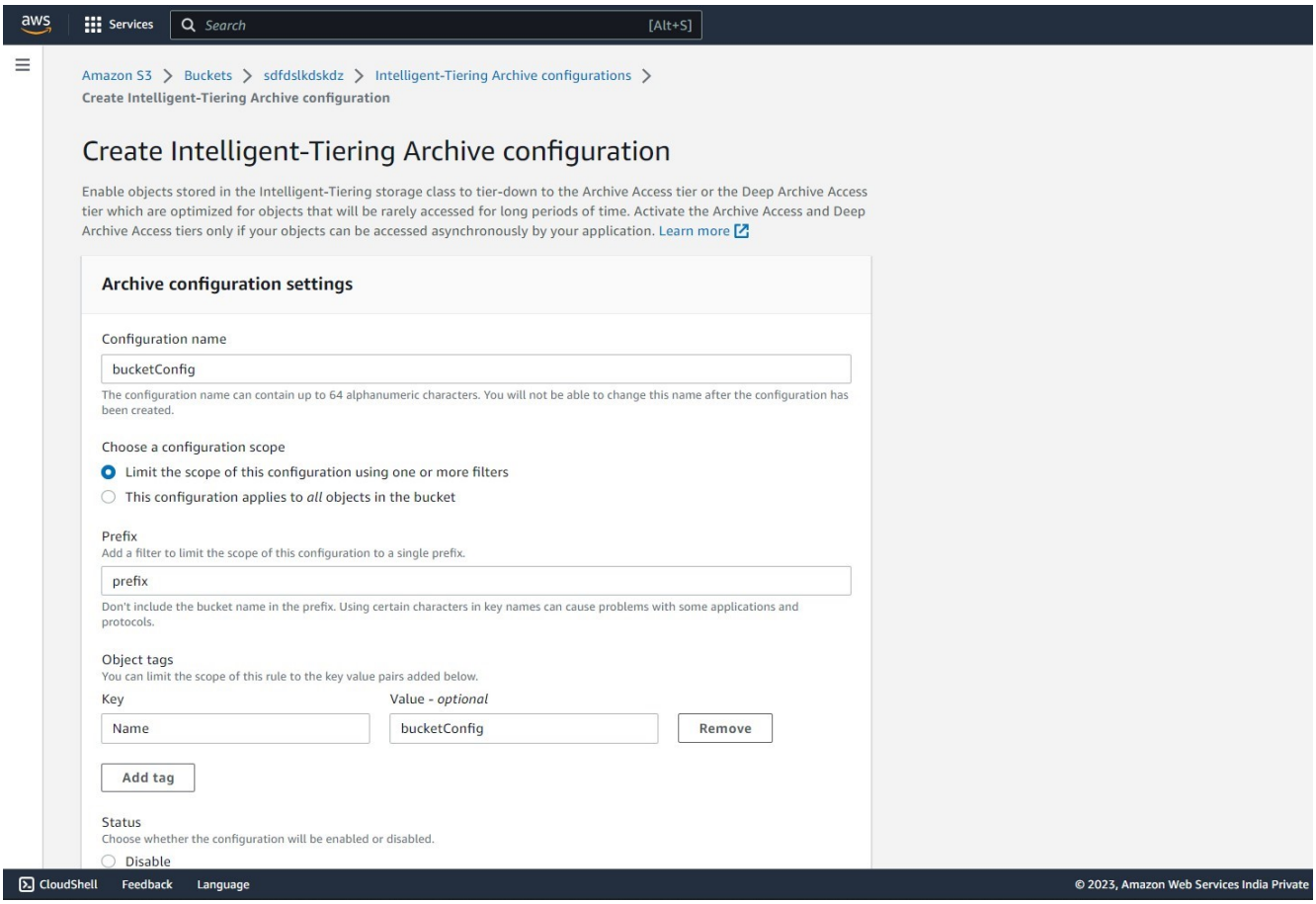
Enter the details of the S3 bucket you want to create, like it’s name and region. Also sleect the ownership of objects written in this bucket.



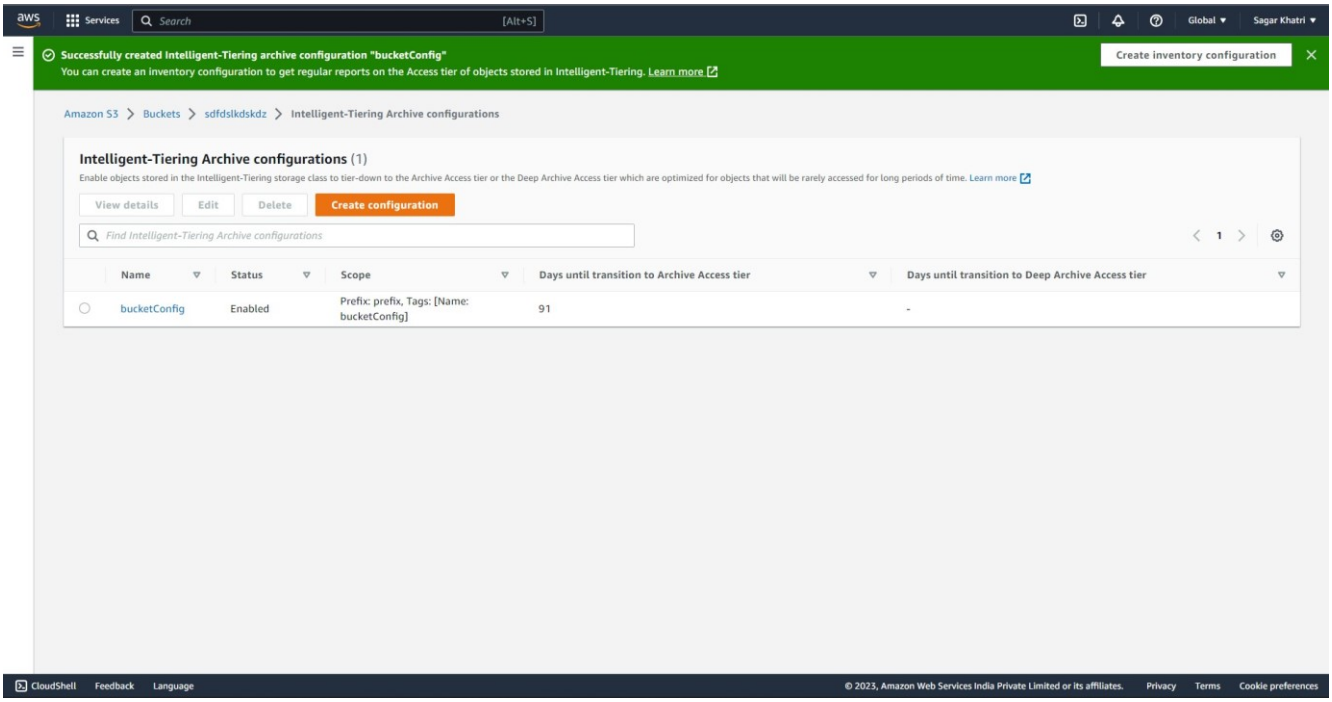
click Finis to finish the creation of bucket. You will be able to see the newly created bucket in the dashboard.



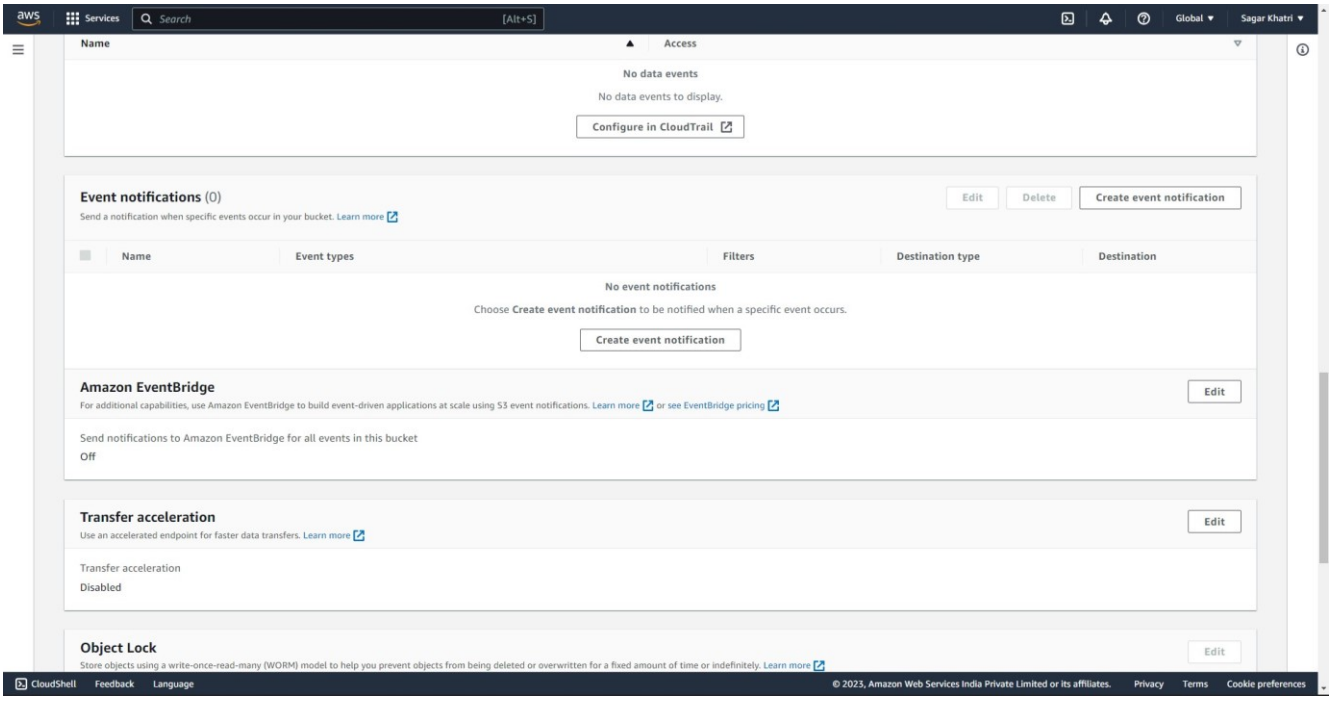
Now we should create an intelligent tiering archive config as bucketConfig.



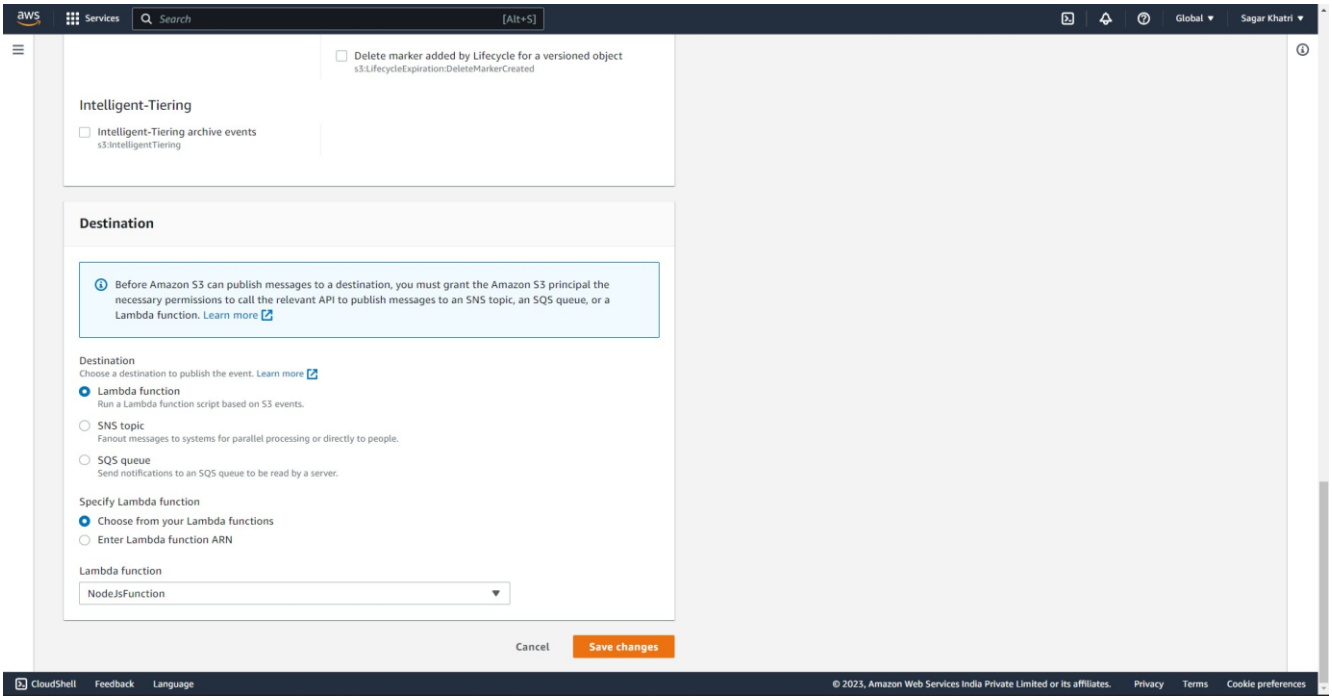
fill the details to create and view the bucket config



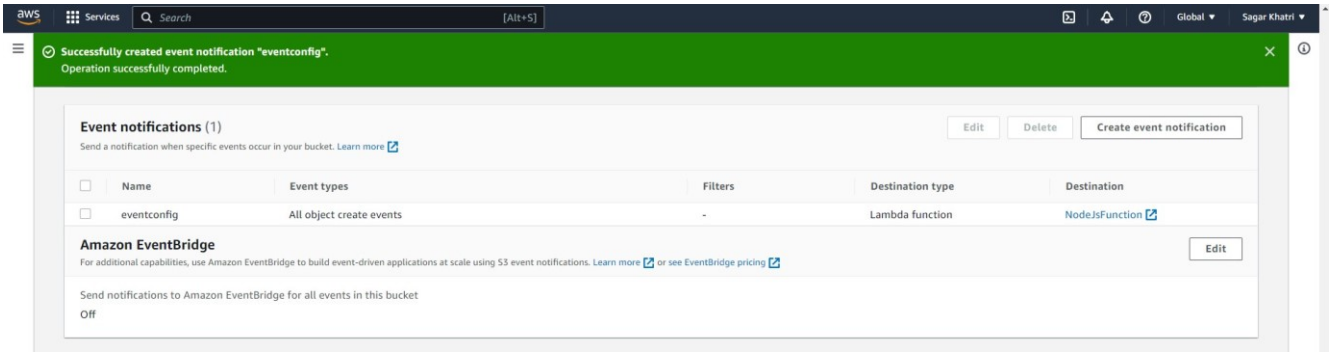
In the bucket configuration, create an event config by scrolling down to event config



Click create a new config button to start.



choose the information like destination and the trigger like “put” triggers to trigger the event config. Choose Lambda function in the destination section and then click “Save changes”



On successful creation you have now created a Lambda that should trigger as soon as you upload a file in the S3 bucket.