**EXPERIMENT-5**

**NAME:** SHAIKH MUBASHIRA TUFEL AHMED

**ROLL NO:** 612055 **COURSE:** ADVANCE DEVOPS(ITL504)

**BRANCH:** T.E. INFORMATION TECHNOLOGY (SEM 5)

1. **What is AWS Lambda?**

🡺Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, you can run code for virtually any type of application or backend service. All you need to do is supply your code in one of the [languages that Lambda supports](https://docs.aws.amazon.com/lambda/latest/dg/lambda-runtimes.html).

1. **What is serverless computing?**

* **Serverless computing** is a cloud architecture that allows organizations to get on-demand access to the resources they need. Customers only pay for the resources they use. Resources are not allocated to an application when it is not in use.
* In a serverless computing architecture, a server’s code execution is fully managed by the cloud provider. Therefore, the provider’s customers do not need to develop and deploy the underlying infrastructure that would traditionally be required to run applications and programs. The primary objective of serverless computing is to make it easier for software developers to create code that is intended to run on cloud platforms and perform a clearly defined role.

1. **What languages does AWS Lambda support?**

* C#
* Go
* Java
* Node.js
* PowerShell
* Python
* Ruby

1. **What is AWS DynamoDB Table**

🡺Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling. DynamoDB also offers encryption at rest, which eliminates the operational burden and complexity involved in protecting sensitive data.

1. **Explain AWS IAM service**

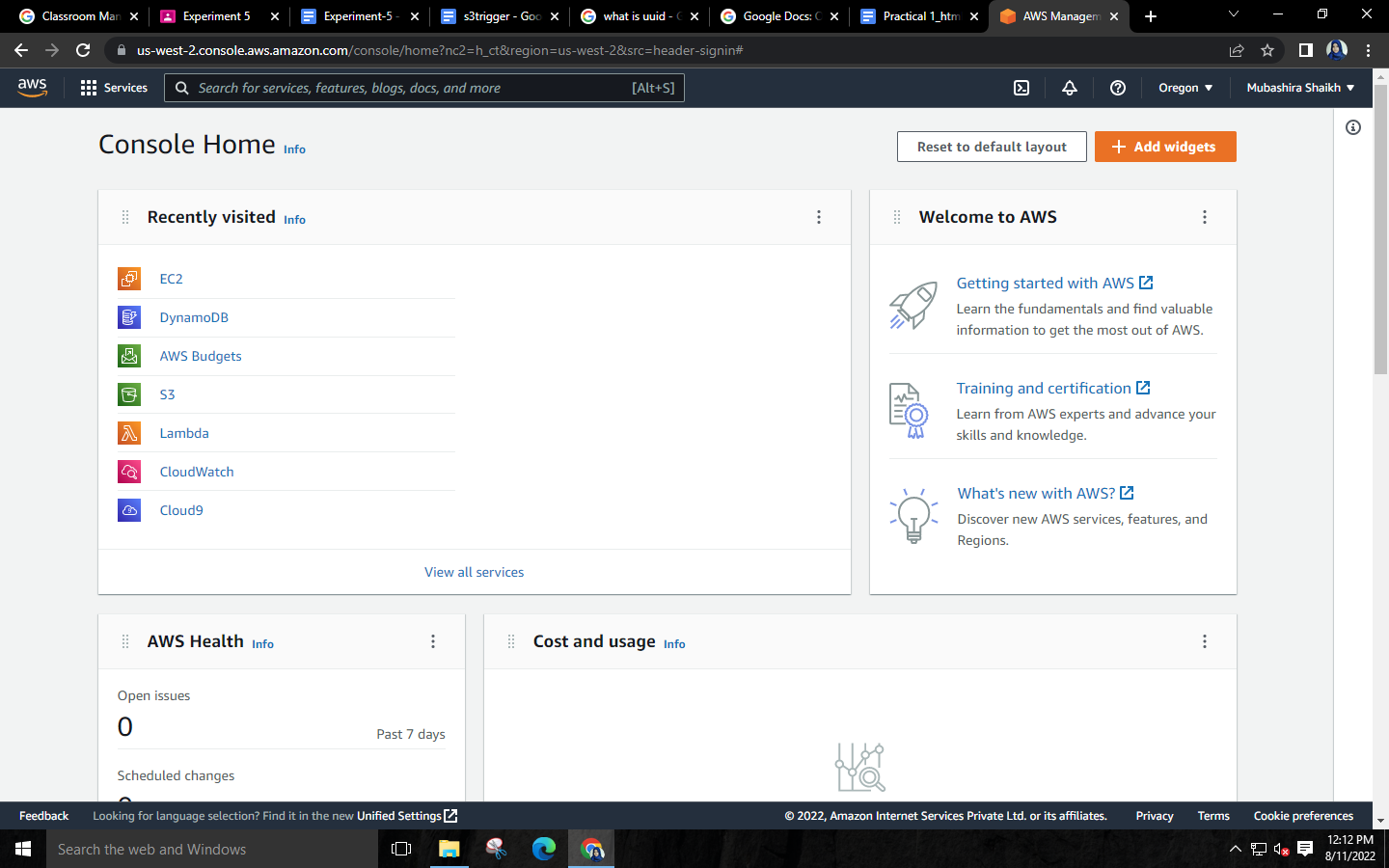
🡪AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

🡪When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the [best practice of using the root user only to create your first IAM user](https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#create-iam-users). Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

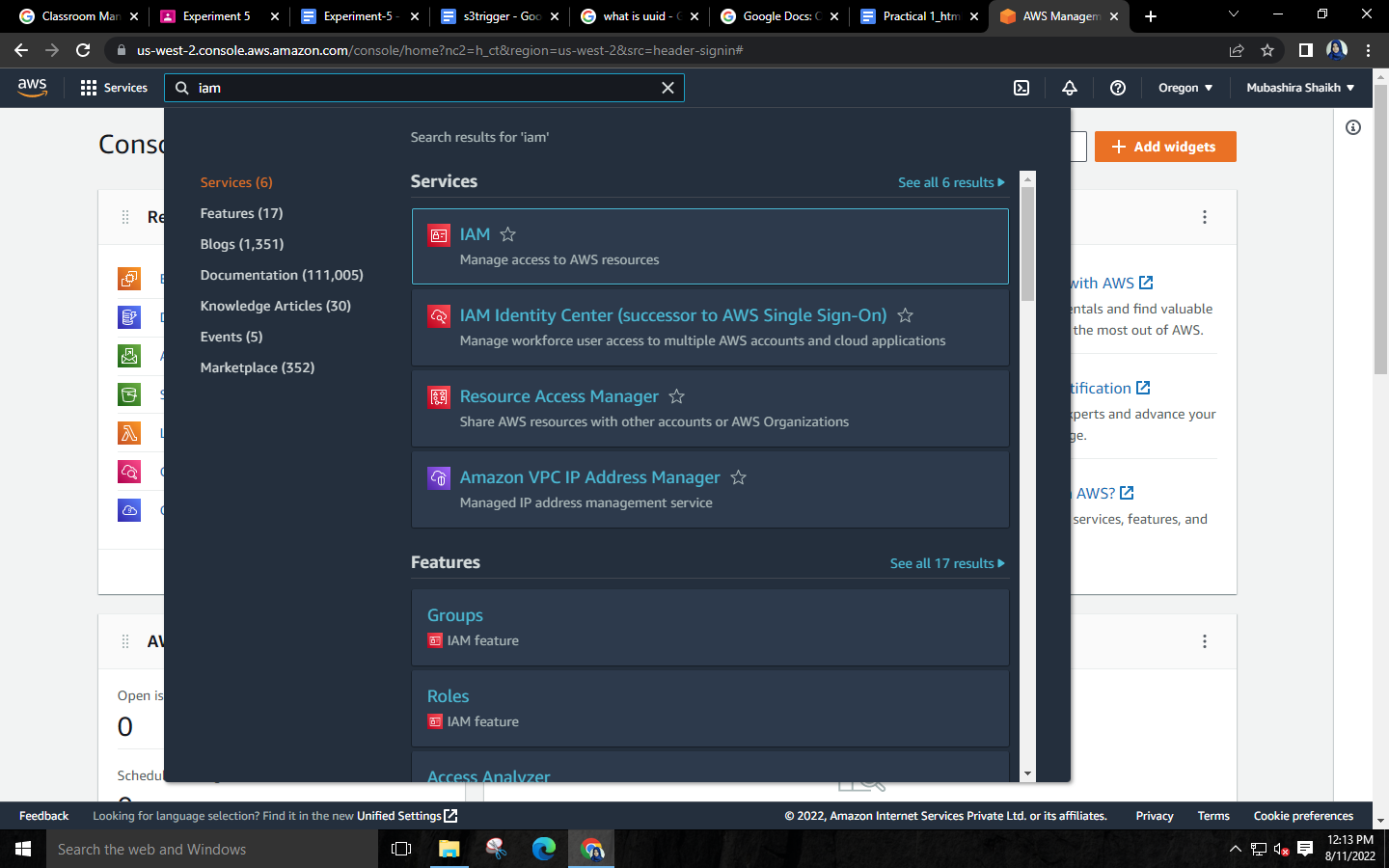
1. **To understand AWS Lambda, create your first Lambda functions using Python / Java / Nodejs.**

**Create AWs Lambda function and configure a trigger for Amazon Simple Storage Service (Amazon S3). The trigger invokes your Lambda function every time that you add an object to your Amazon S3 bucket. Allow AWS Lambda to access Amazon DynamoDB Table .Create IAM role that allows full access to DynamoDB Table.**

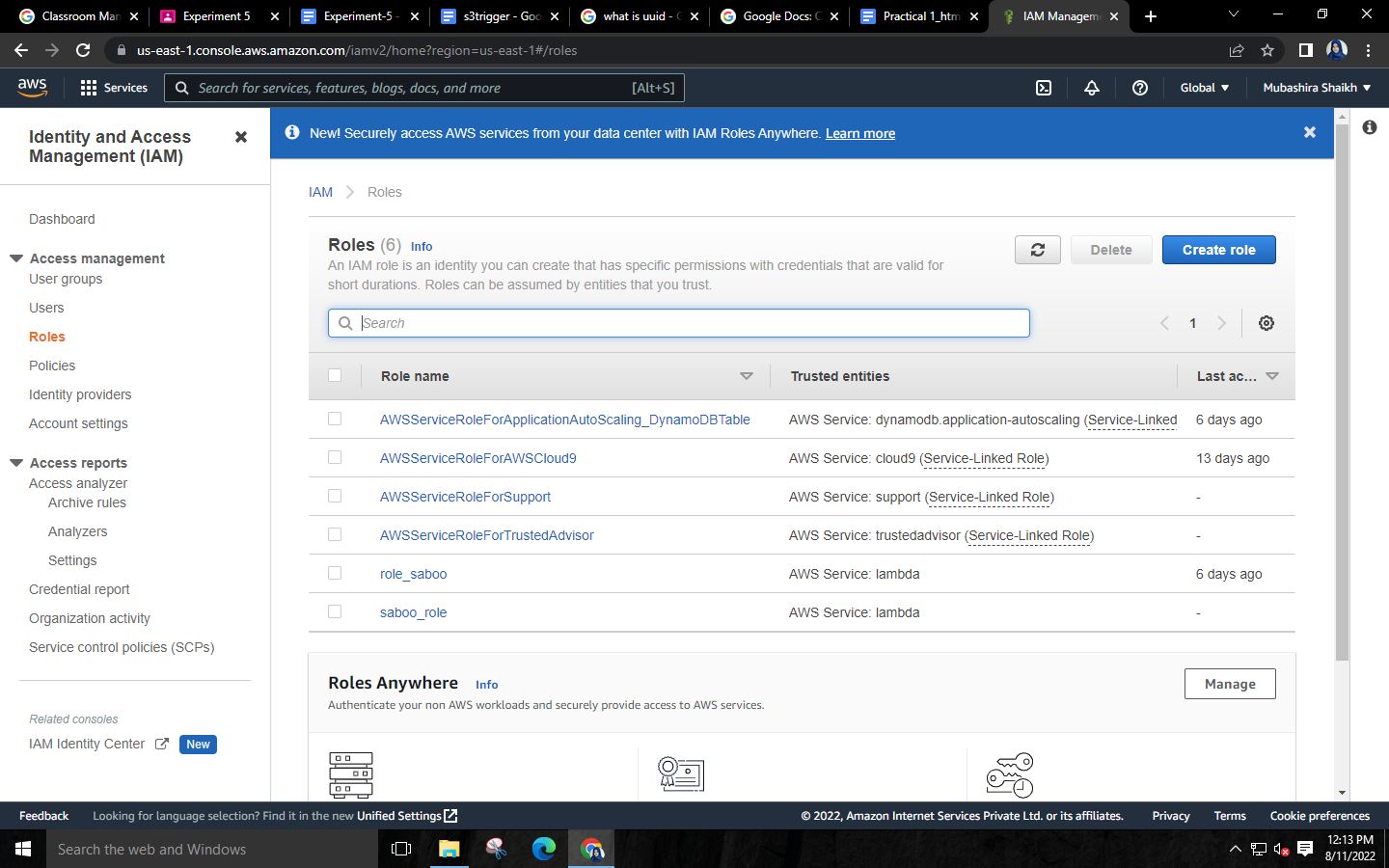
**Step 1: AWS Management Console Dashboard.**

****

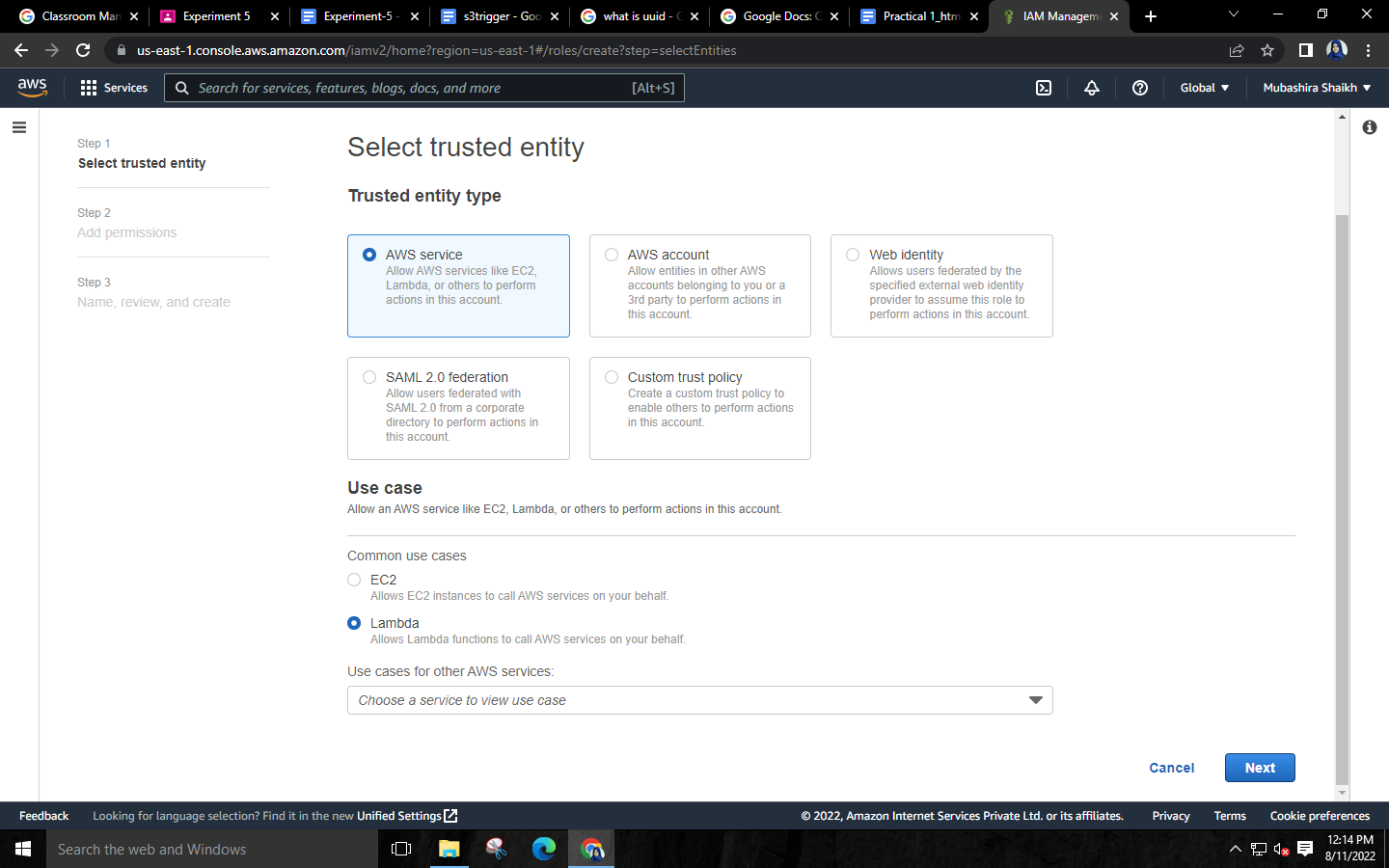
**Step 2: Search for “IAM” and select it.**

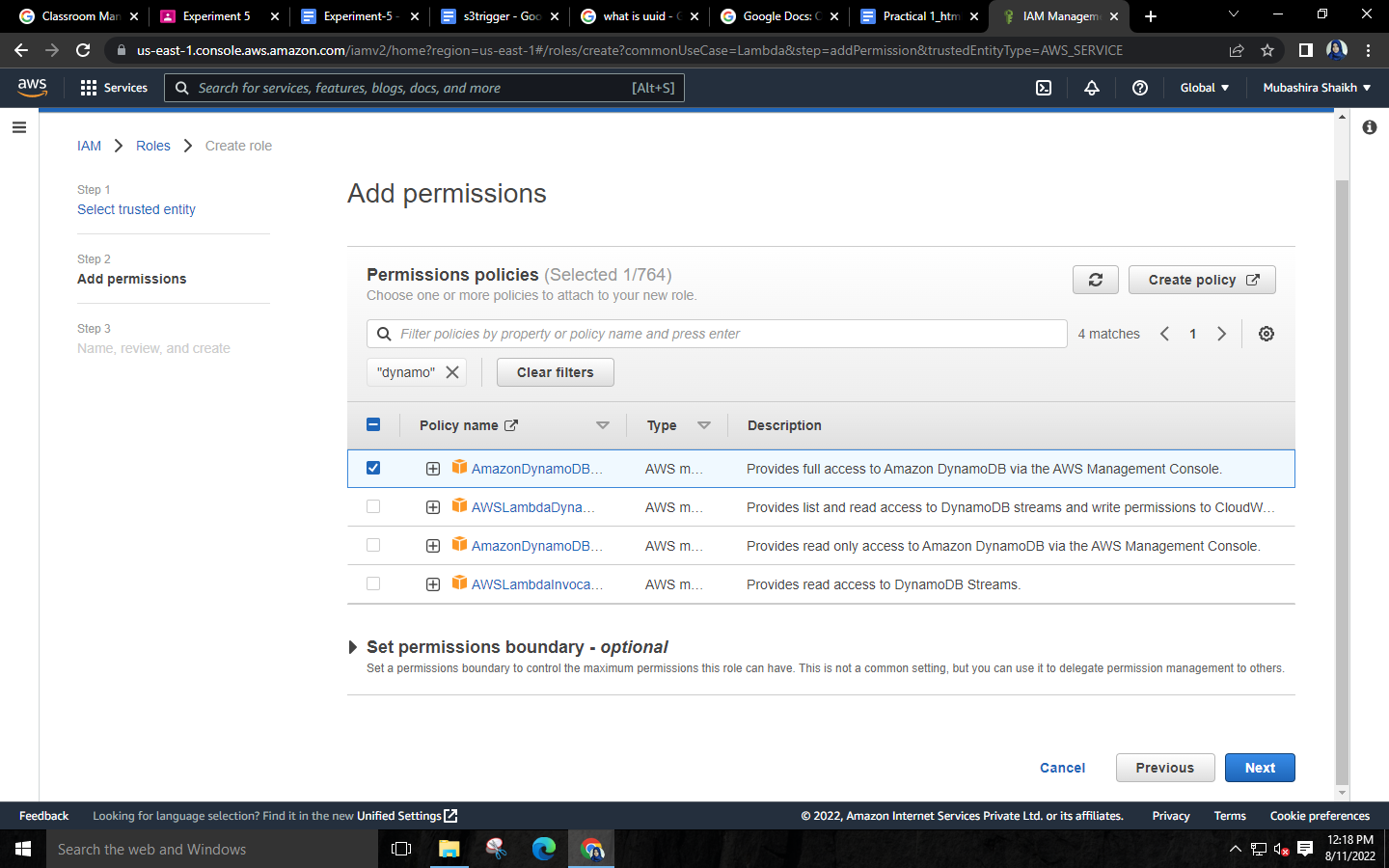
****

**Step 3: Go to role and click on “Create role”.**

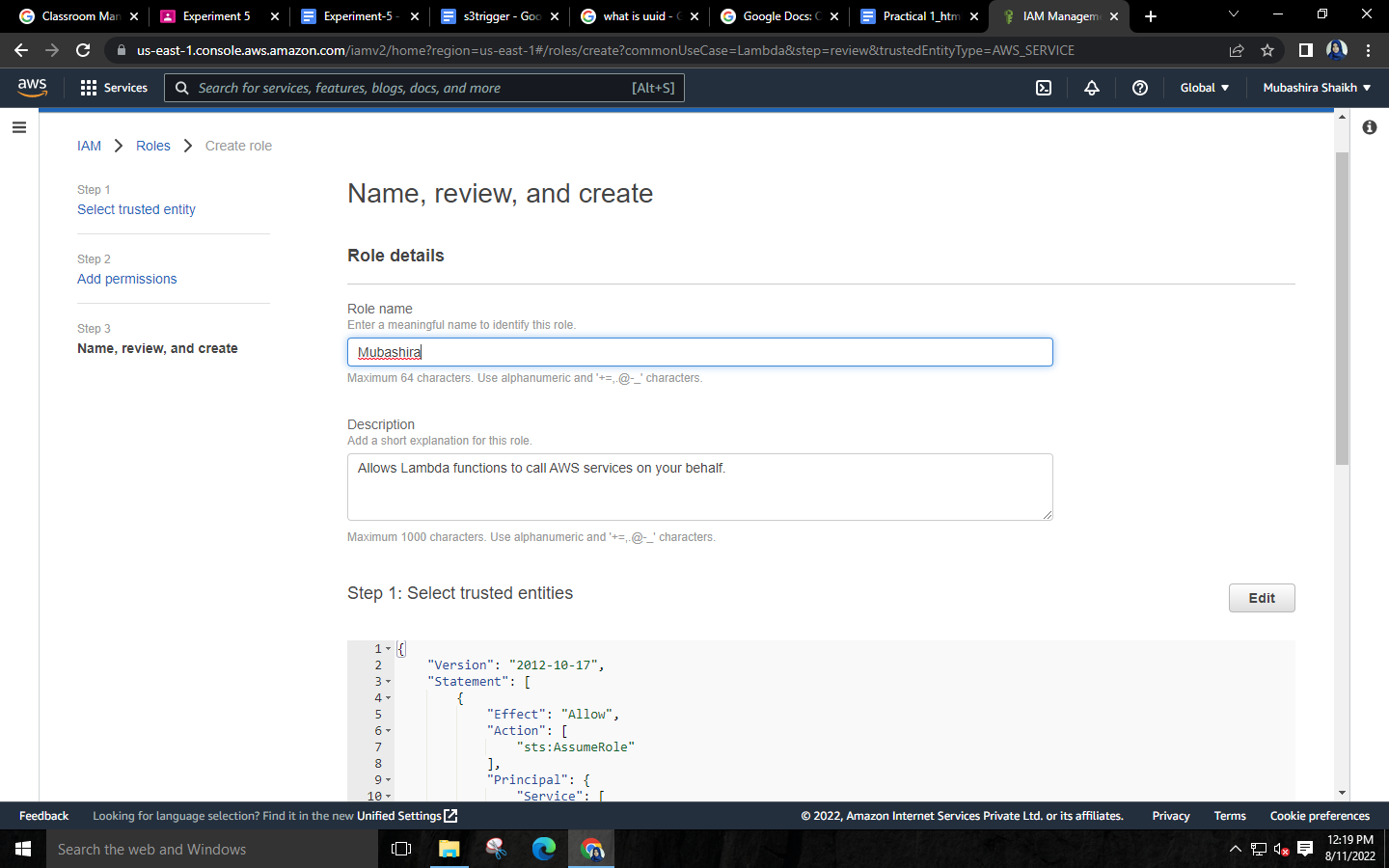
****

**Step 4: Select lambda usecase🡪Click on next.**

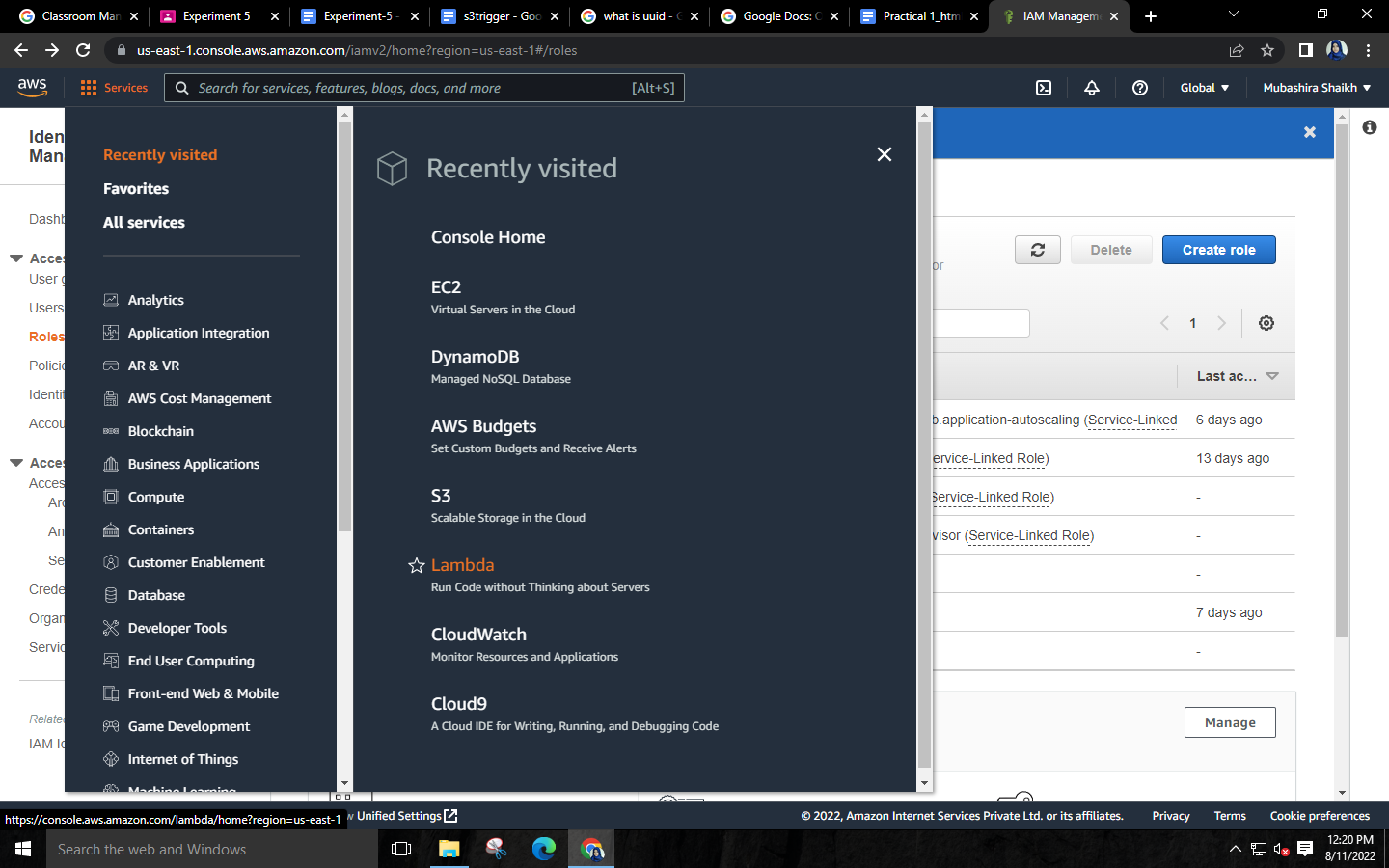
****

**Step 5: Search Amazon DynamoDB in permission policies🡪Select policy which provides “full access to DynamoDB”🡪Click on next.**

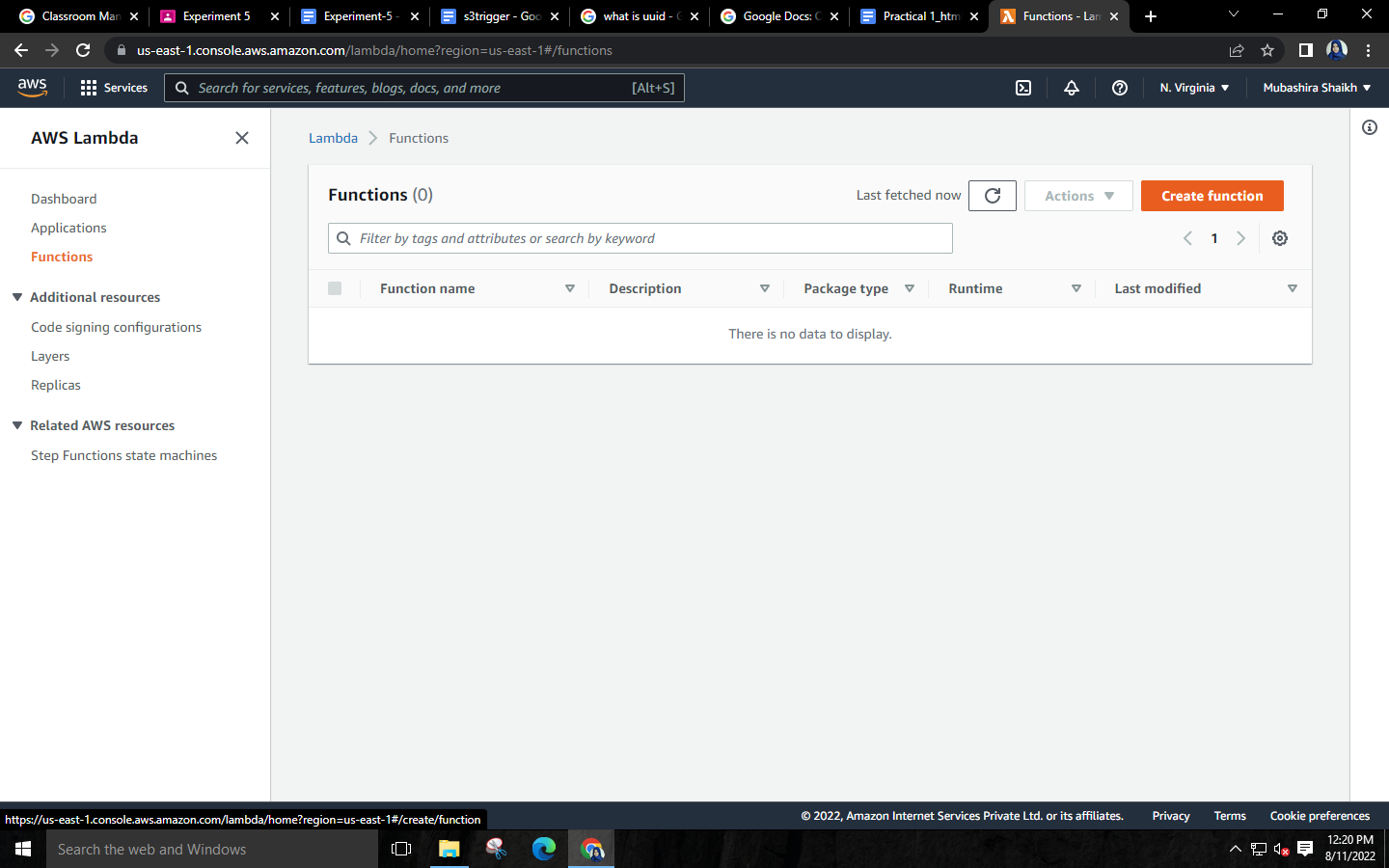
**Step 6: Give name to your role🡪Click on Create role.**

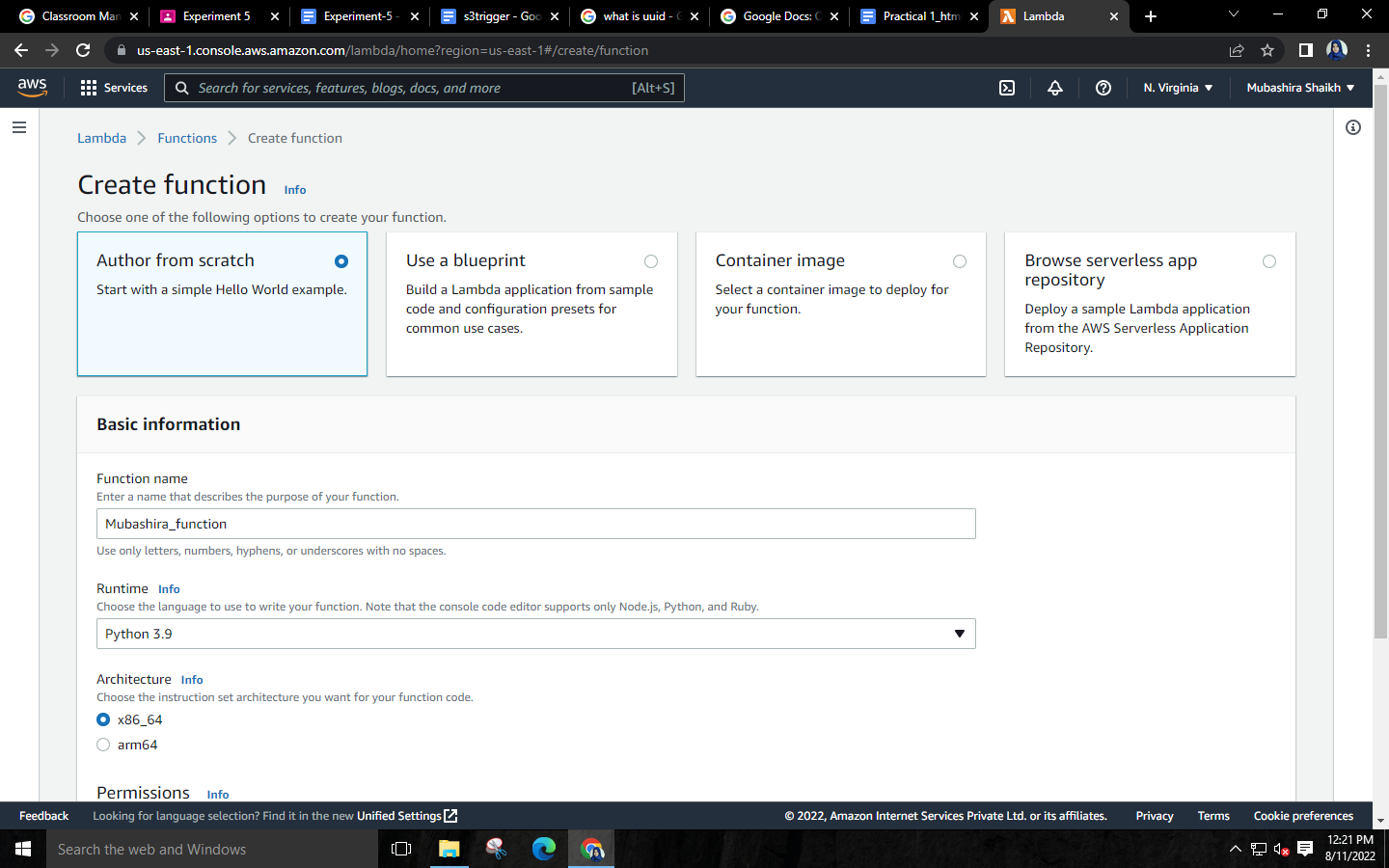
****

**Step 7: Go to services🡪click on “Lambda”.**

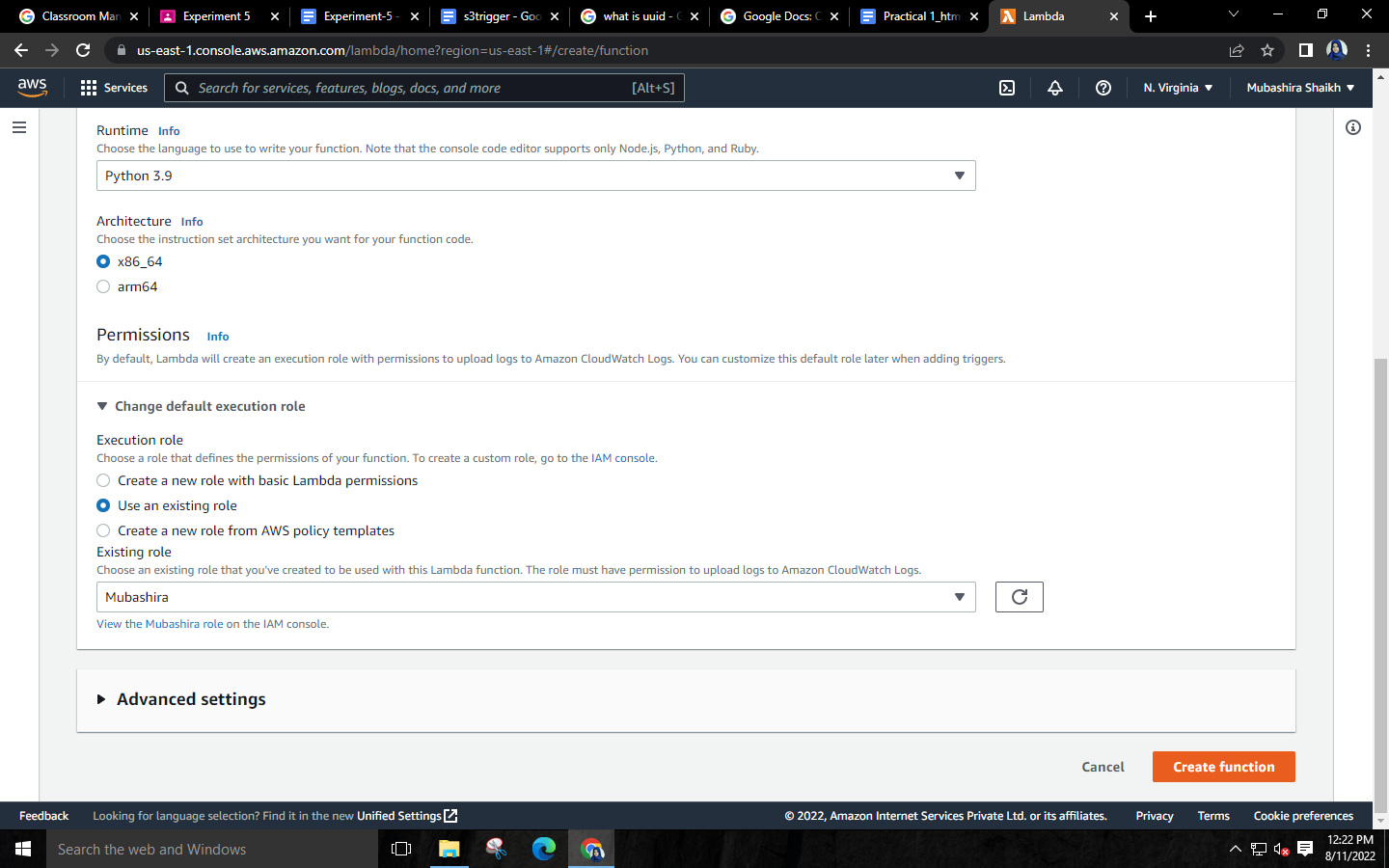
****

**Step 8: Click on Create function.**

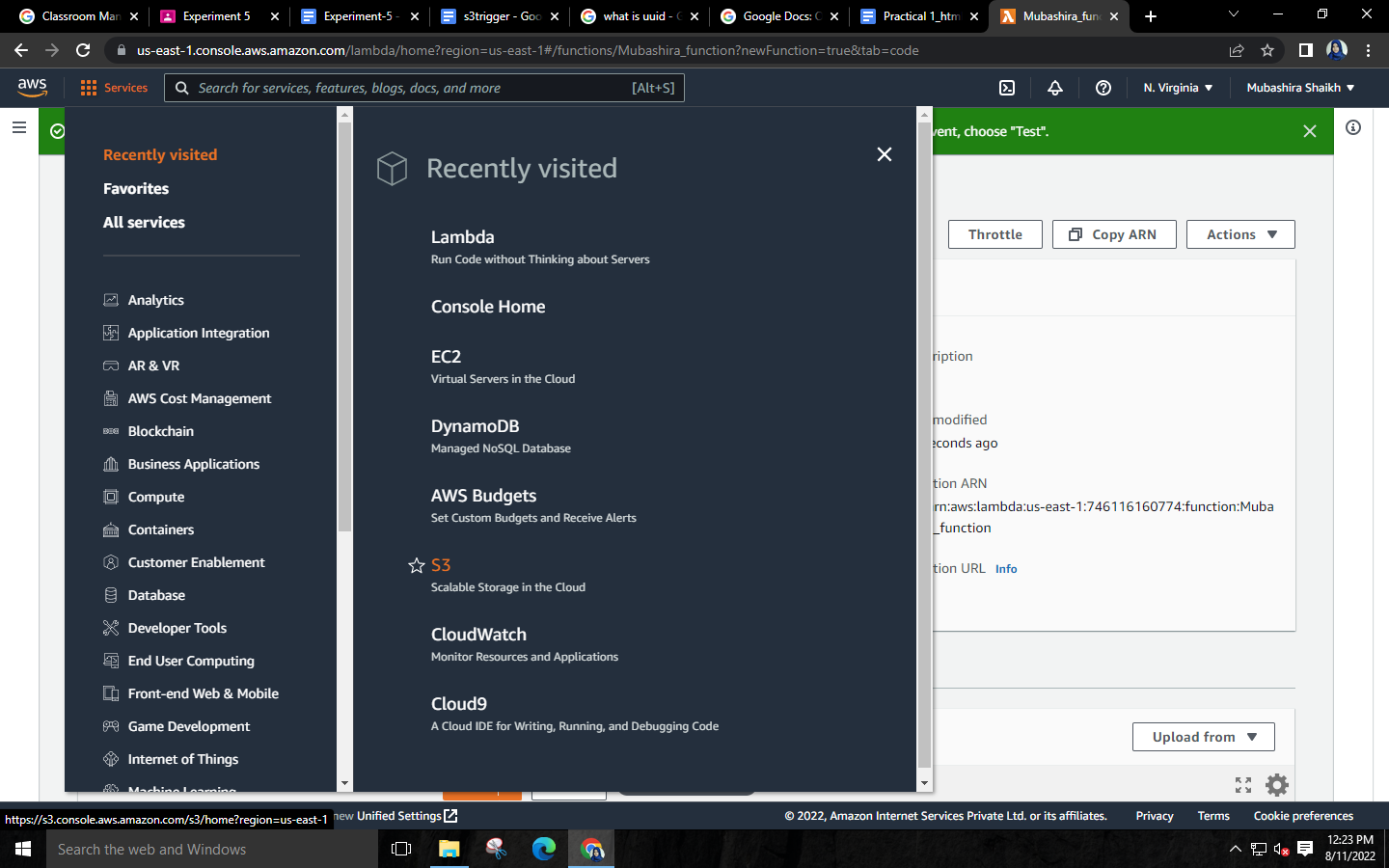
****

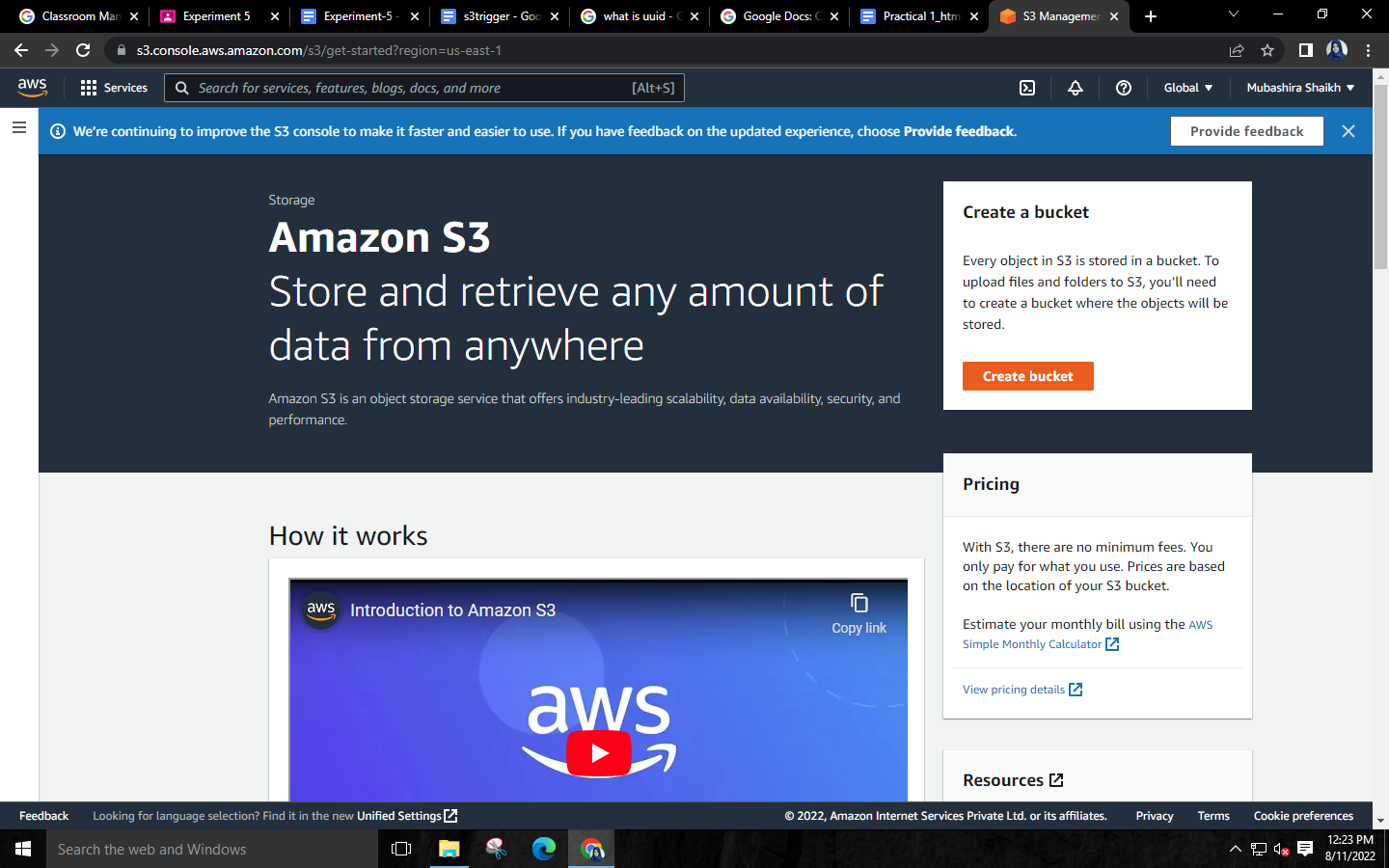
**Step 9: Choose “Author from Scratch”🡪Give name to your function🡪Choose “Python” language in runtime.**

**Step 10: Permissions🡪Change execution role🡪Choose “Use an existing role”🡪Click on Create function.**

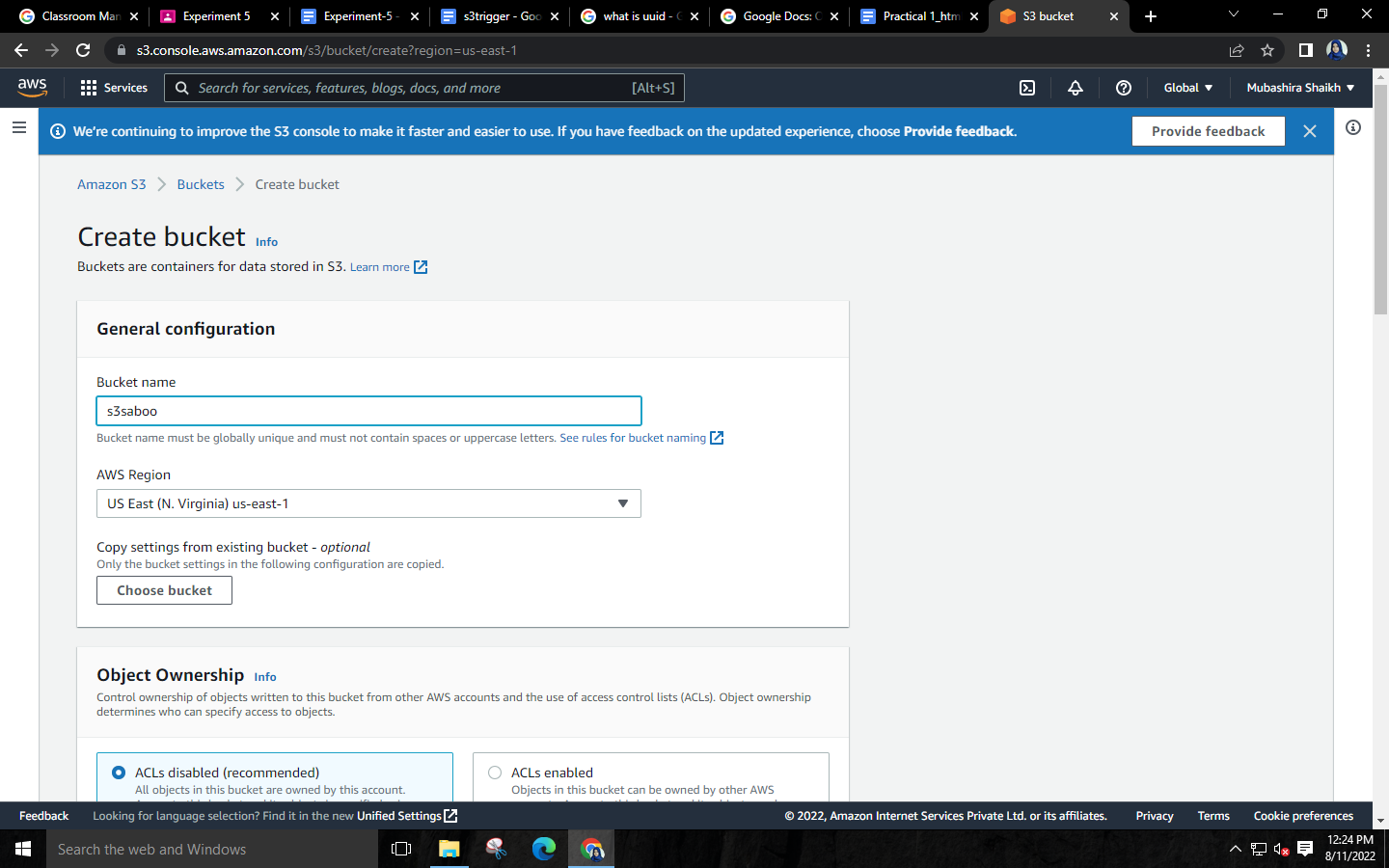
****

**Step 11: Go to Services🡪Select S3.**

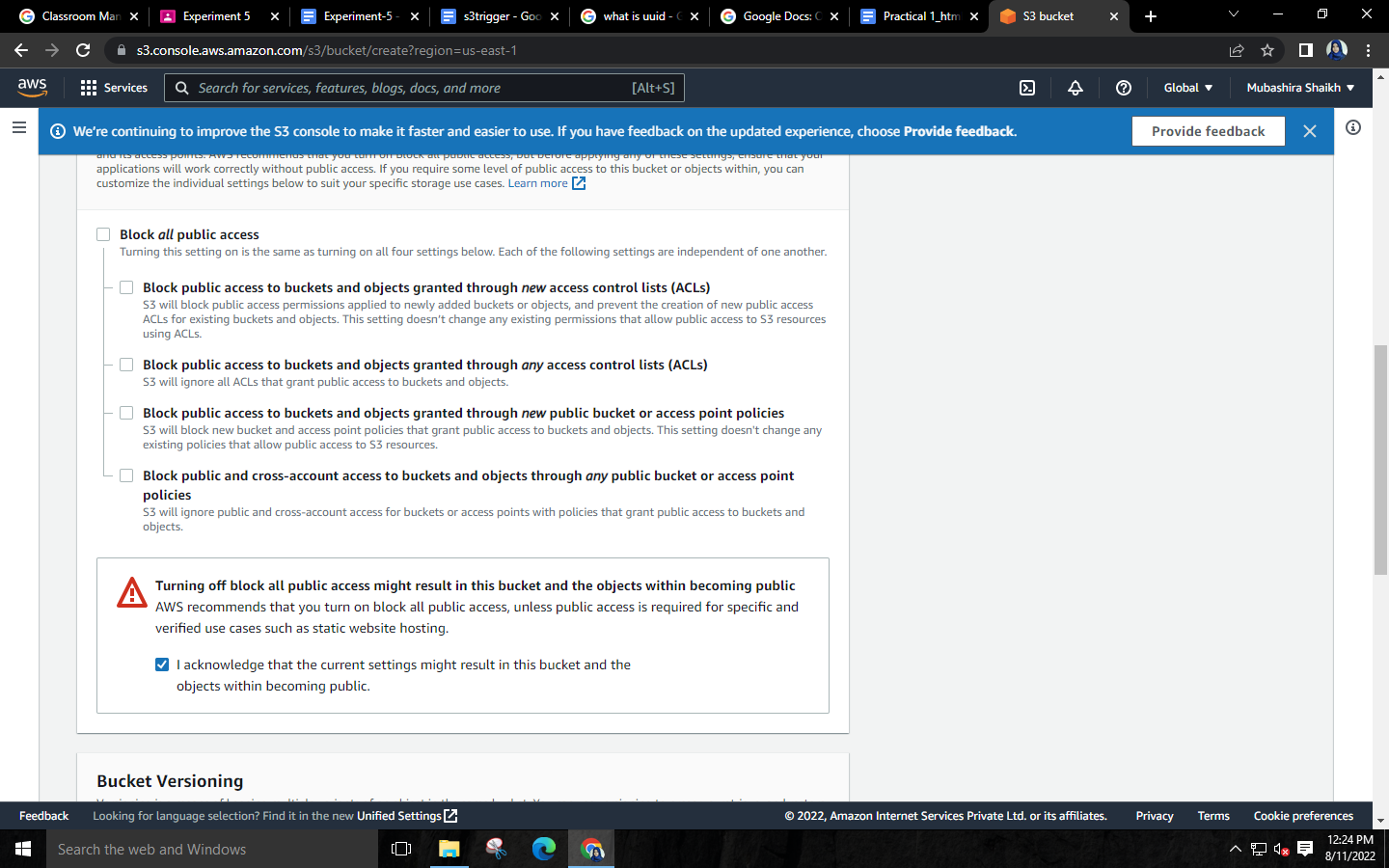
****

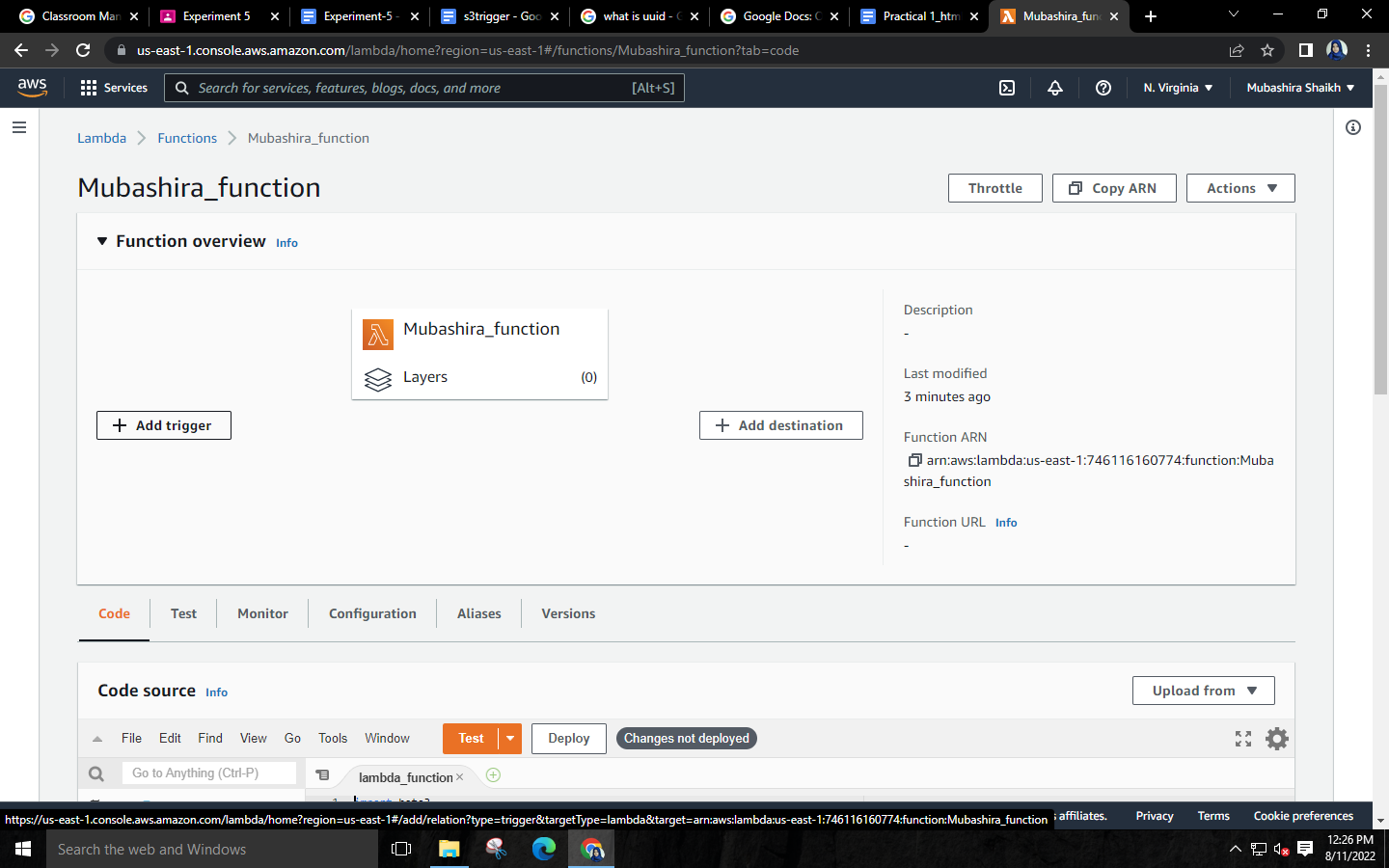
**Step 12: Click on “Create bucket”.**

**Step 13: Give name to your bucket.**

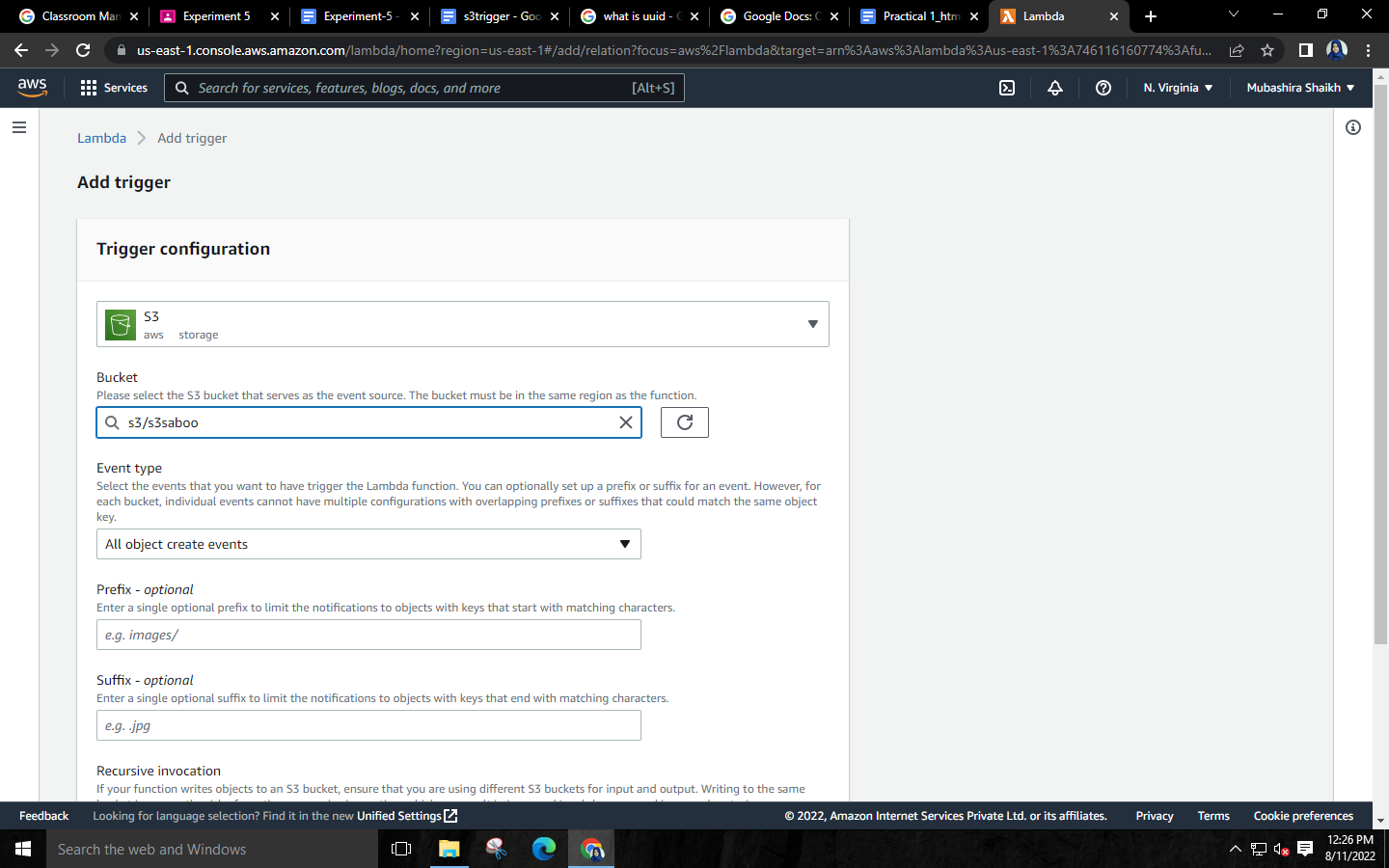
****

**Step 14: Unblock all public access.**

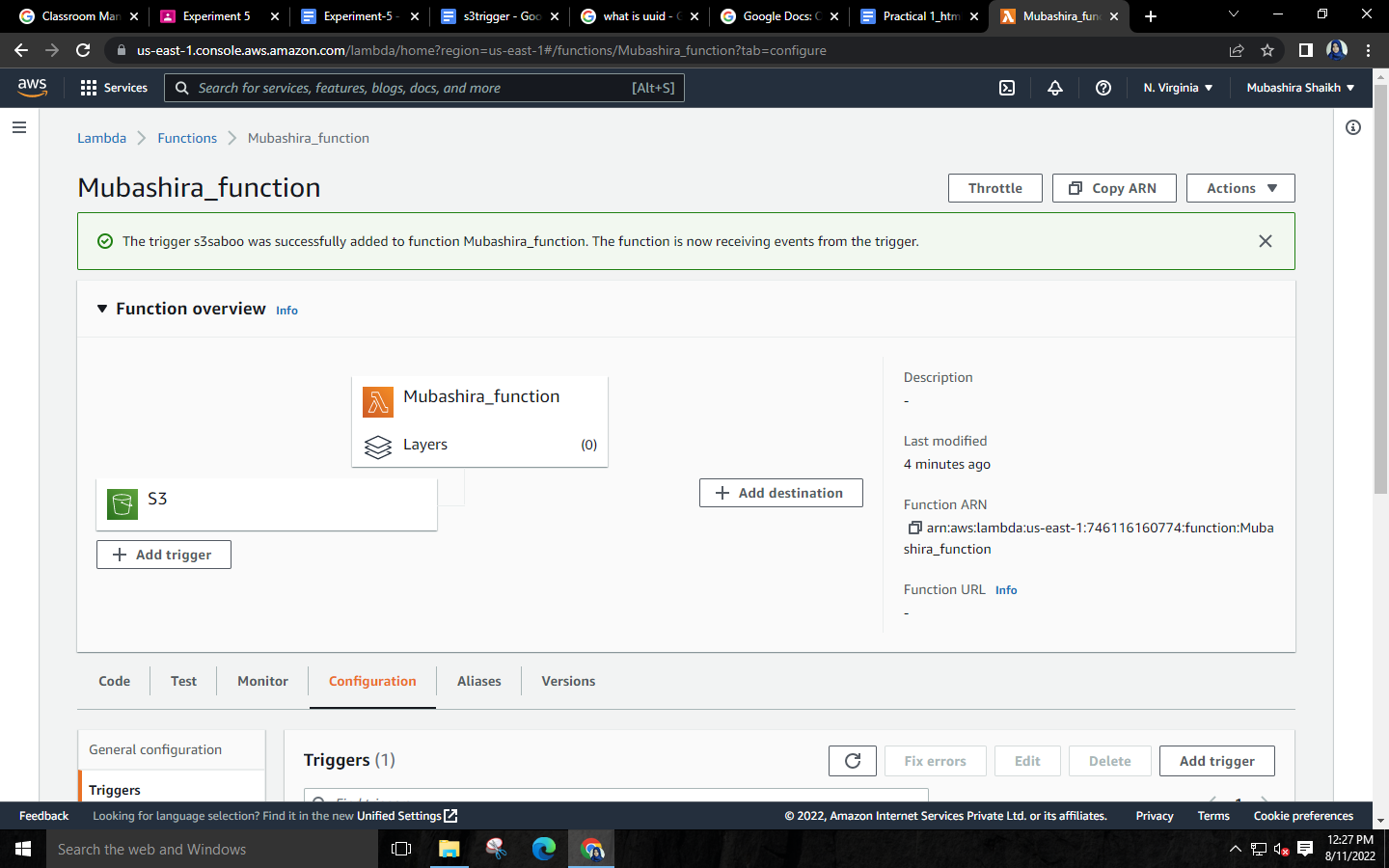
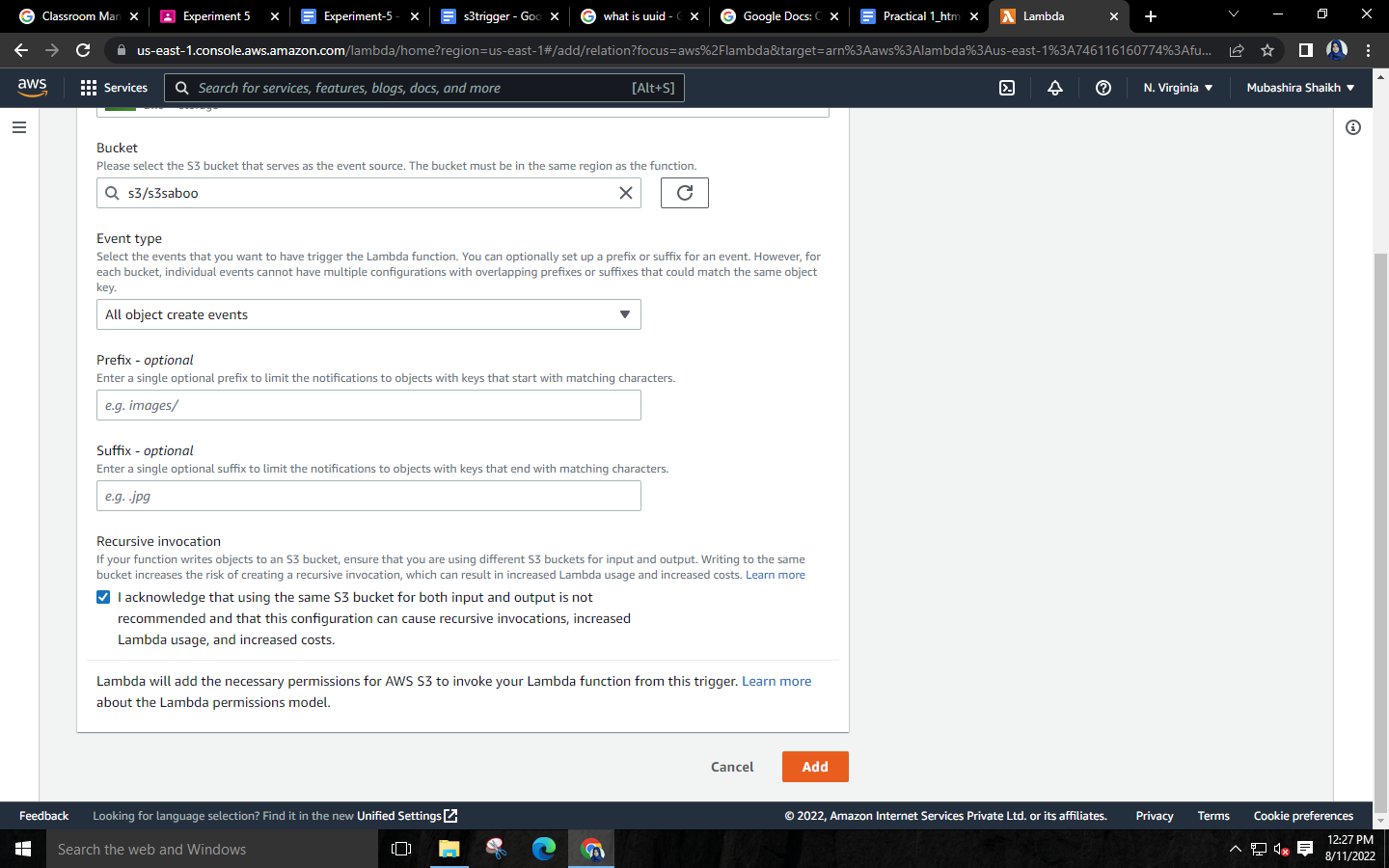
**Step 15: Click on Add trigger.**

****

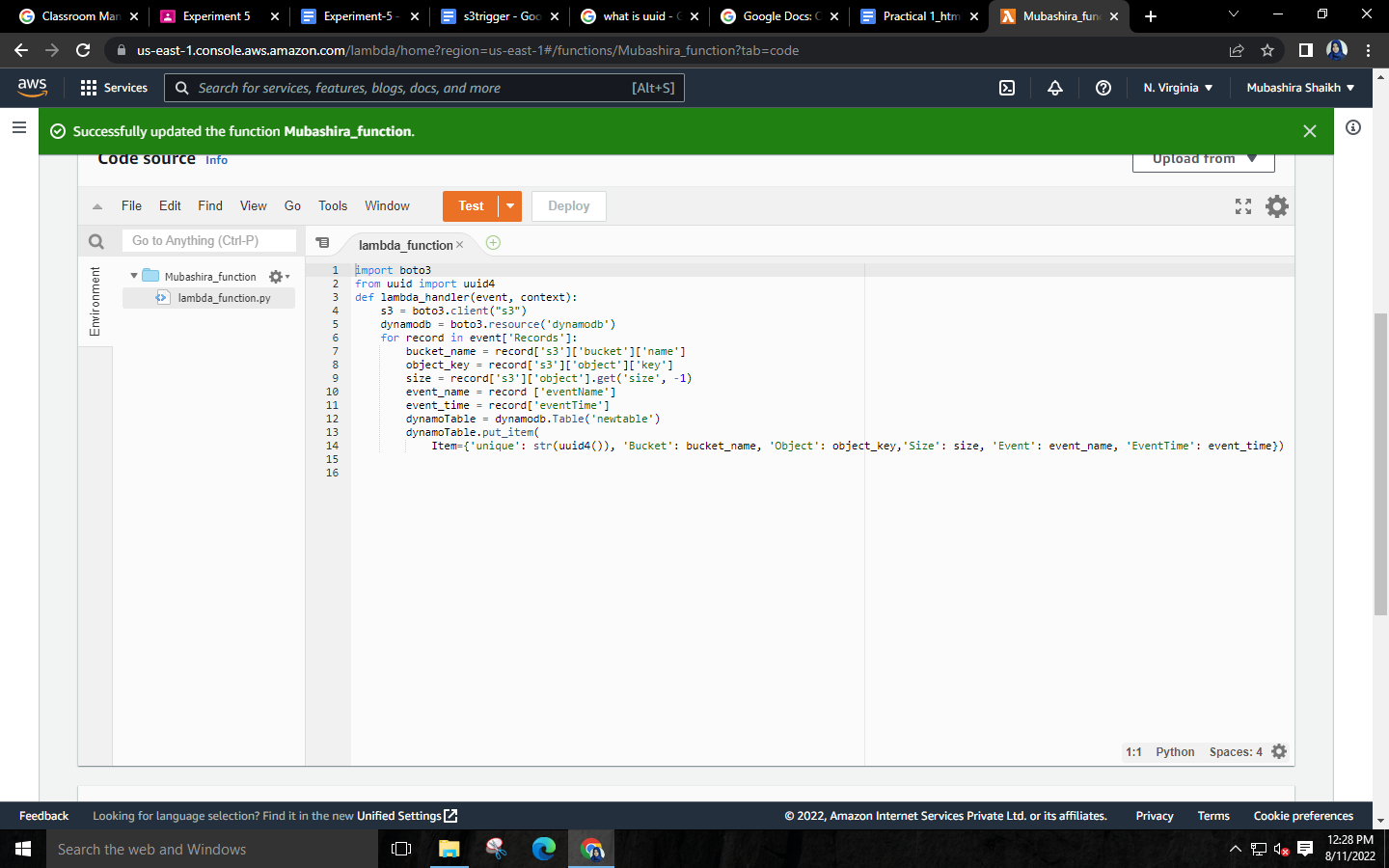
**Step 16: Select your S3 bucket.**

****

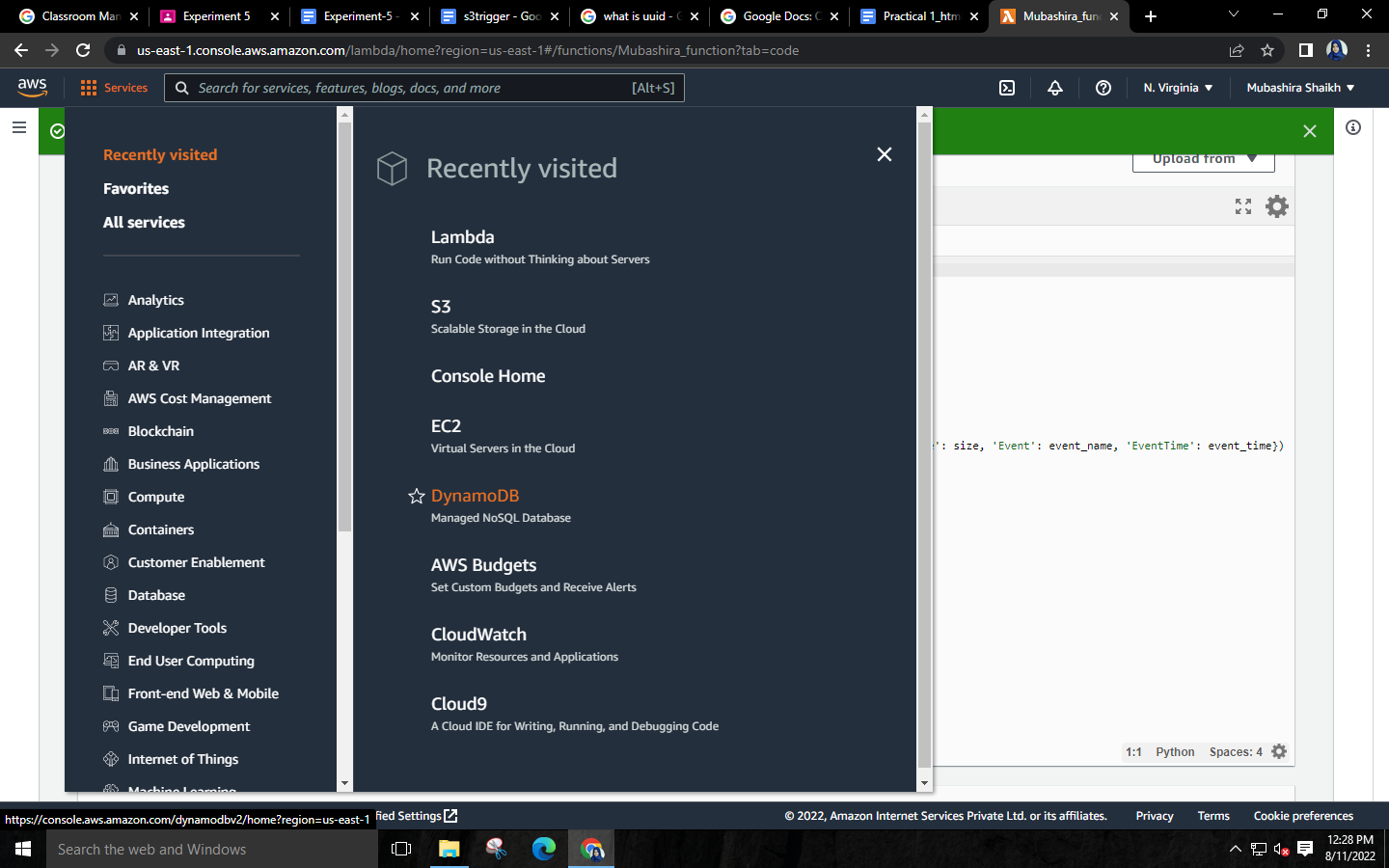
**Step 17: Acknowledge Recursive invocation.**

****

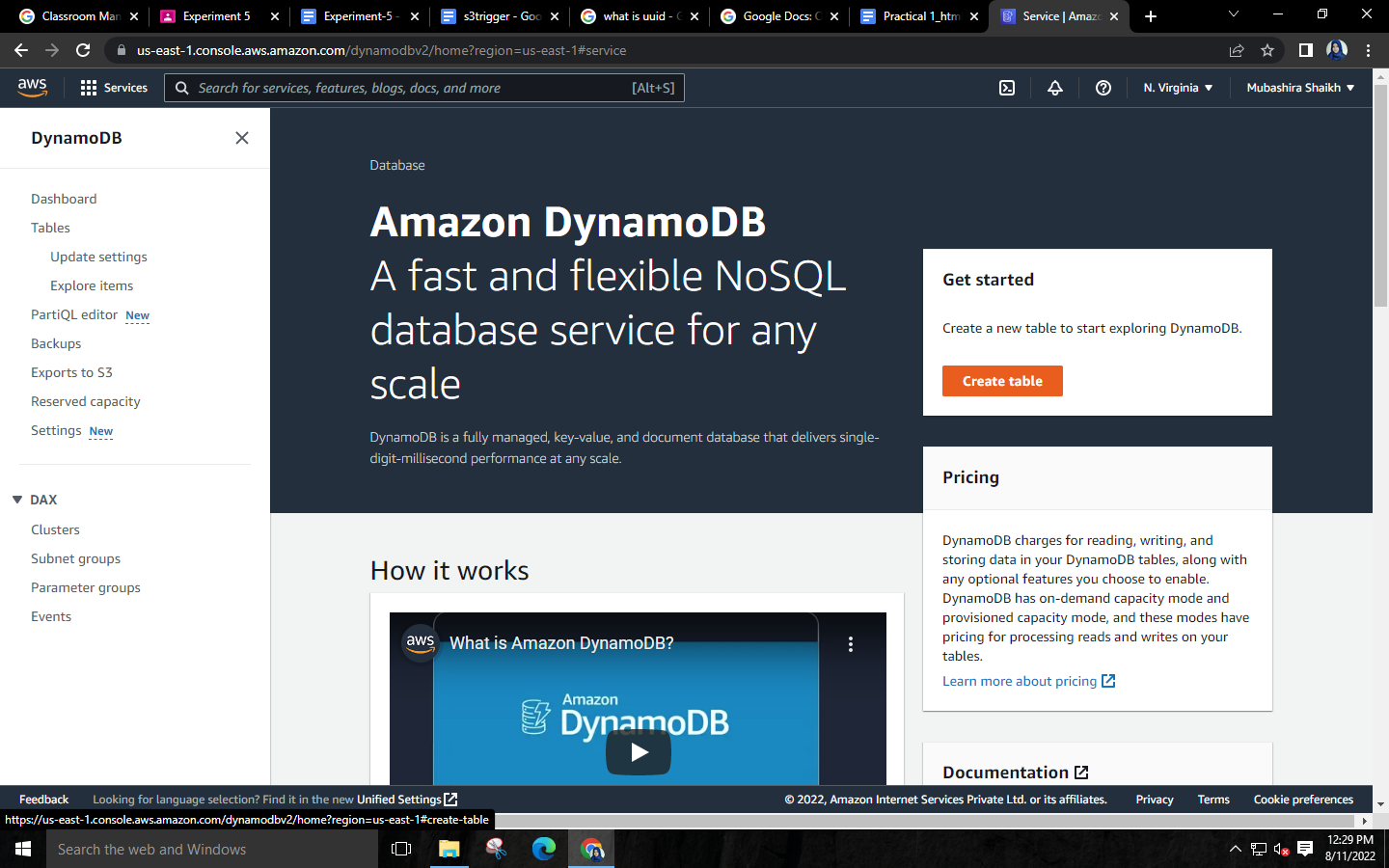
**Step 18: Write the python code for S3 trigger🡪Save the code🡪Deploy it.**

****

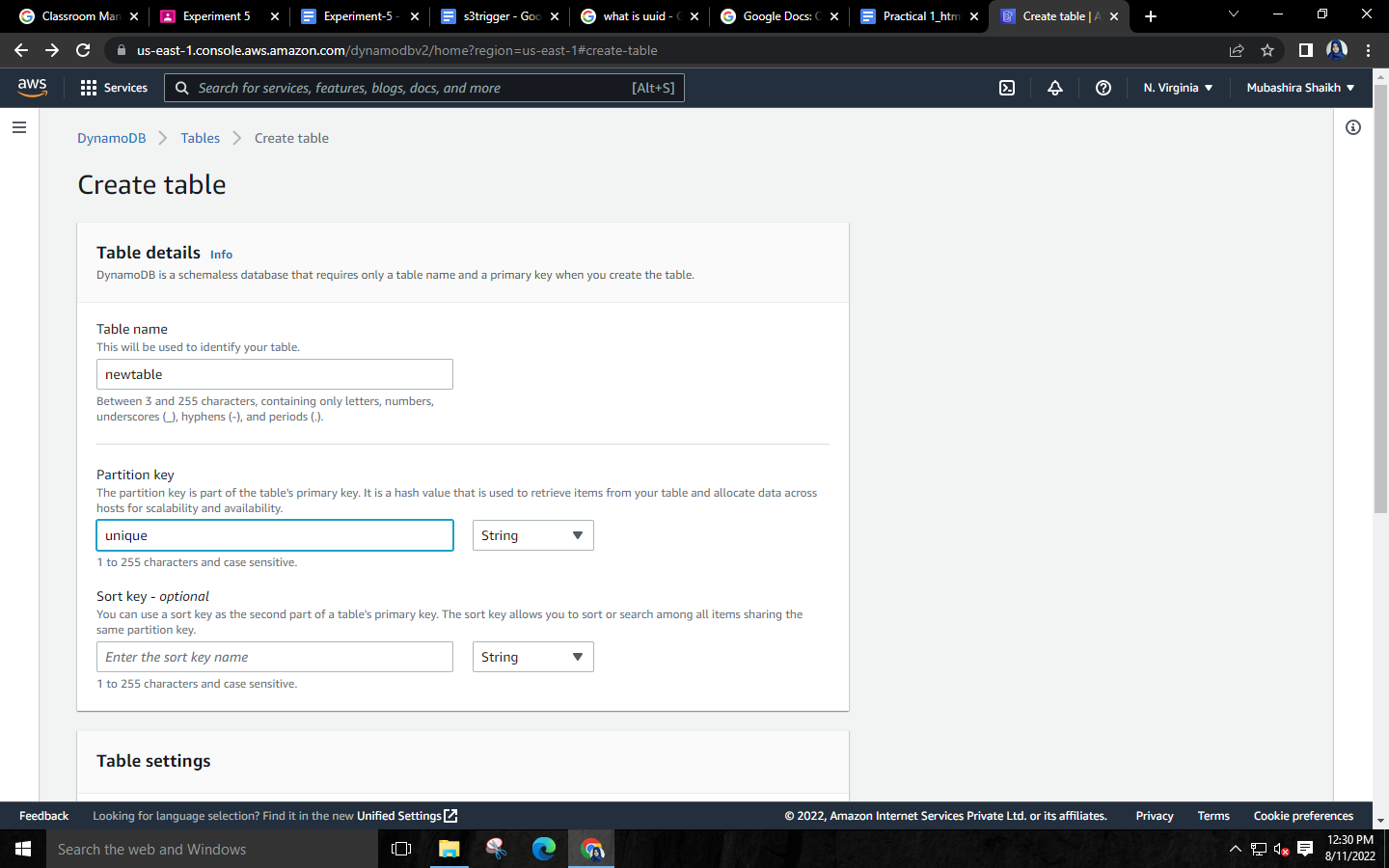
**Step 19: Go to services🡪Select DynamoDB.**

****

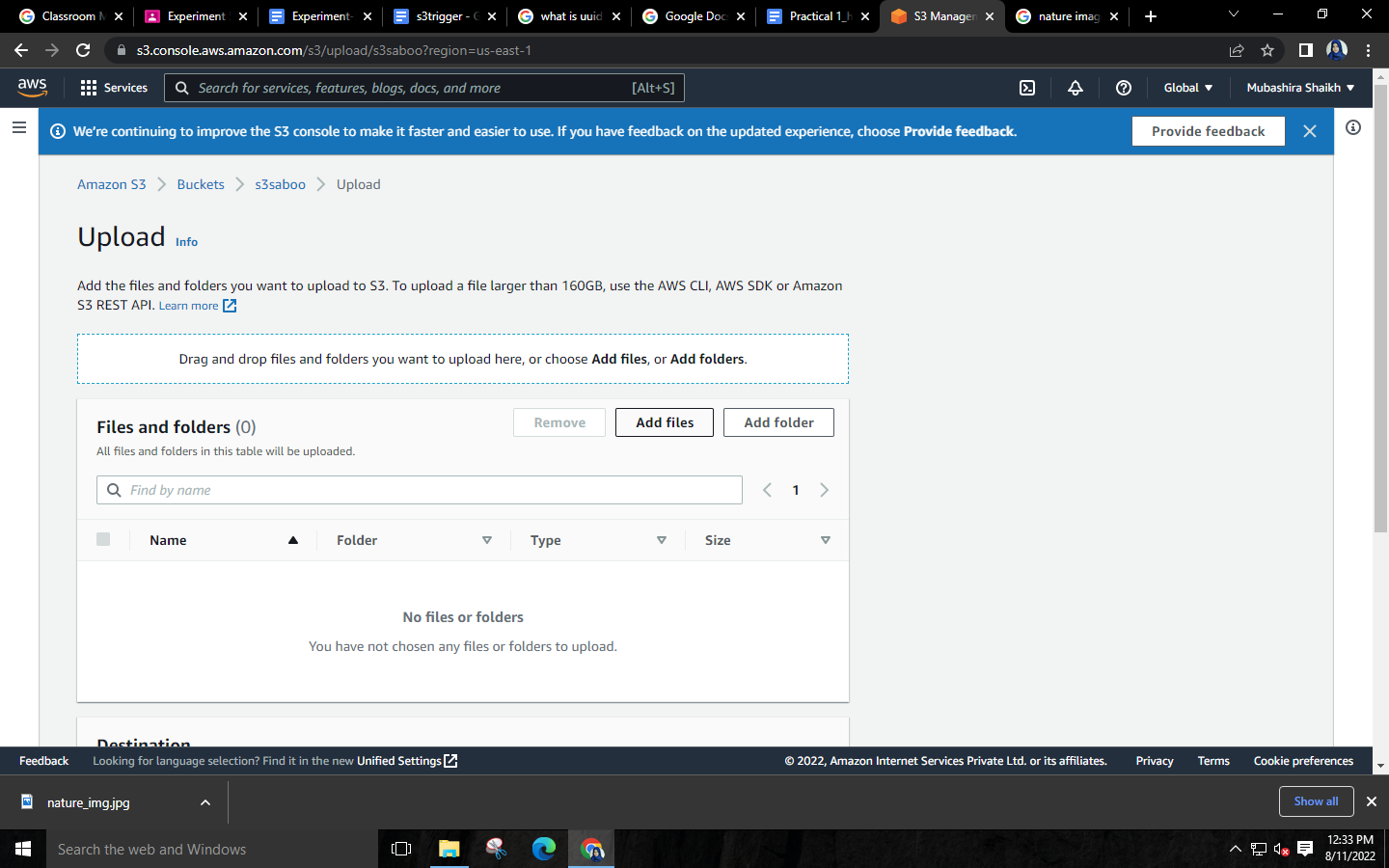
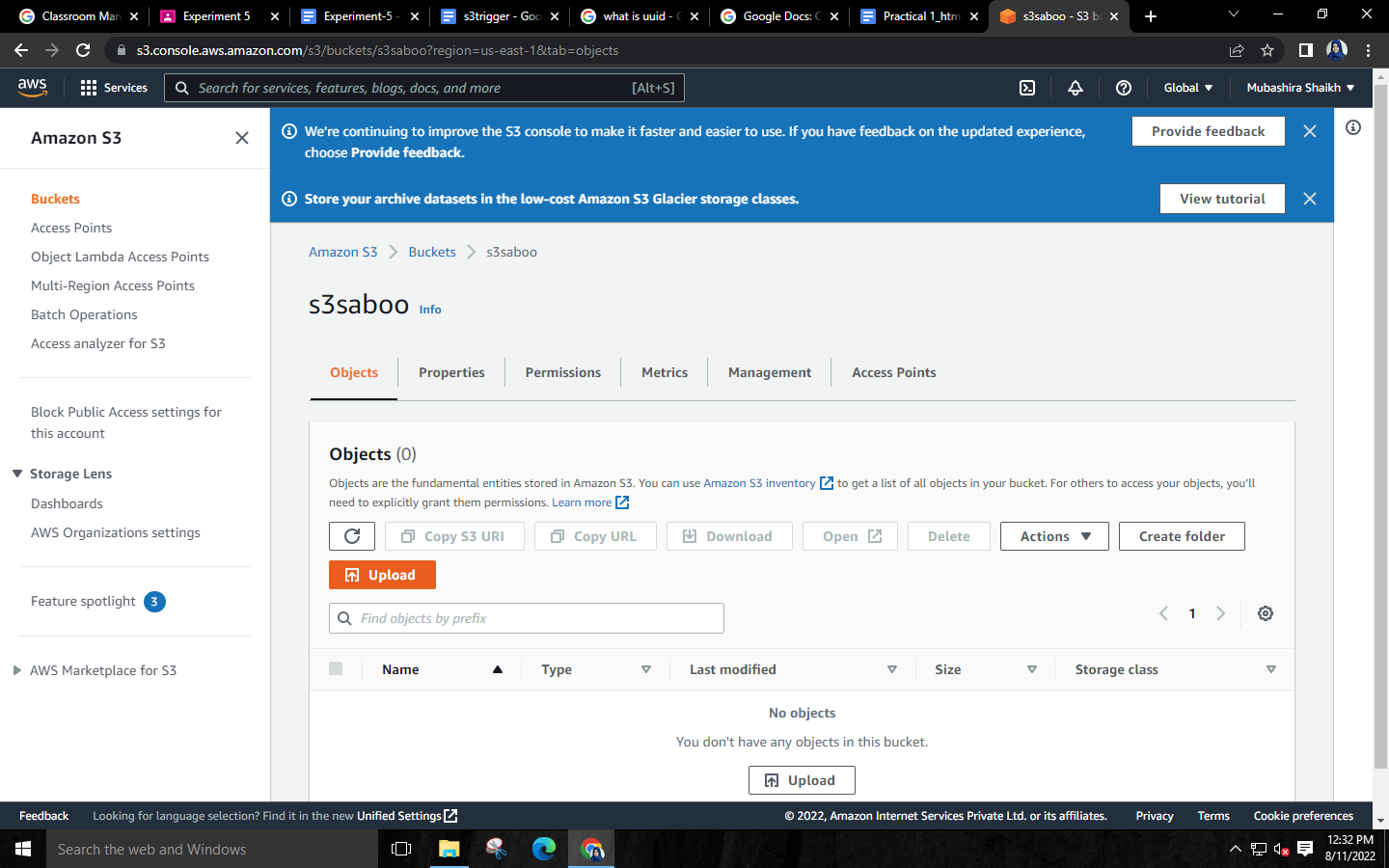
**Step 20: Create the table.**

****

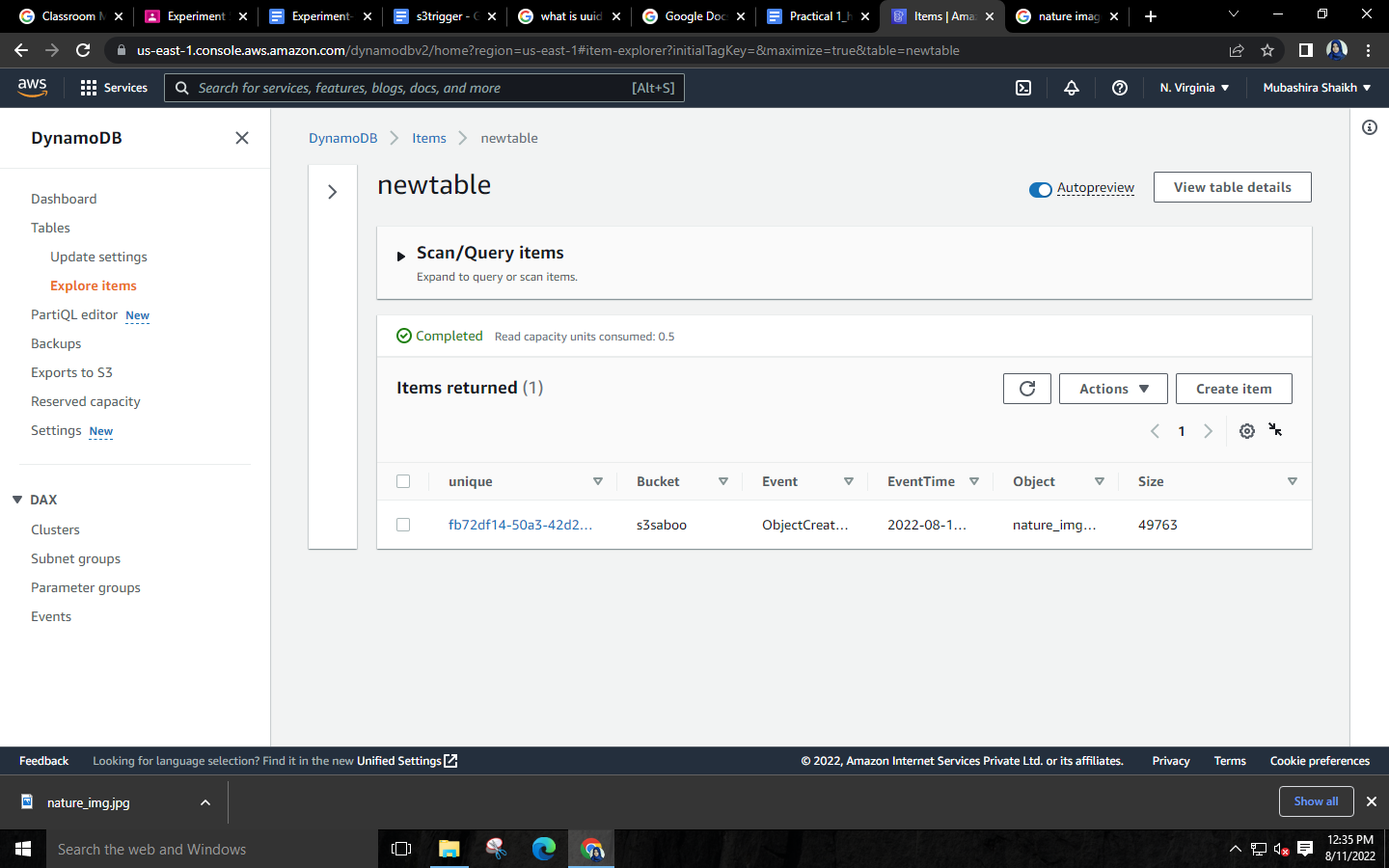
**Step 21: Give name and partition key same as mentioned in the code🡪Click create table.**

****

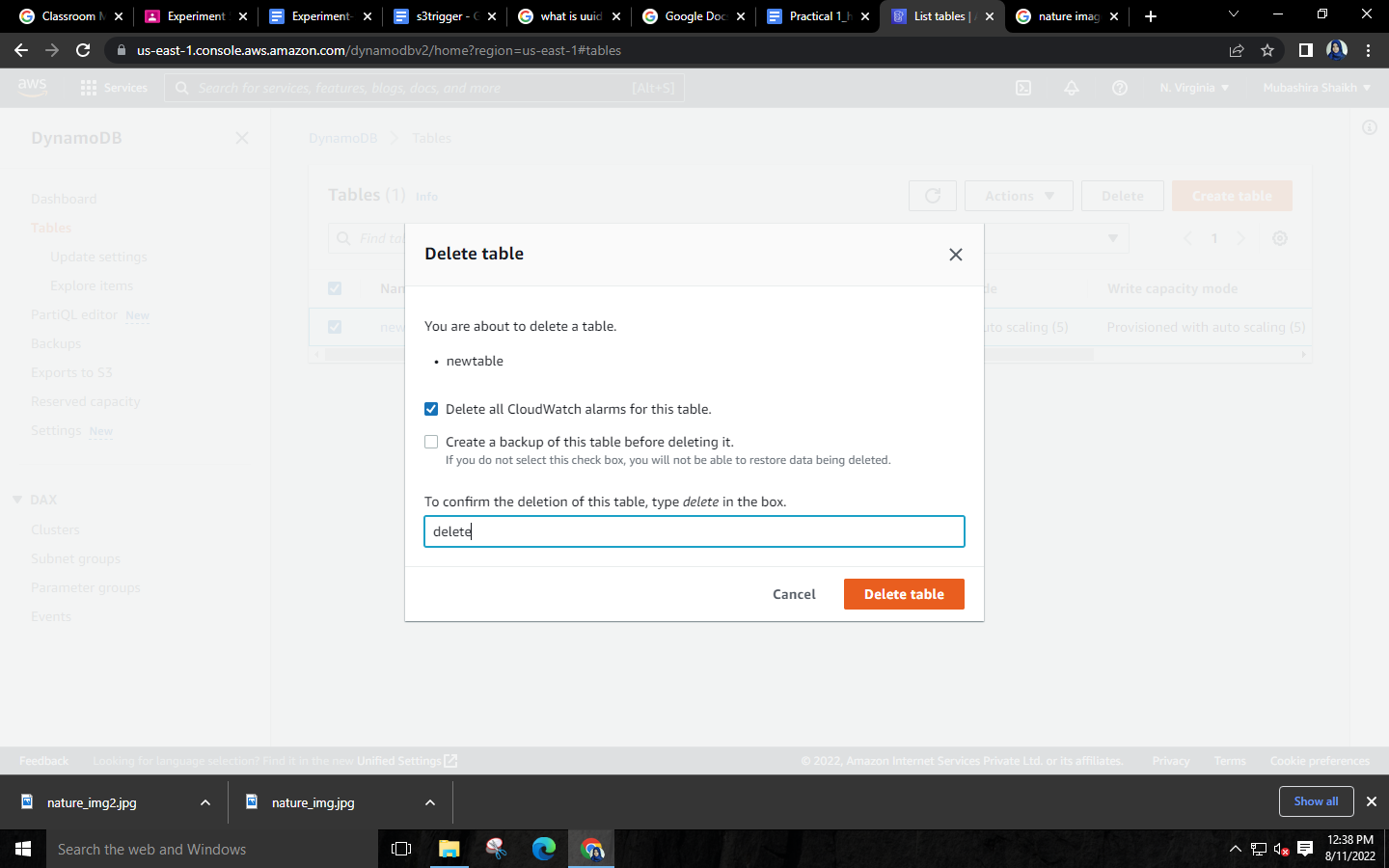
**Step 22: Go to your bucket and upload an object by clicking on add files.**

****

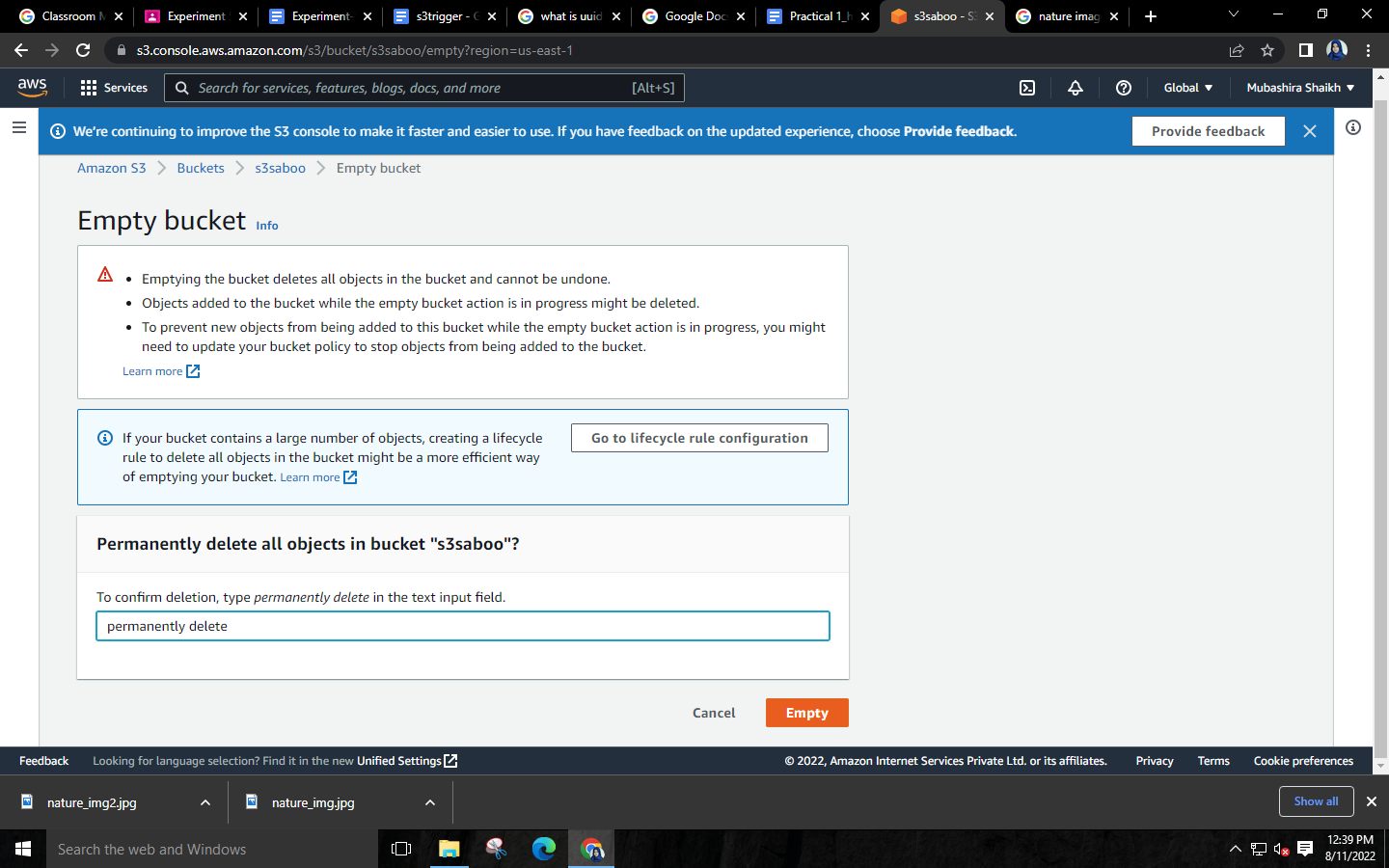
**Step 23: Go to your table and check the details of the uploaded file.**

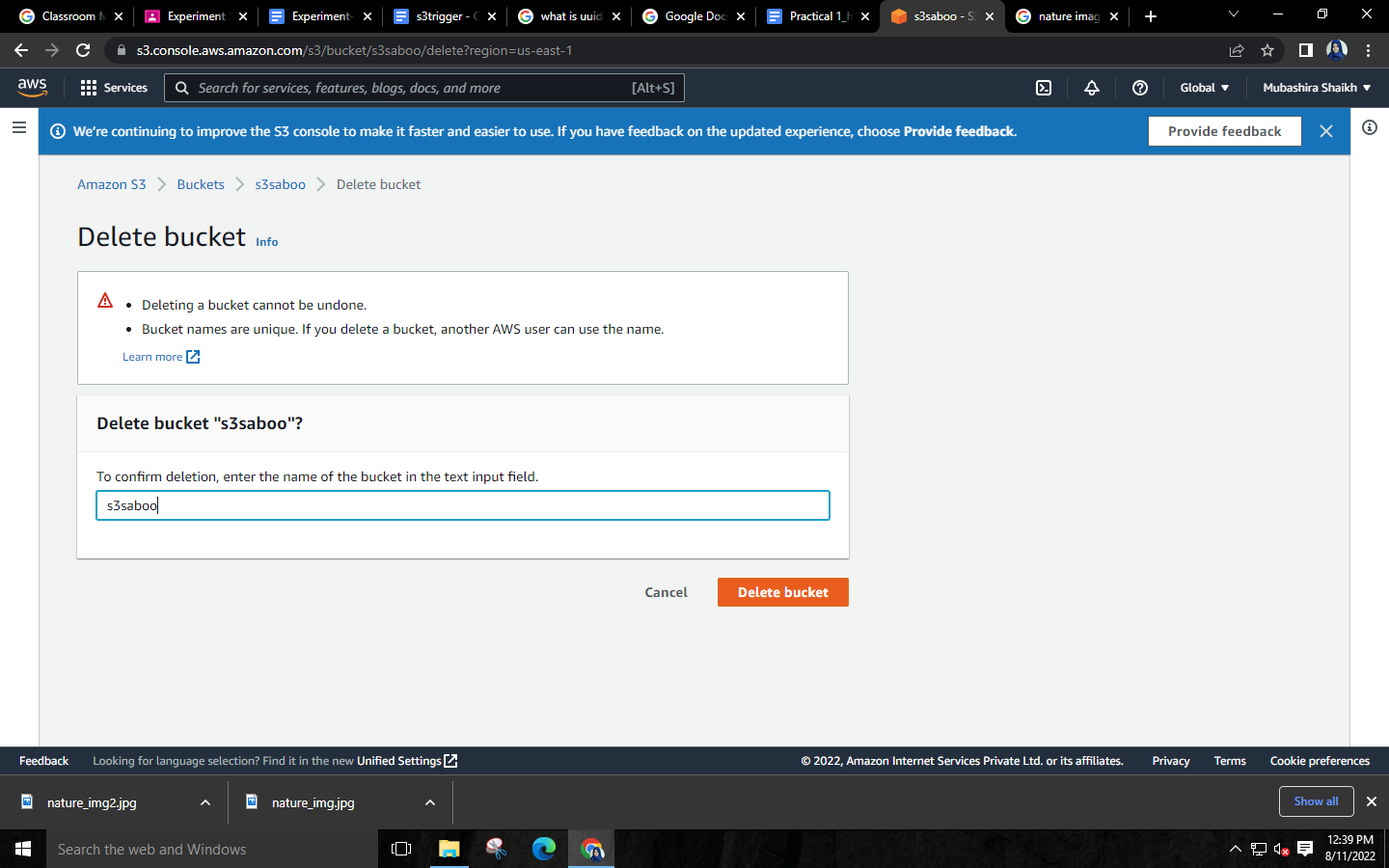
****

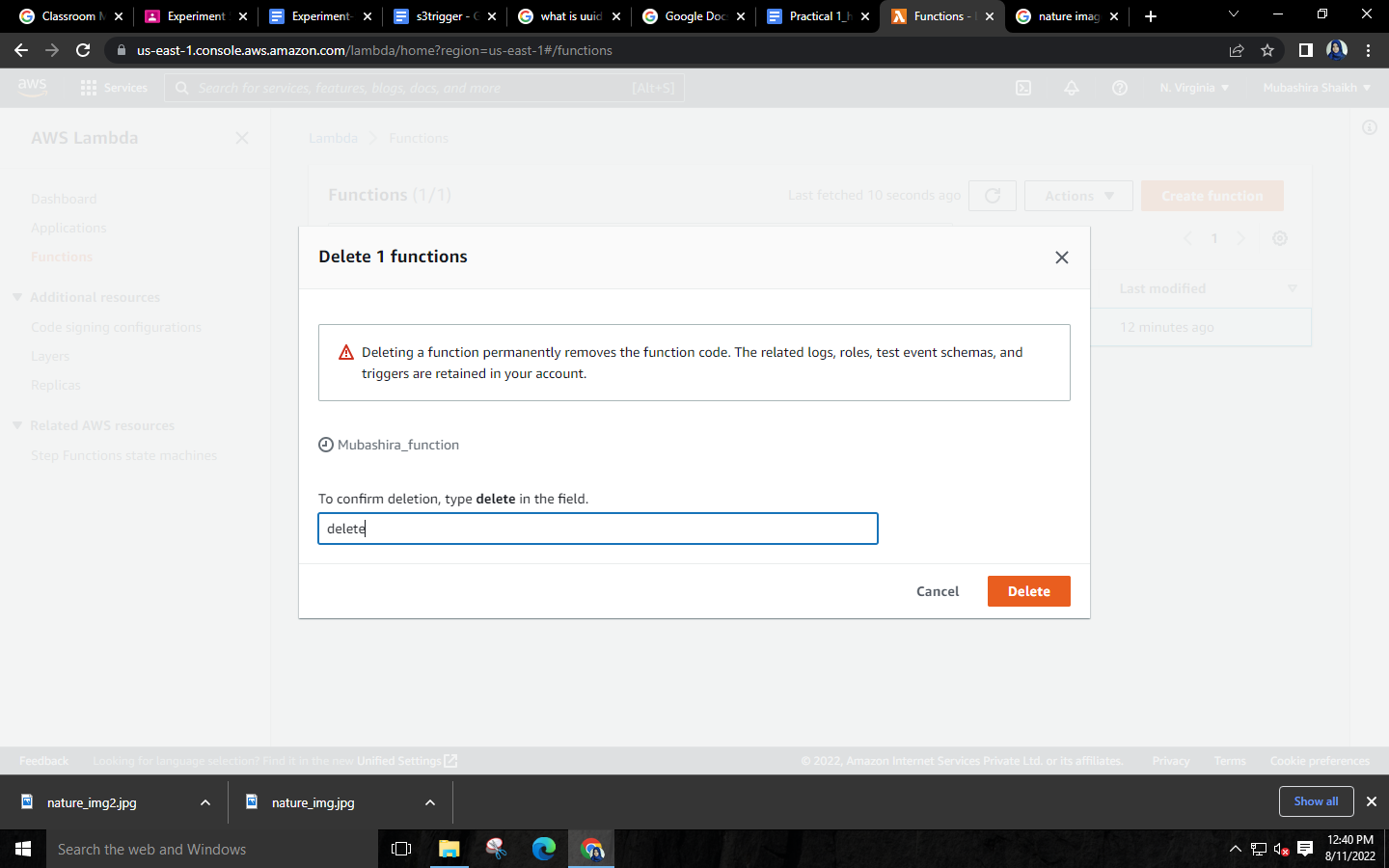
**Step 24: Delete the table**

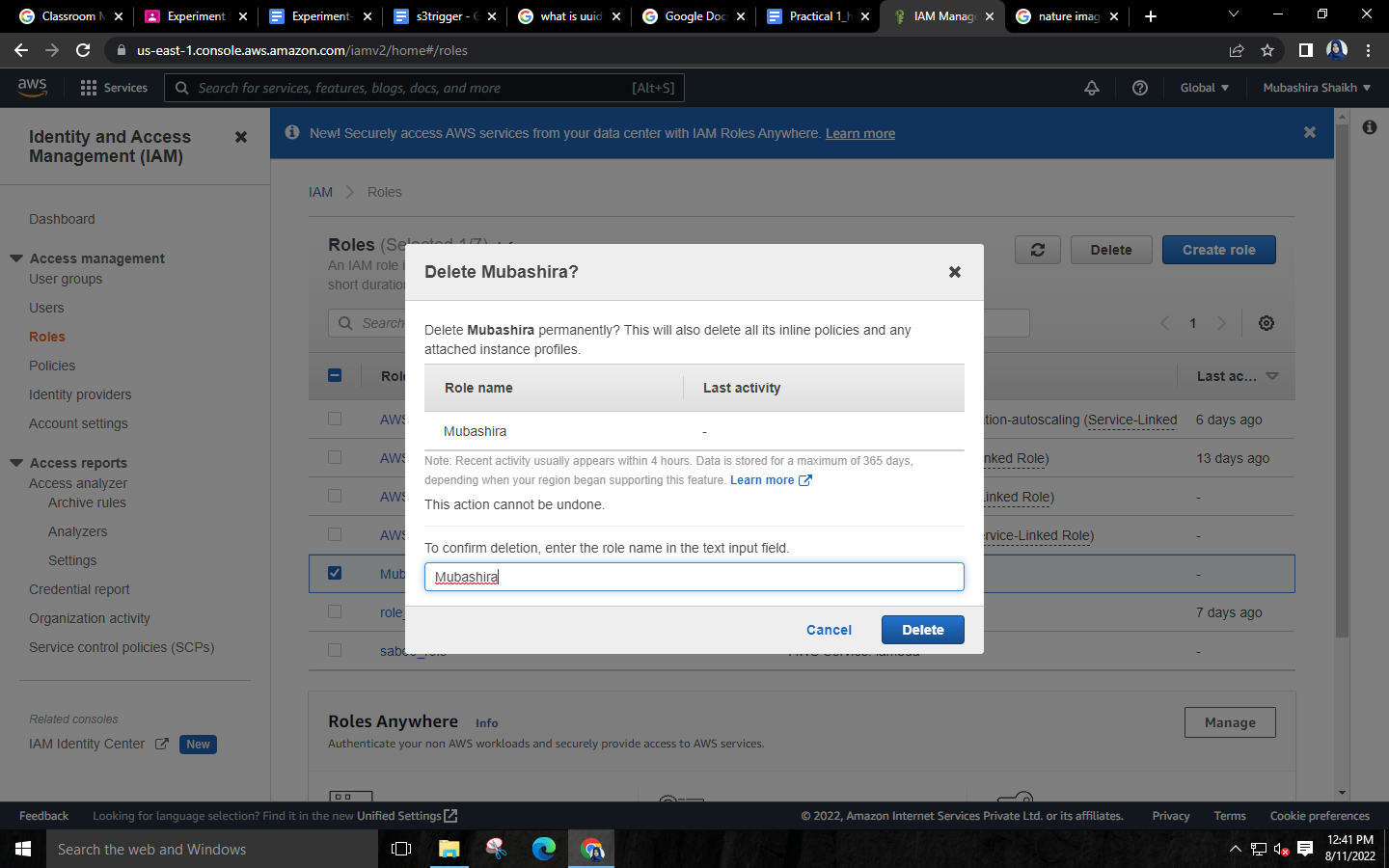
****

**Step 25: Empty your bucket🡪then delete it.**

****

****

**Step 26: Delete your lambda function**

**Delete the role**