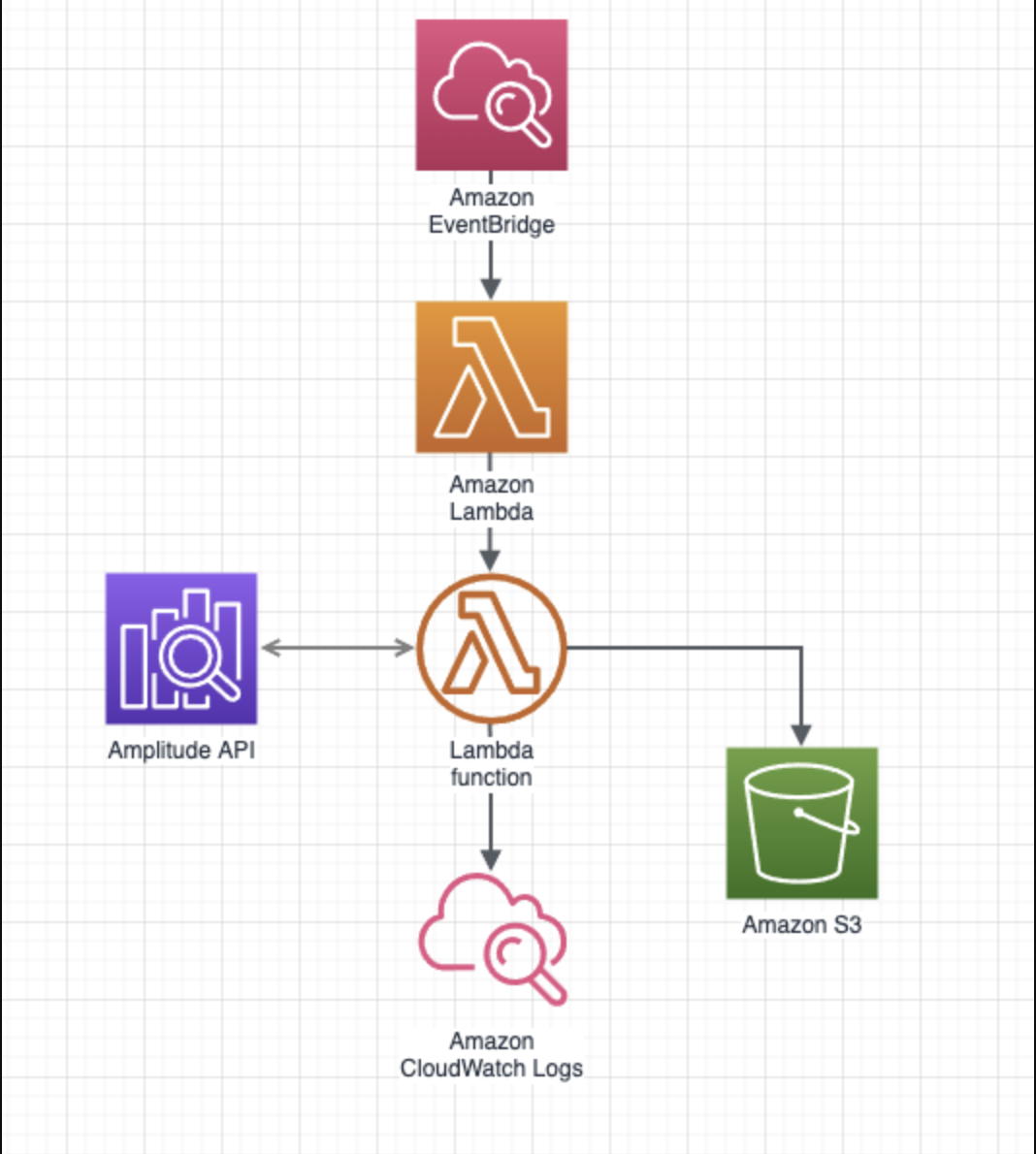
**What is Lambda?**

AWS Lambda is a serverless computing service provided by **Amazon Web Services** (AWS). It allows developers to run without provisioning or managing servers. AWS Lambda is a popular choice for building serverless applications, real-time data processing and event-driven architectures.It's event-driven, meaning your code runs in response to events, such as changes in data in an Amazon S3 bucket, updates in a DynamoDB table, or HTTP requests from an API Gateway.

**Key Features:**

* **Serverless**: No need to manage servers or infrastructure.
* **Event-driven**:Lambda functions are triggered by events
* **Scalability**: Lambdaautomatically handle large workloads.
* **Cost-effective**: You only pay for the compute time consumed by your functions.

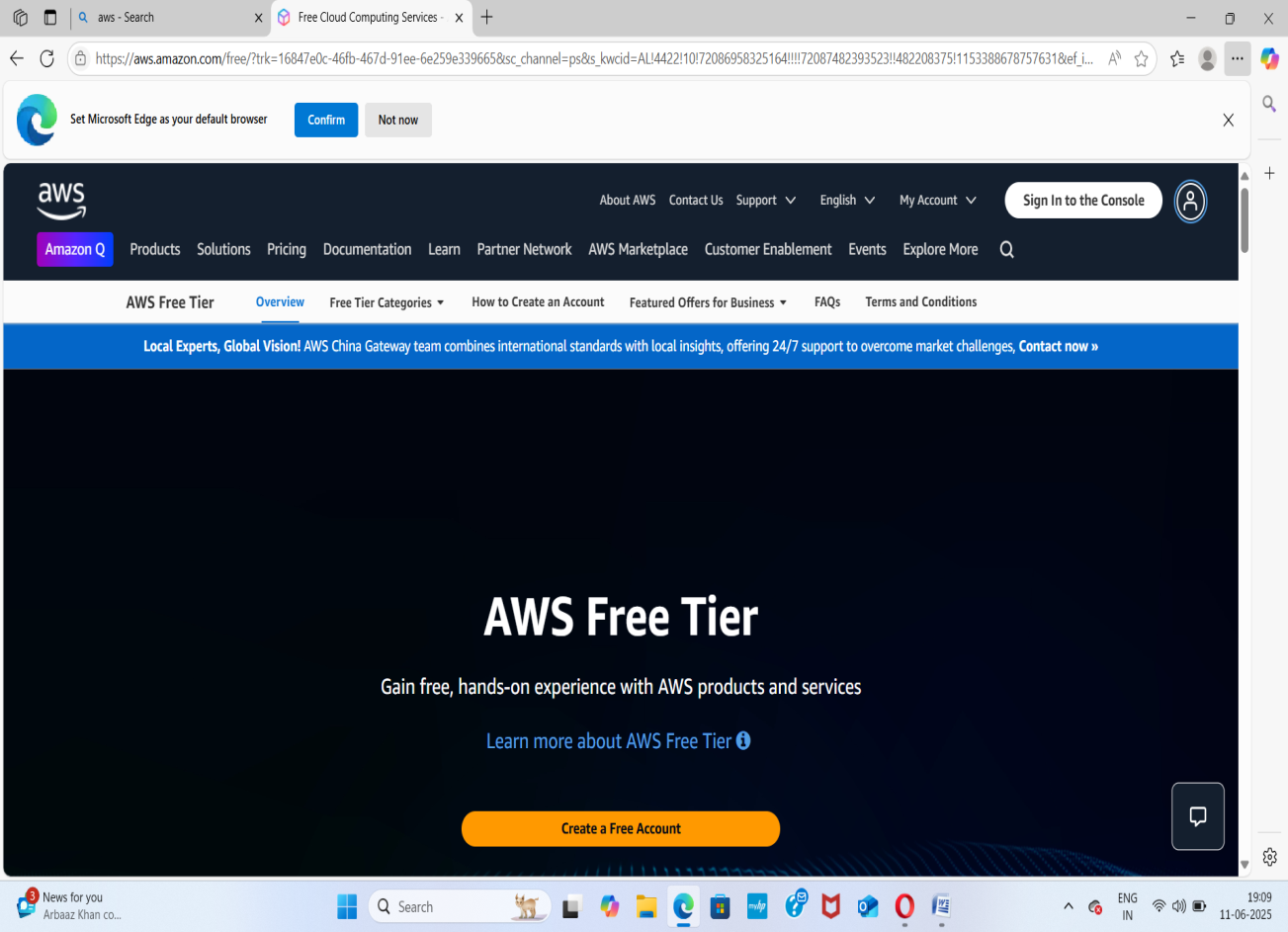


* **LAB TASK**

Step-by-Step to Creating and Using an AWS Lambda Function:

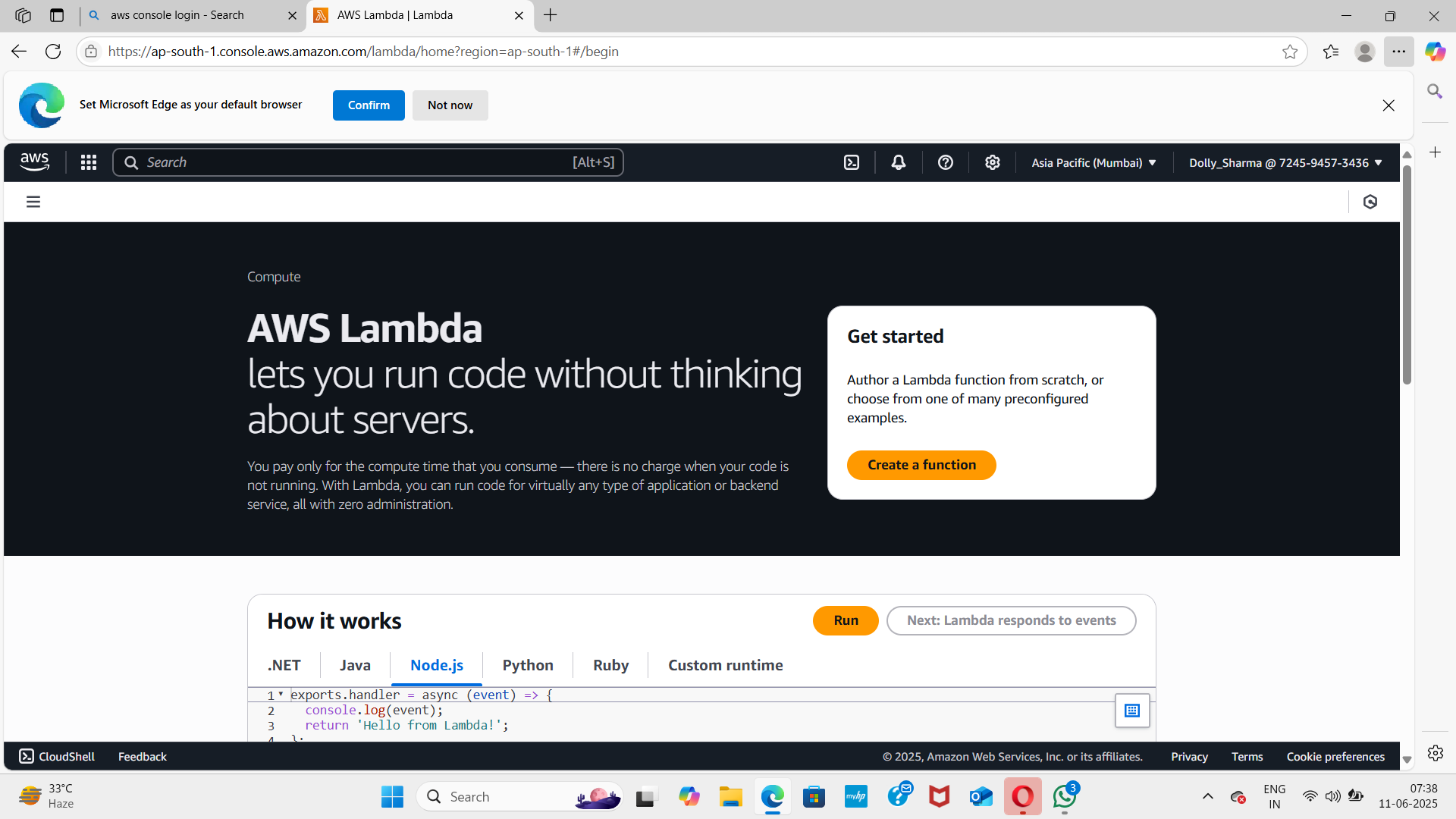
**Step 1: Set up your AWS Account**

* Sign in to the AWS Management Console: If you don't have an AWS account, create one.
* Choose a Region: Select an AWS Region **Asia pacific(Mumbai) ap-south-1**.

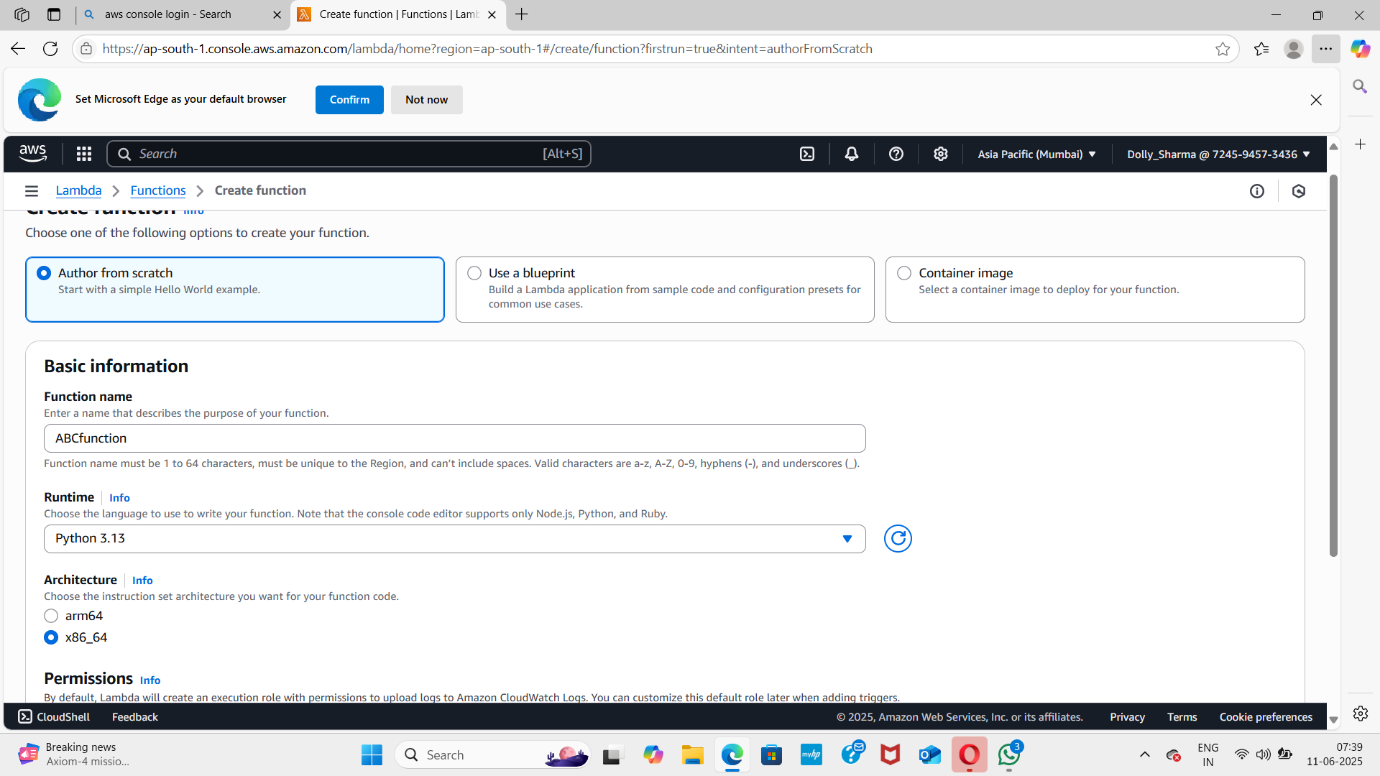


**Step 2: Create your Lambda Function**

* Navigate to Lambda: In the AWS Management Console, search for "Lambda" and open the service.
* Click "Create function".



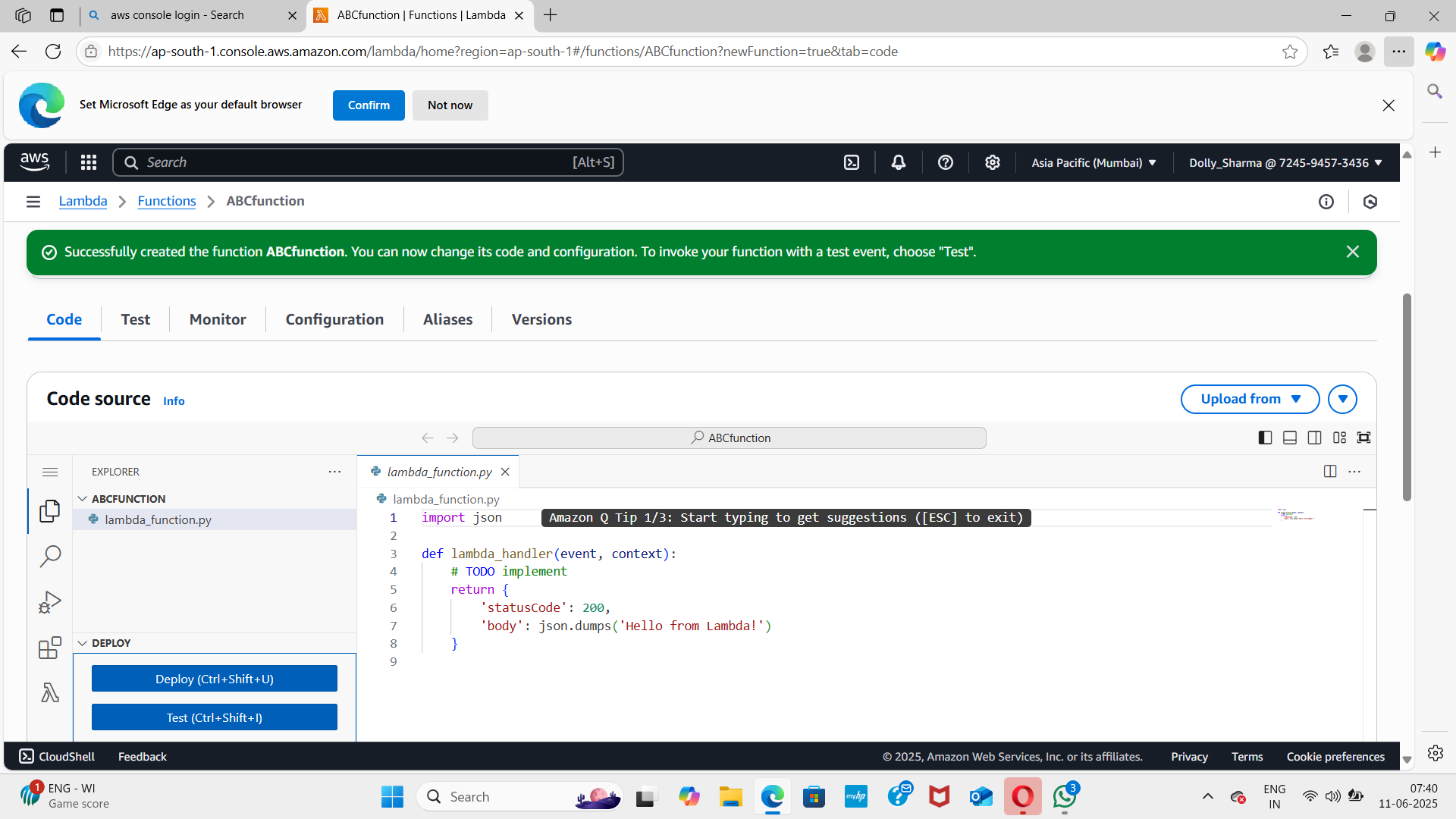
* Author from scratch: This is the most common option.
* Function name: Enter a unique and descriptive name (e.g., MyHelloWorldFunction).
* Runtime: Choose your preferred language ( Python 3.13).
* Architecture: Leave it as default (x86\_64) unless you have a specific reason to choose arm64.



**Step 3: Write Your Lambda Function Code**

Once the function is created, you'll be redirected to its configuration page.

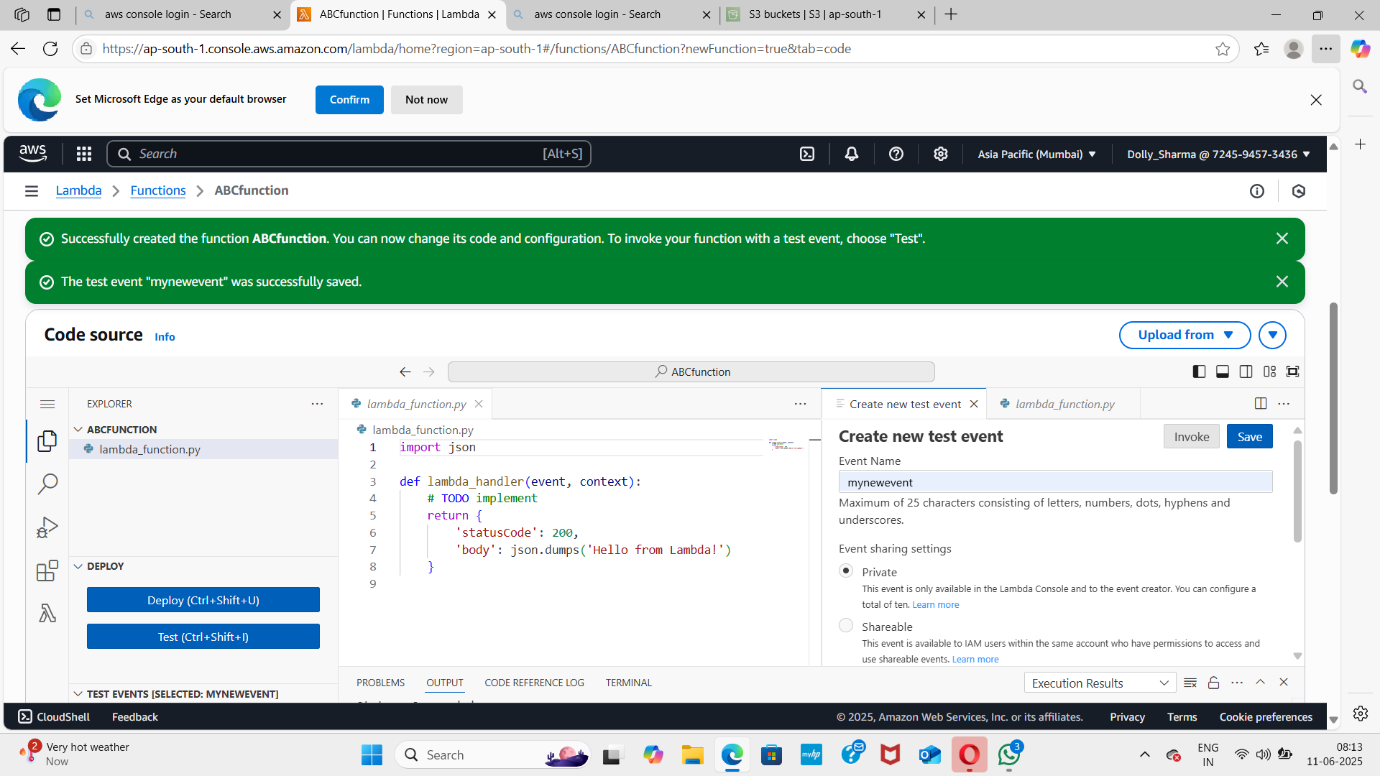
* Code source: In the Code tab, you'll see an inline code editor.
* For Python, the default code might look like this:



* Click "Deploy" to save your code.

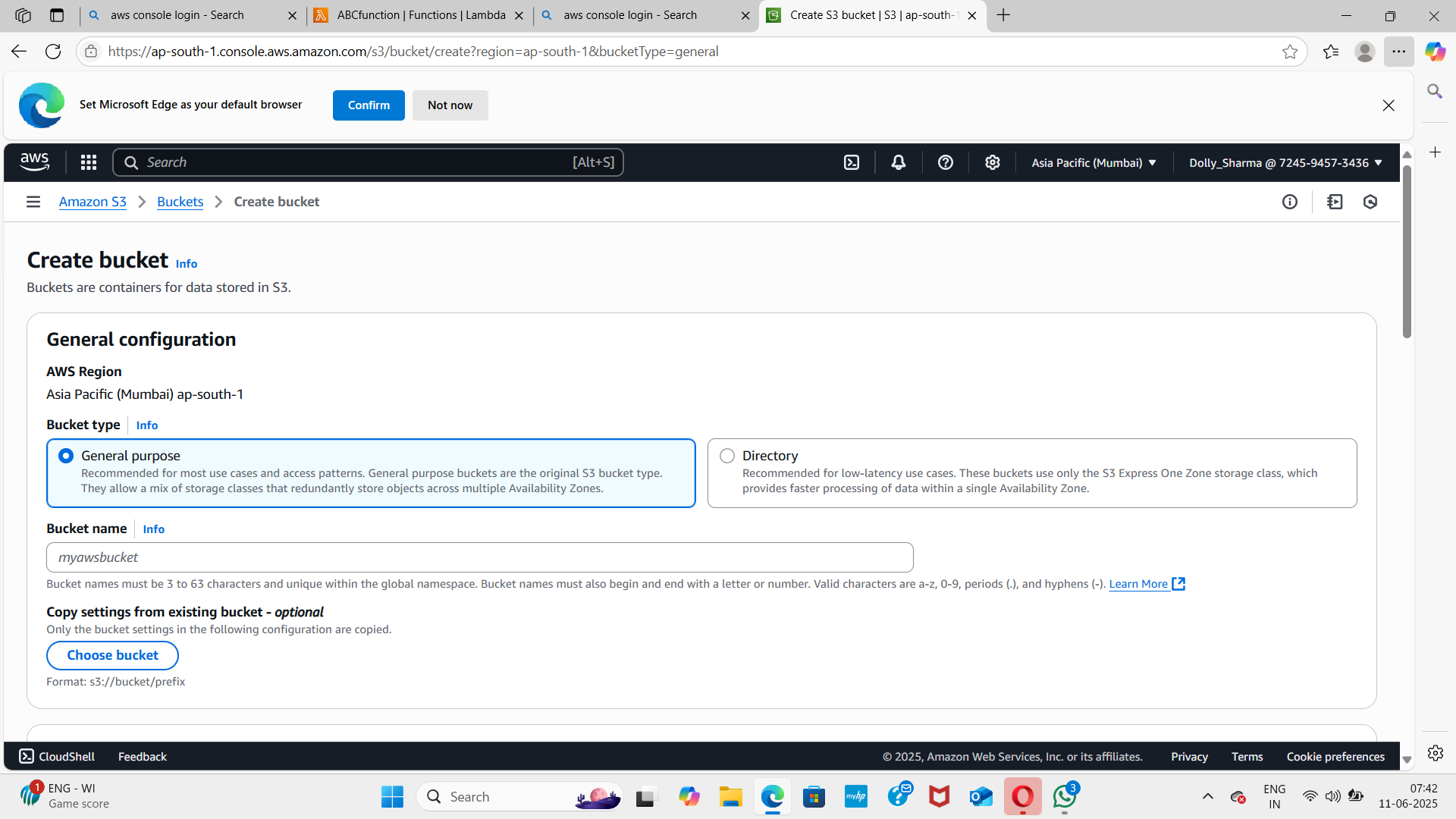
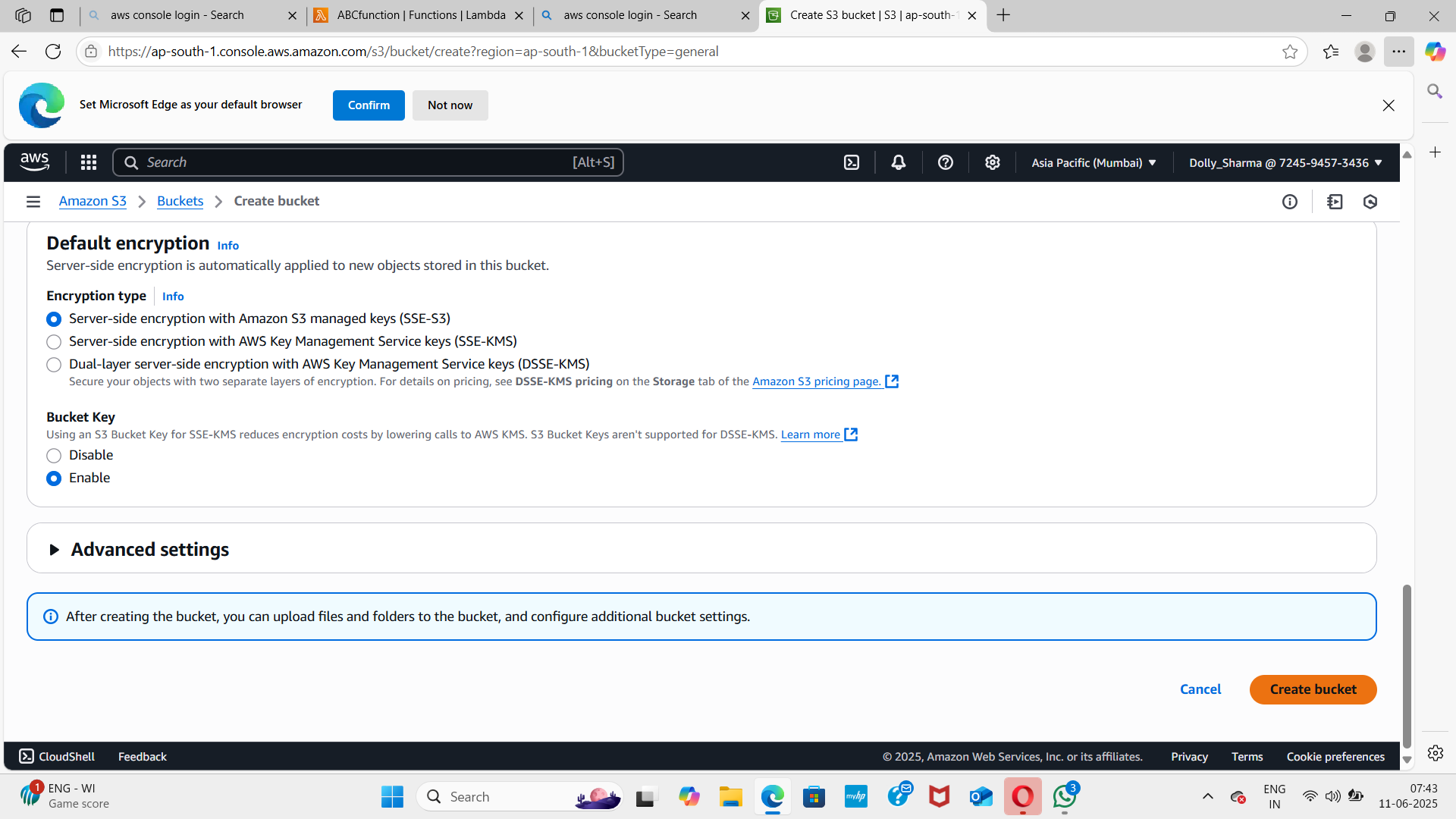
**Step 4: Test Your Lambda Function**

* + Test button: On the function's configuration page, click the Test button.
  + Configure test event:
  + Event name: Give your test event a name (e.g., MyTestEvent).
  + Event JSON: This is the input data for your function. You can create a simple JSON payload.
  + Click Save.
  + Invoke: Click Test again to run your function with the configured event.
  + Review results:
  + Execution results: You'll see the statusCode and body returned by your function.
  + Log output: This section displays any print statements from your function (these are also sent to CloudWatch Logs).

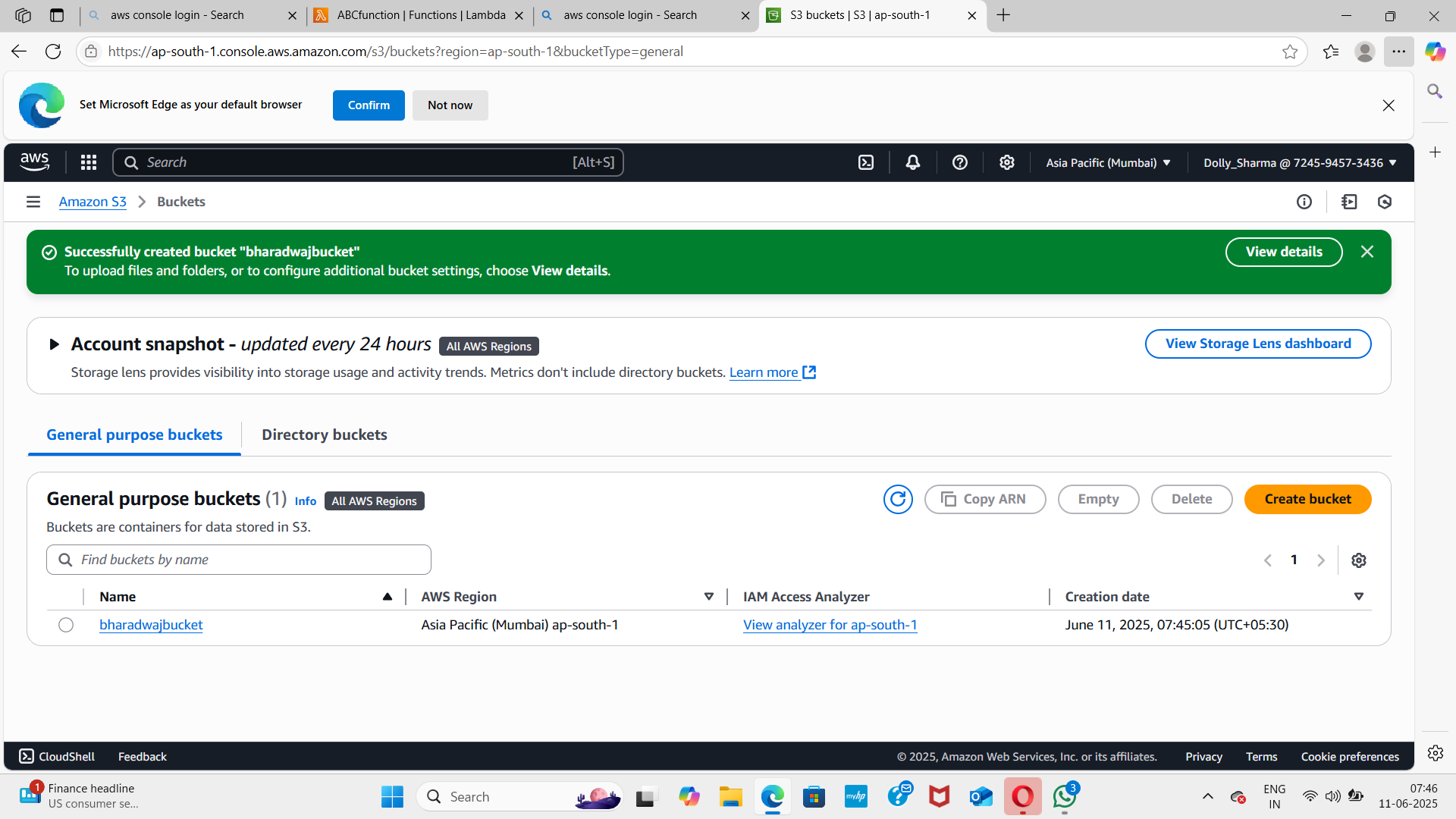
****

**Step 5: Configure a Trigger (e.g., S3)**

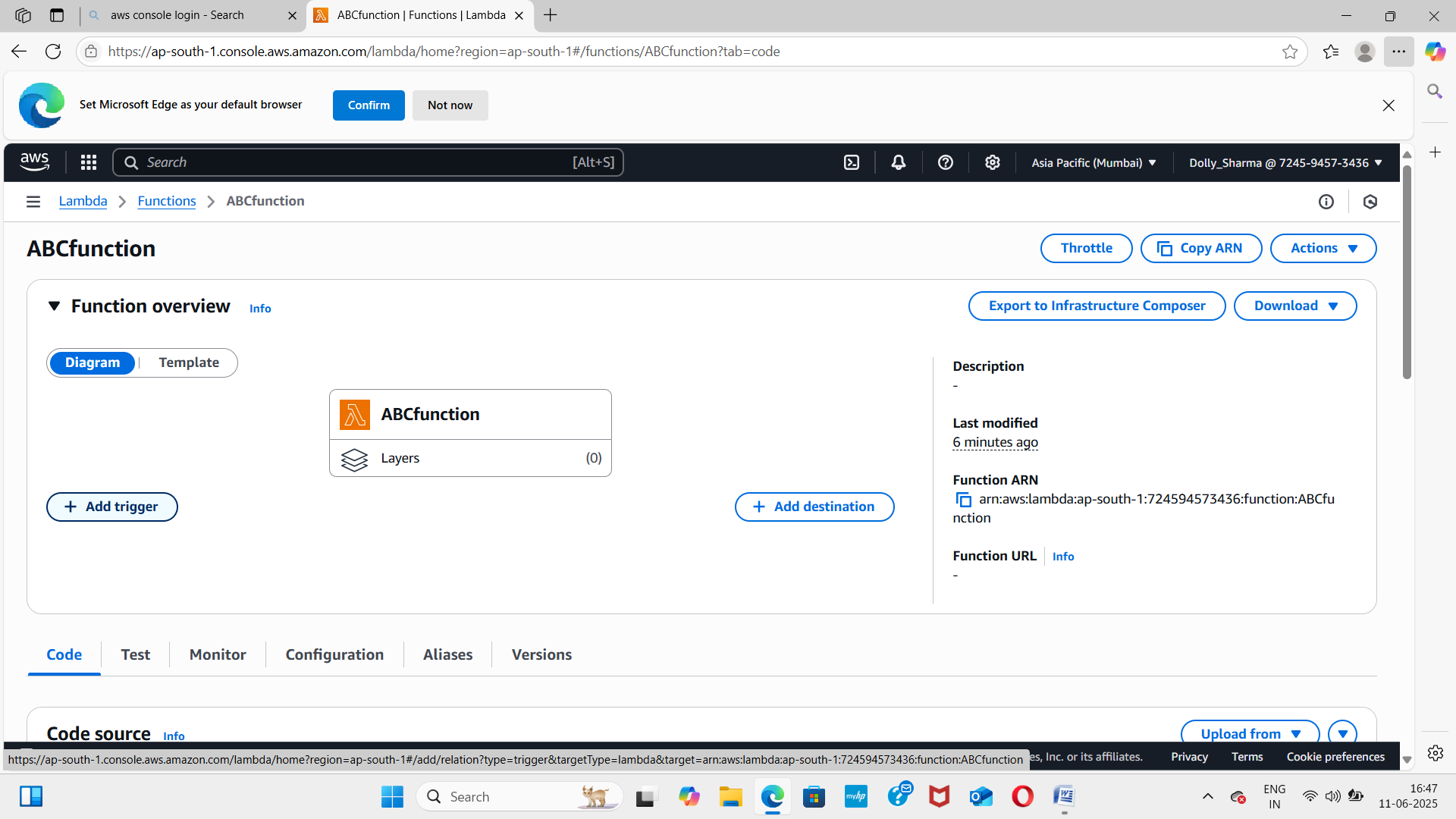
* Navigate to **AWS S3** by clicking on **Services** in the top left corner.S3 available under **Storage**.
* In the S3 dashboard, click on the **Create Bucket** button and fill in the bucket details.
* Bucket Name: Enter the bucket name(eg..,myawsbucket).
* Region: Select Asia pacific(Mumbai) ap-south-1.

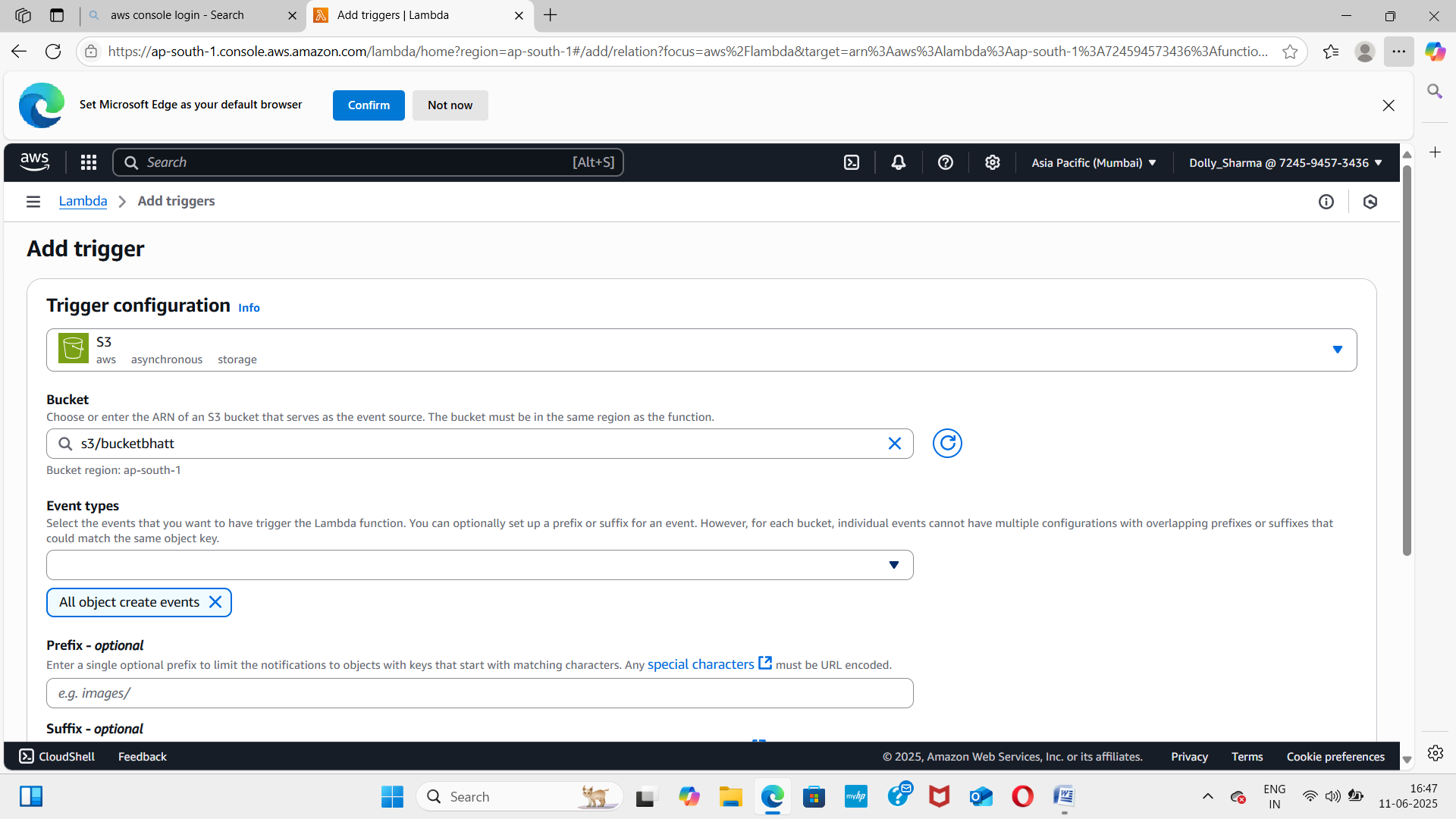


* Upload any file in the bucket by clicking on the bucket name.

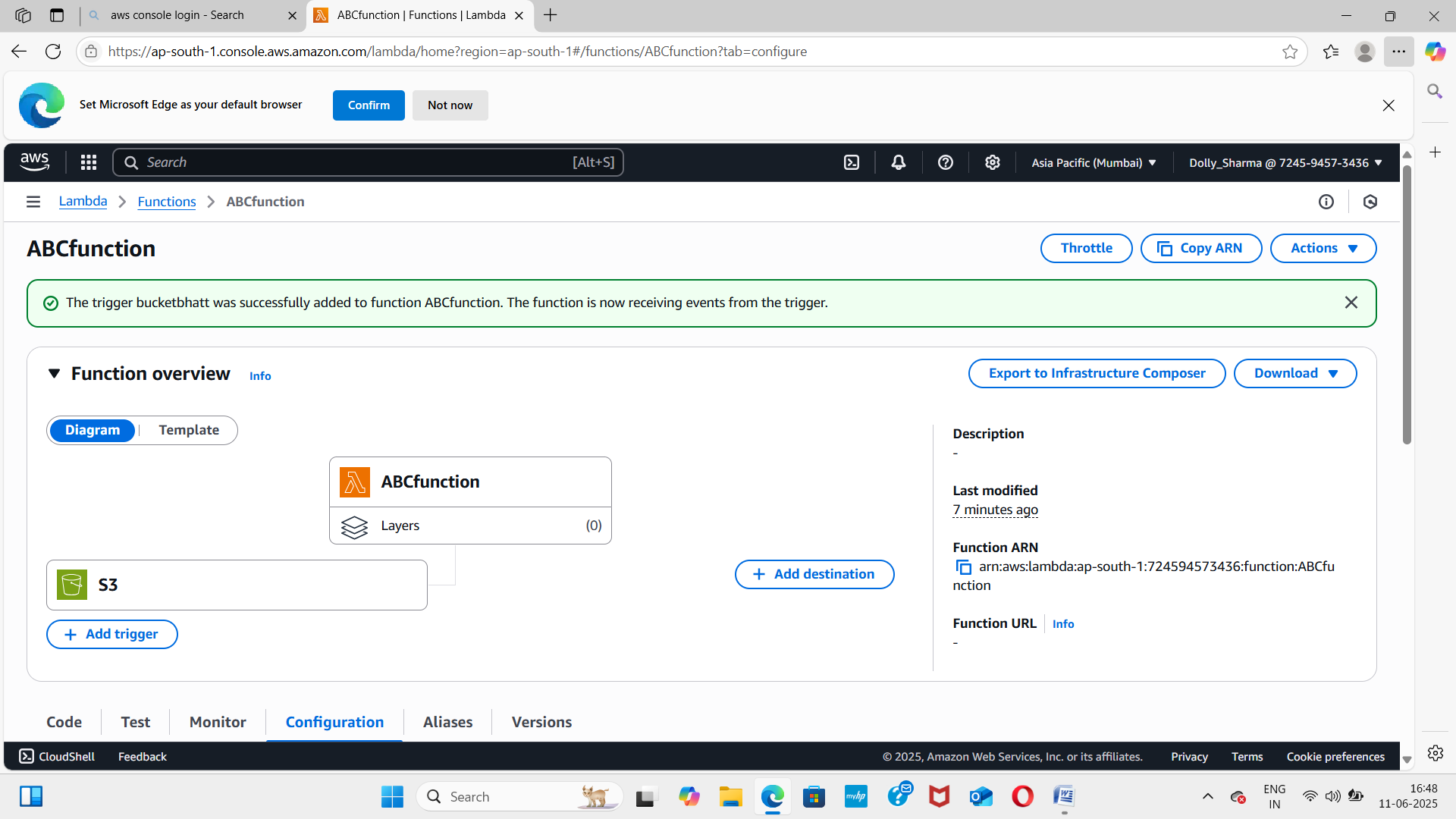


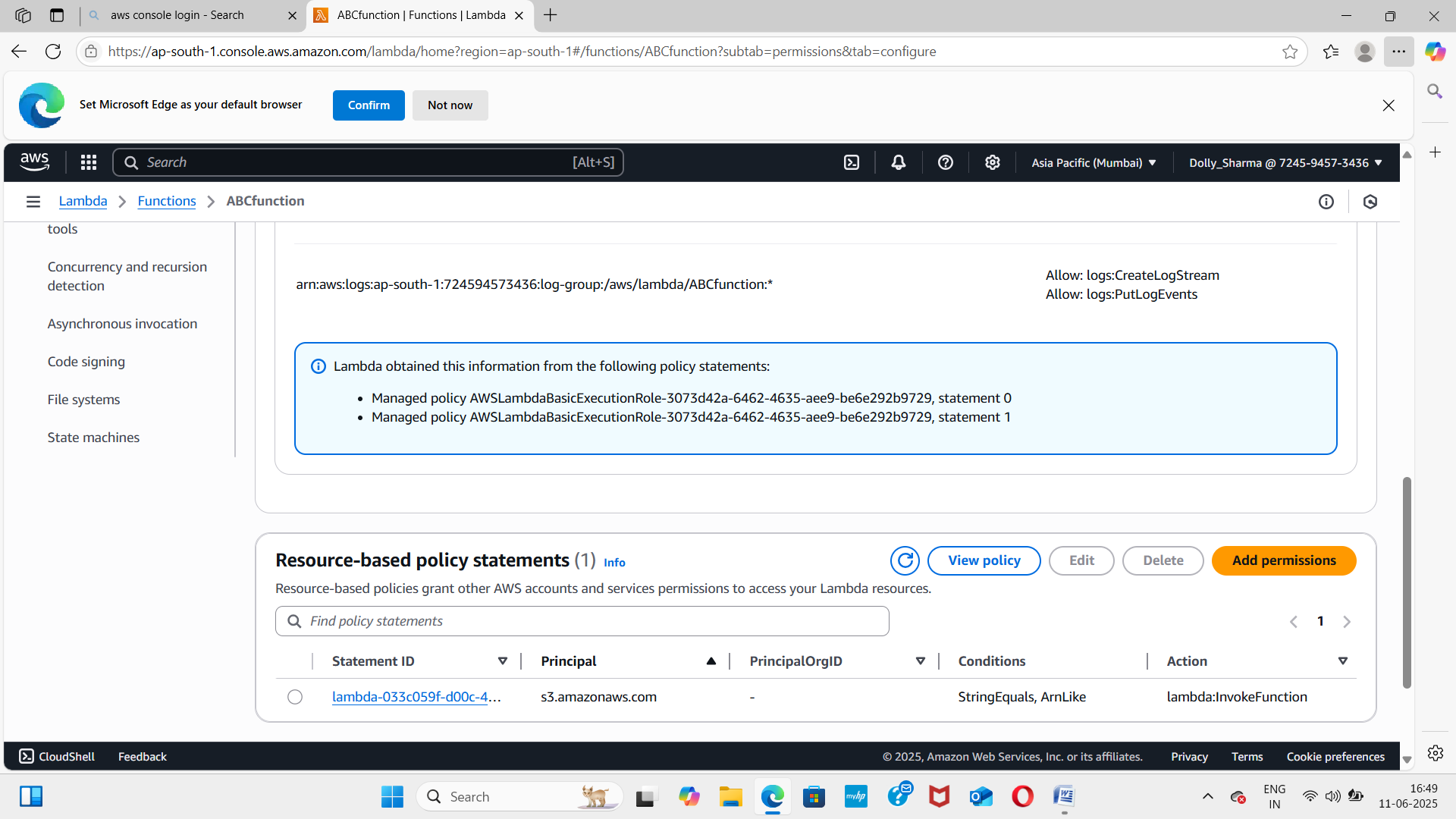
* Click on the **Add Trigger** button.



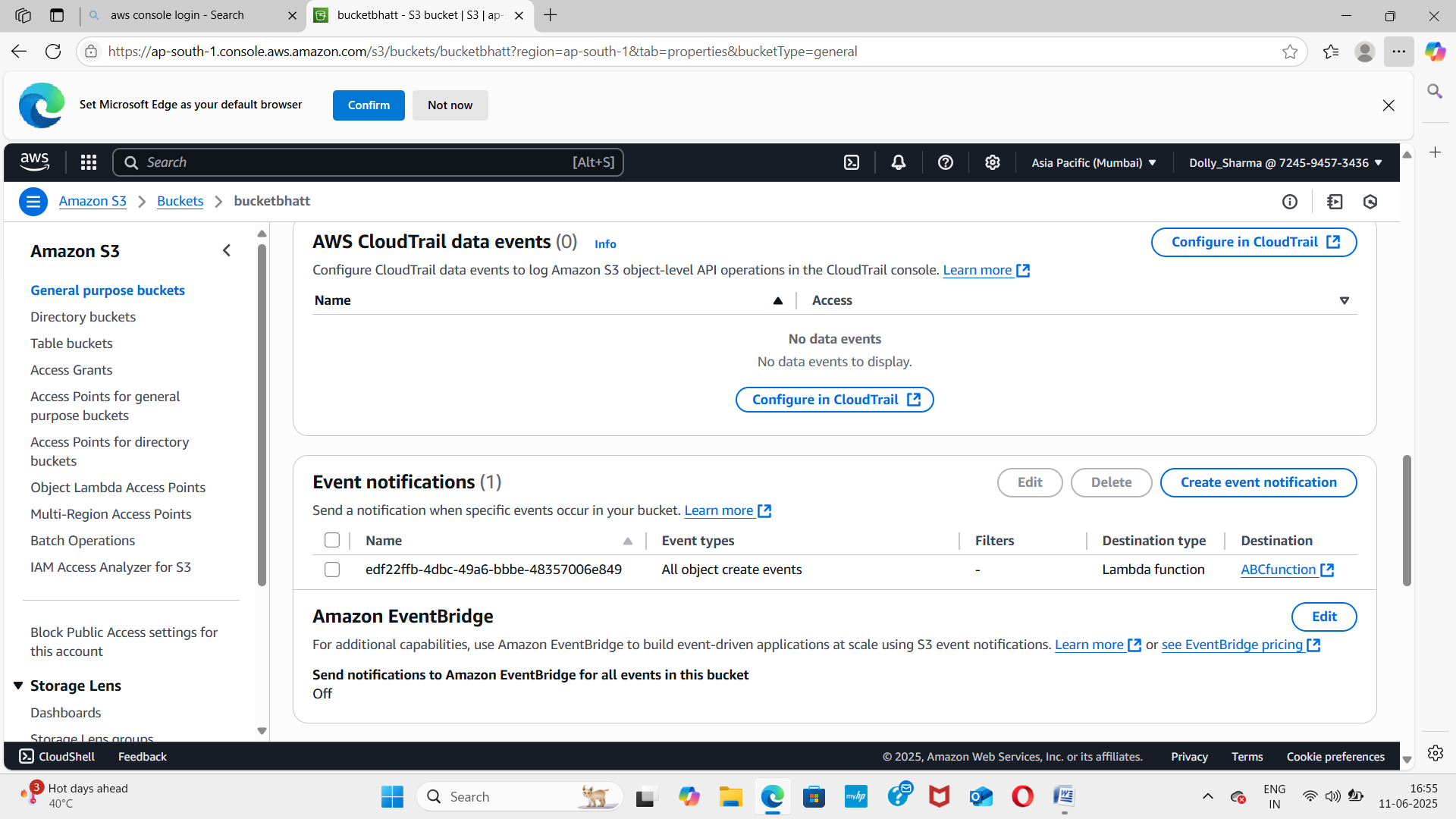


* Scroll down and go to **configuration** and then click on **permissions**.





* There one resource policy is already created.
* You can check by clicking on the bucket name and then click on properties.
* Event is already created.



* You can check this by uploading .zip file in code and then run that code.
* In the last step delete the upload file and then delete bucket after that delete the trigger and lastly delete the function and and then sign out from your aws account.