**Serverless Calculator**

**Services:**

**1)Amplify:**

* Creating an index.html page from scratch in a text editor.
* Deploying and hosting a web page with AWS Amplify.
* Invoking the API Gateway endpoint from the index.html page in Amplify.
* Re-deploying our index.html page using Amplify.

**2)Lambda:**

* Creating a Python Lambda function to implement our math functionality.
* Configure the Test events in the lambda function.
* Testing our Lambda function .

**3)API Gateway:**

* Creating a REST API for our Lambda function using API Gateway.
* To persist our math result to a database, and also handle permissions.
* Updating the index.html page to call API Gateway.

**4)DynamoDB:**

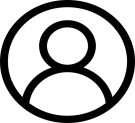
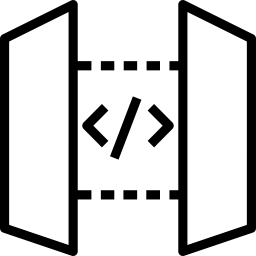
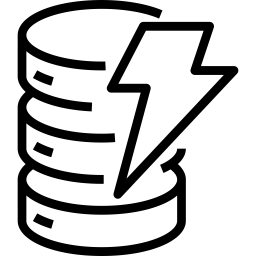
* Creating a new DynamoDB table to store our math result.
* Giving Lambda permission to write to the DynamoDB table.
* Updating the Lambda function code to write to the DynamoDB table.

**5)IAM:**

* By using IAM Roles, we attached DynamoDB with a lambda policy.

**Architecture**

**END USER AMPLIFY API GATEWAY LAMBDA DYNAMODB**

****    

 **IAM USER**

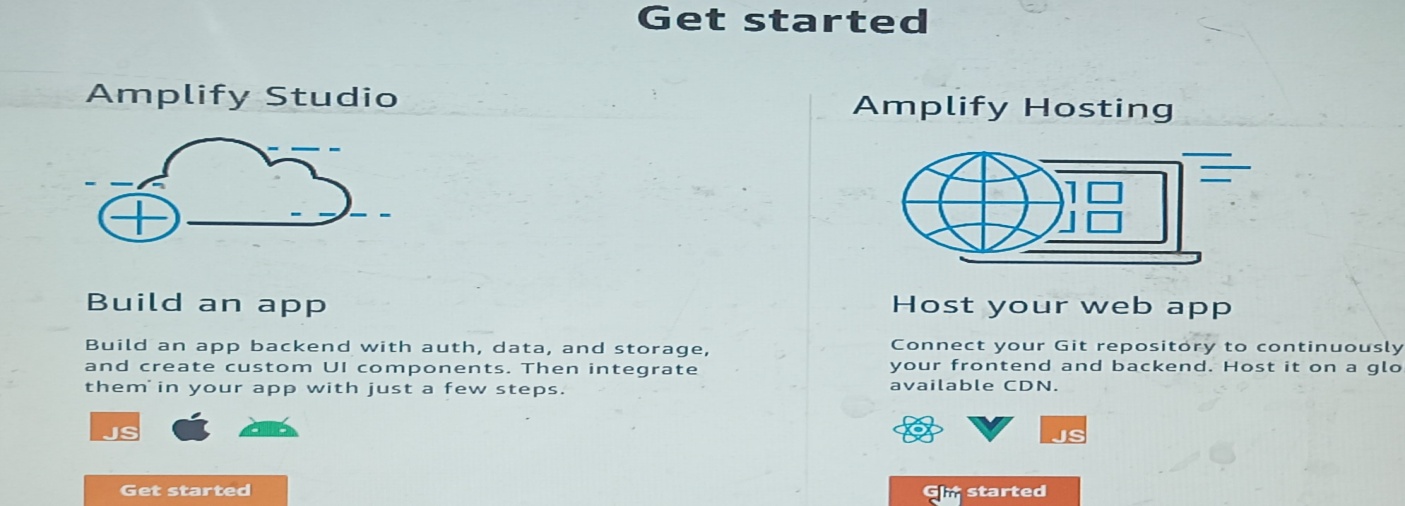
**Step-By-Step Procedure:**

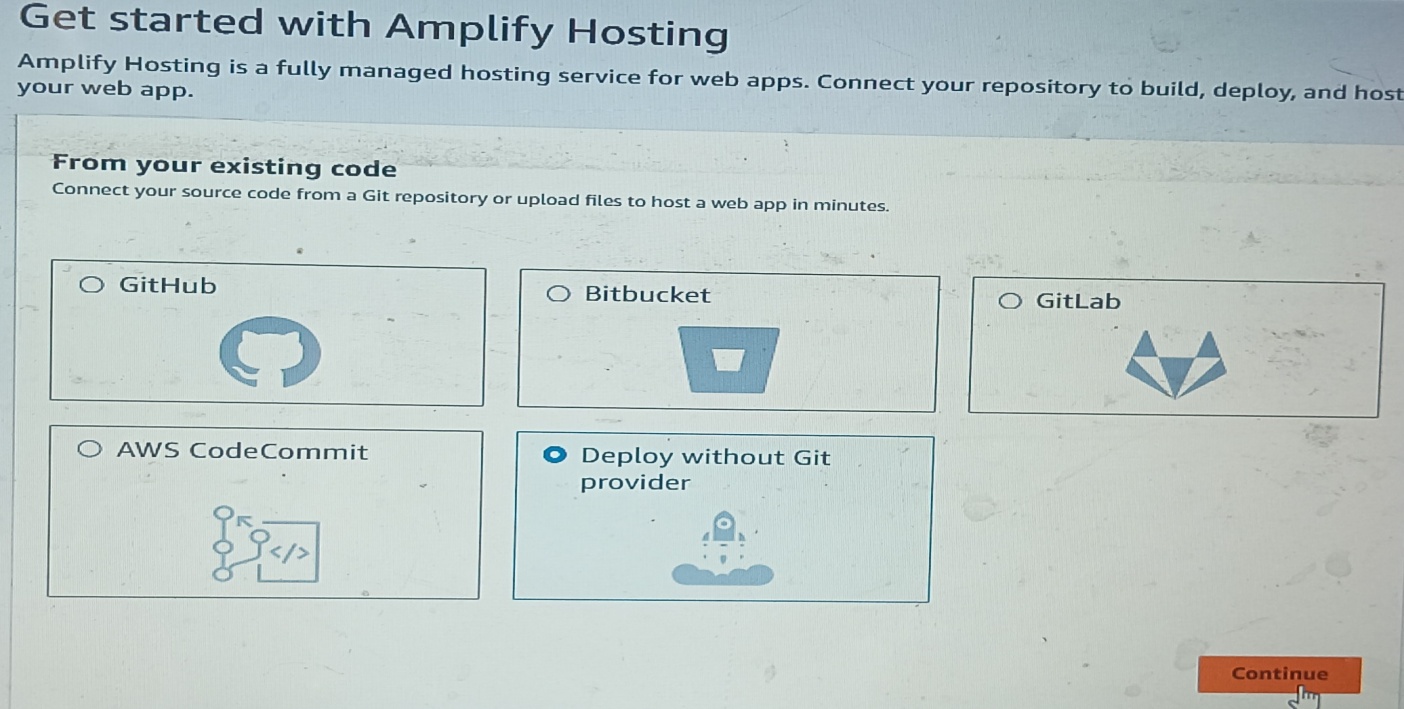
1)Creating an index.html page from scratch in a text editor.We’re basically just going to display ‘To the power of math’ on our browser for now. Next, save it and zip up the file**.(For code refer “source code” folder)**

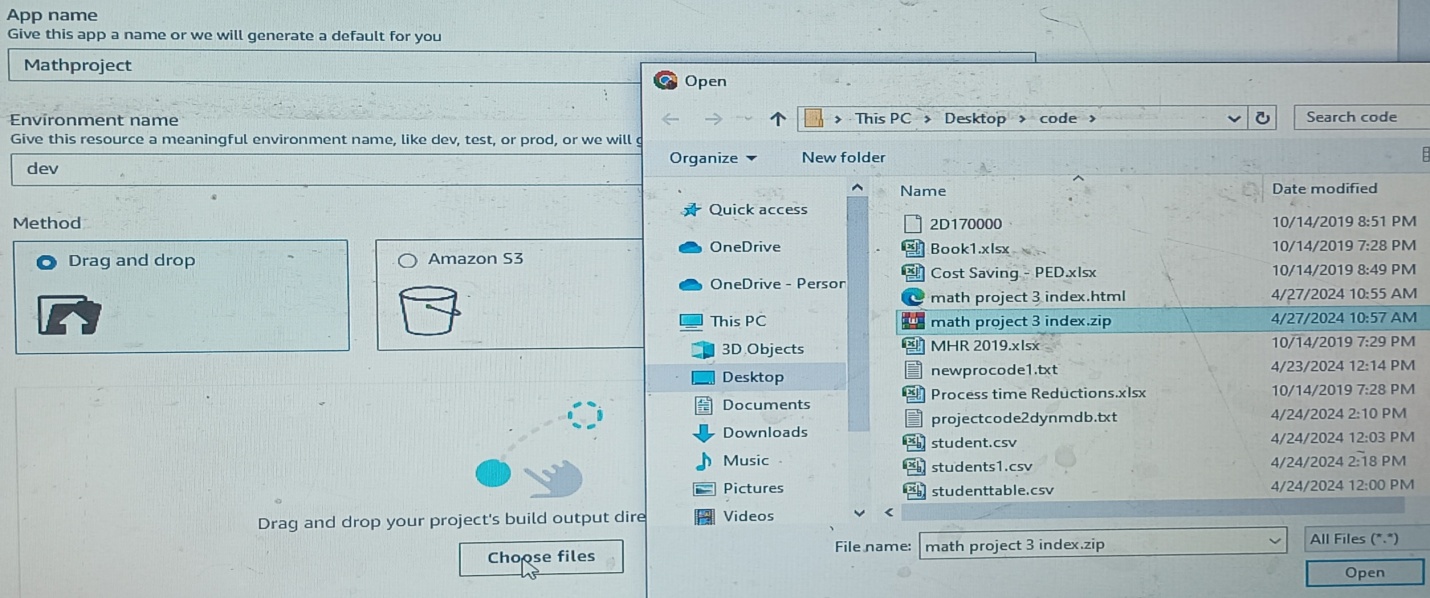
2) Open up the AWS Console and navigate to Amplify. We will be using Amplify to deploy and host our web page.

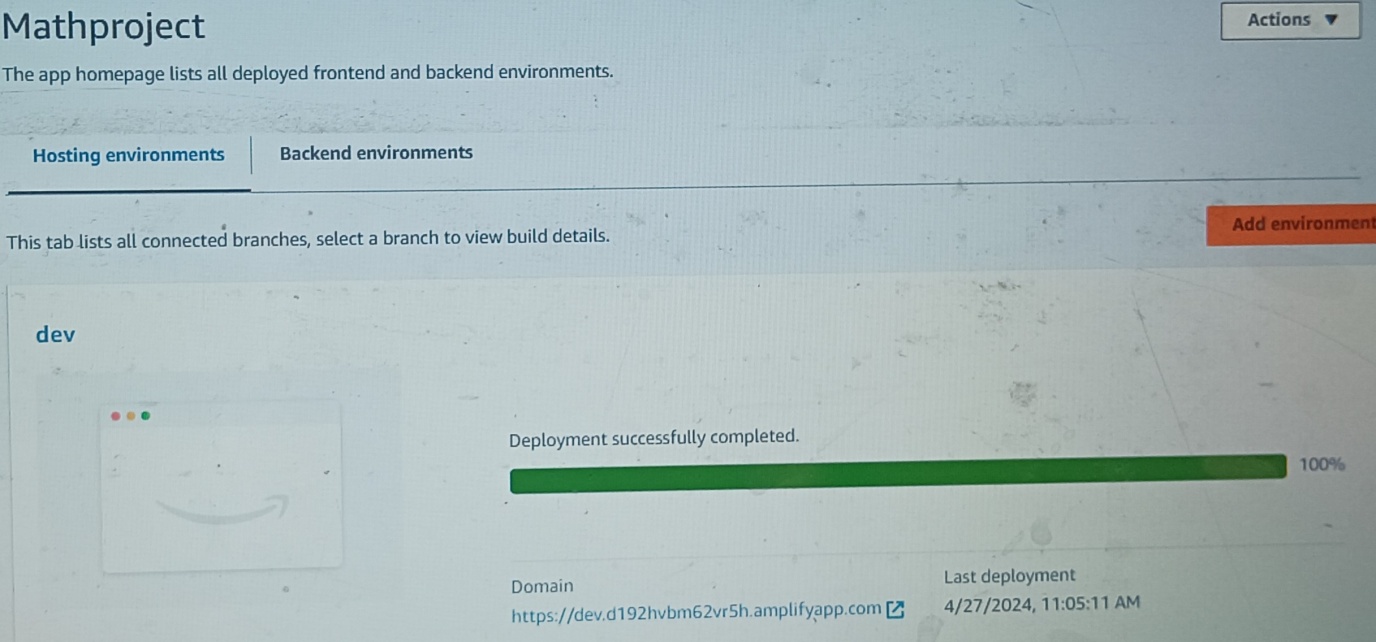
3) Click ‘Get started’ under Amplify Hosting.(Host your web App).

4) Click on “Deploy without Git provider”. Name the App Drag and drop the zip file you created earlier, then click save and deploy.









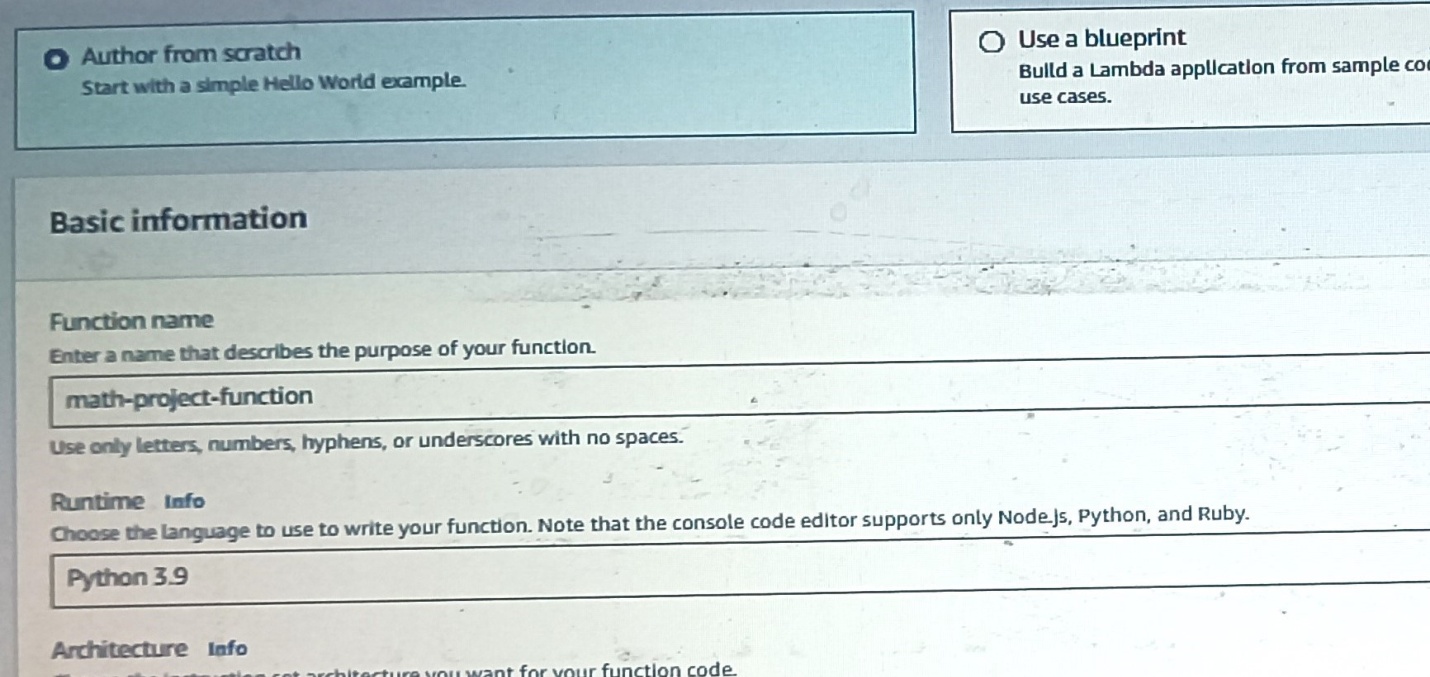
If you browse this domain “To The Power Of Math” will be displayed.

**A way to do some Math**

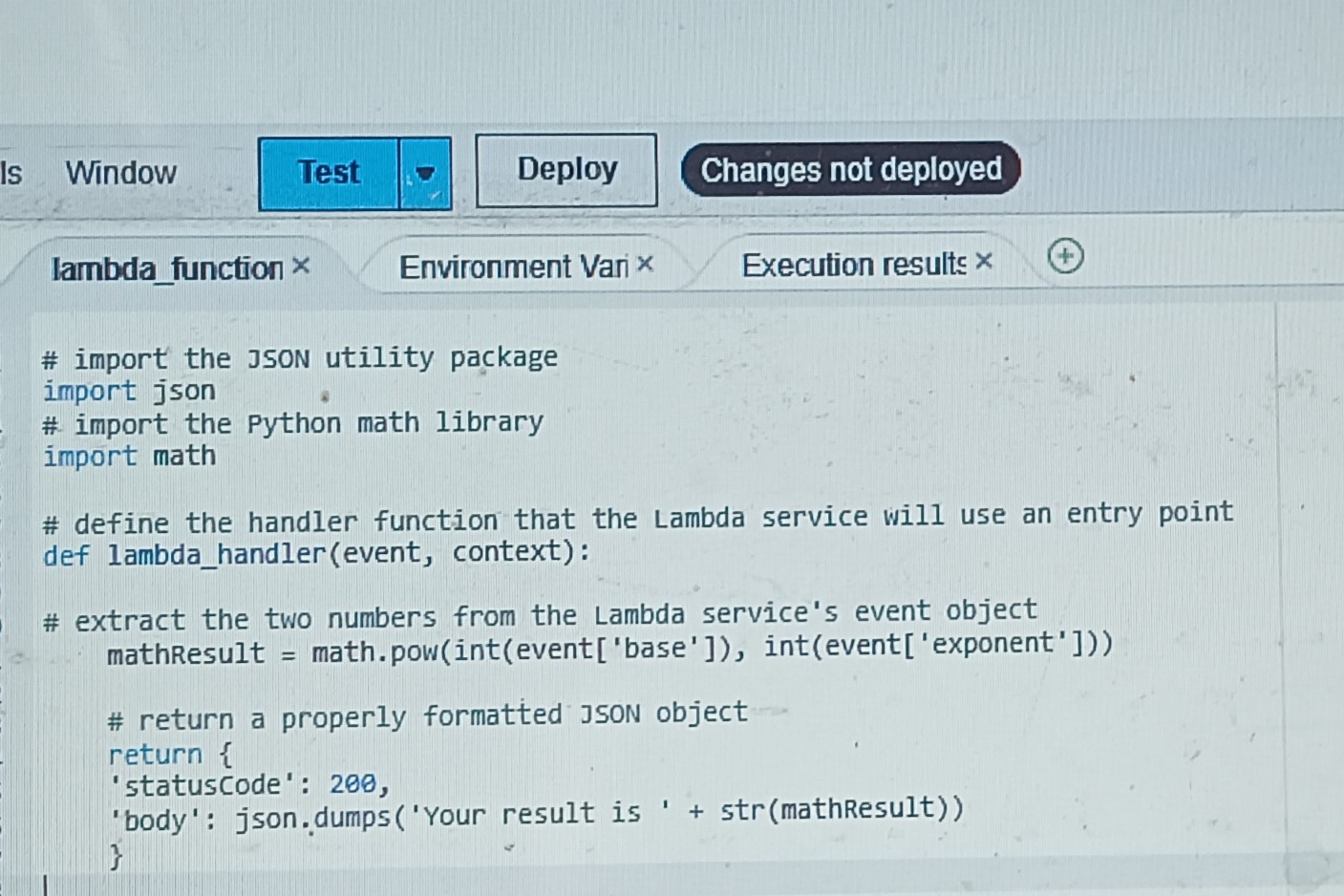
To achieve this, we would use a Lambda function. Lambda function is code that runs (serverlessly) and responds to some triggers.

1)Head over to Lambda and create a function using Runtime (Python 3.9).

2) Input the details as seen below.Select Author from scratch Name the function Run time Python 3.9 (Keep other things default)Create



3) Go to the function Scroll down to the code source under the code tab, and update the code. After inputting the code, save it (Ctrl+S) and most importantly, deploy it by clicking deploy at the top.



4) Next lets test to make sure this functions as we expect. Click Test > Configure test event Name the event.

5) Scroll down and edit ‘Event JSON’. Put in the values you want. Click save.

6)Then click Test [What we did earlier was to configure the test event, now we would run the test by clicking the Test button]

We should get a status code of “200” and verification that your result is “16” [depending on the values you entered].

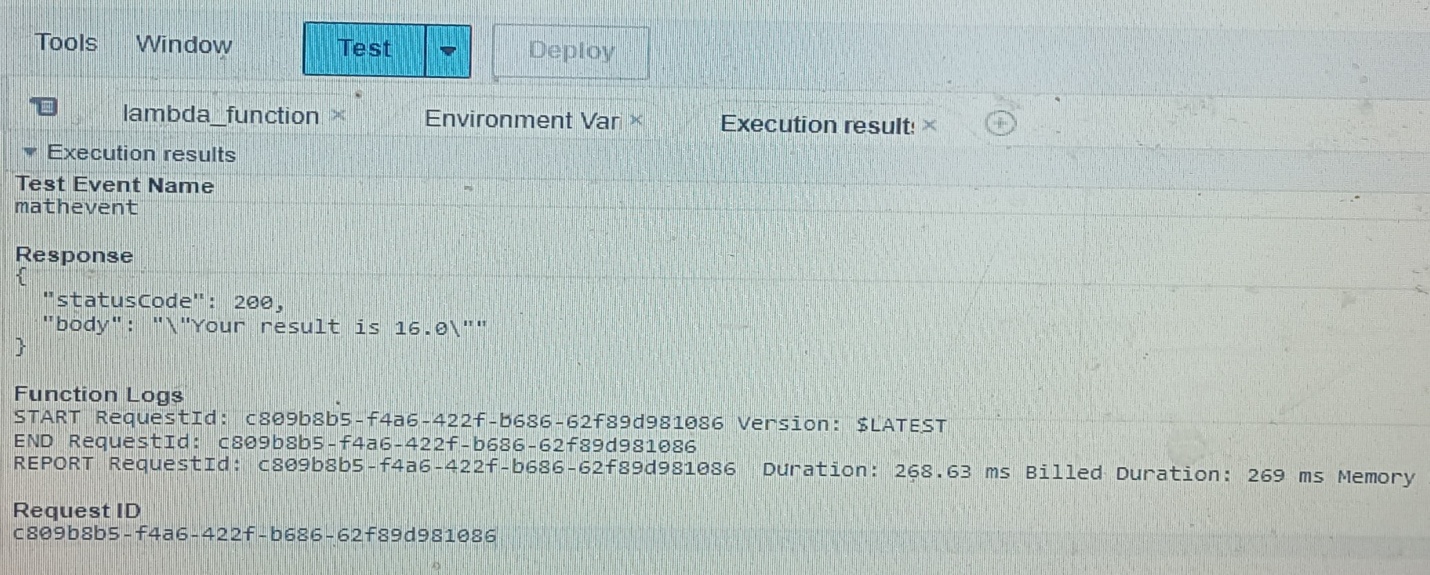


{

“base”: 2,

“exponent”: 4

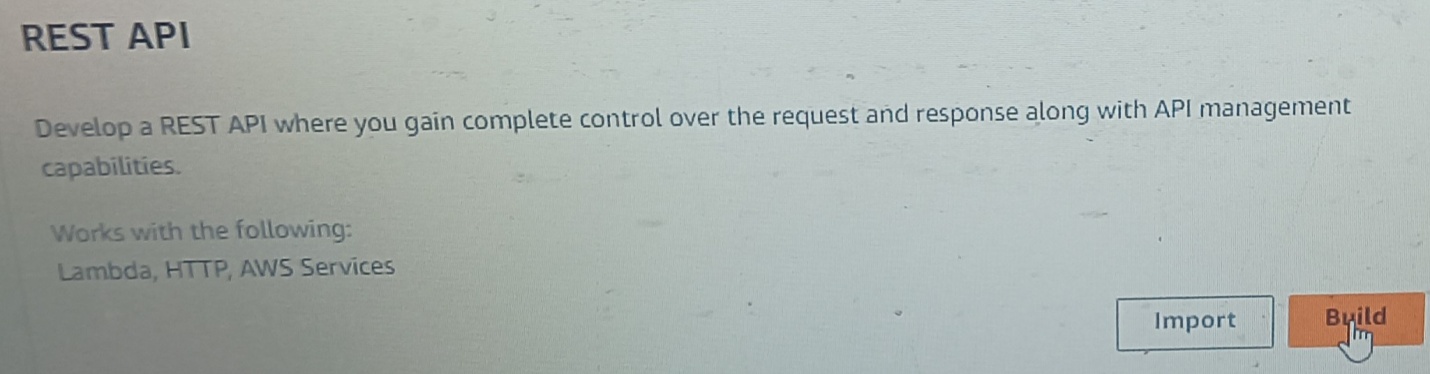
}

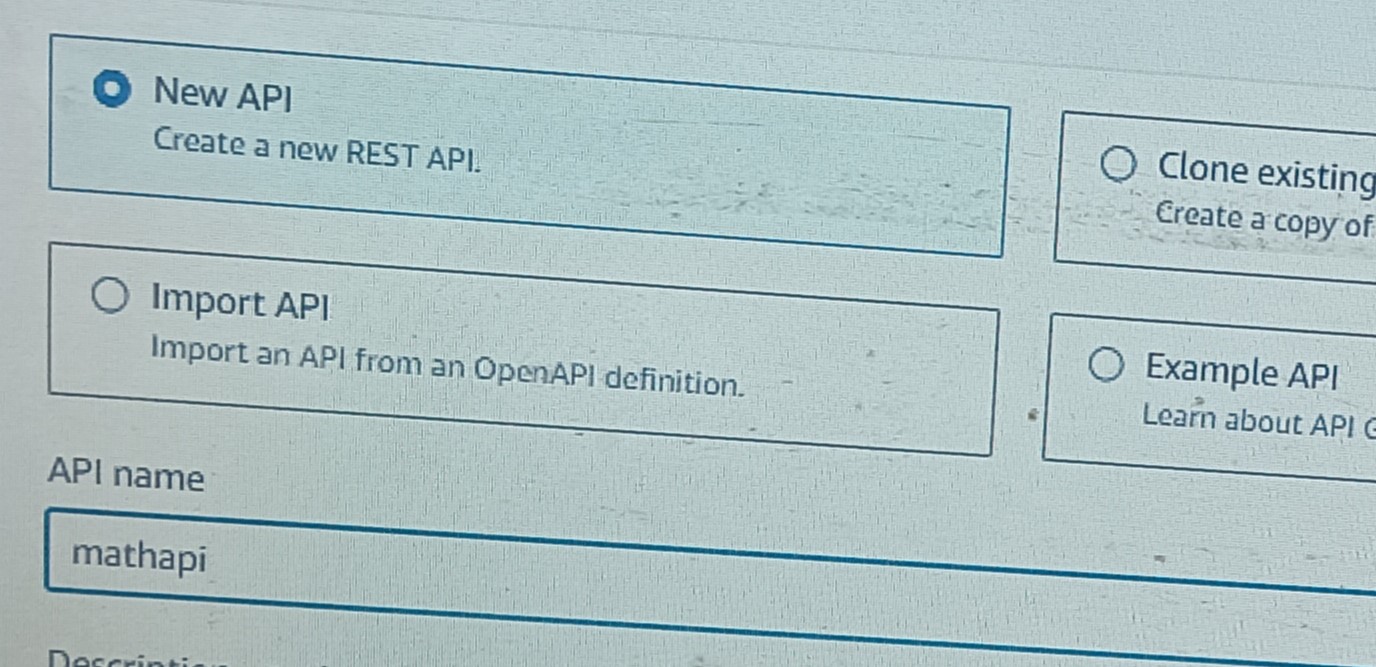


# ****A way to invoke the Math functionality****

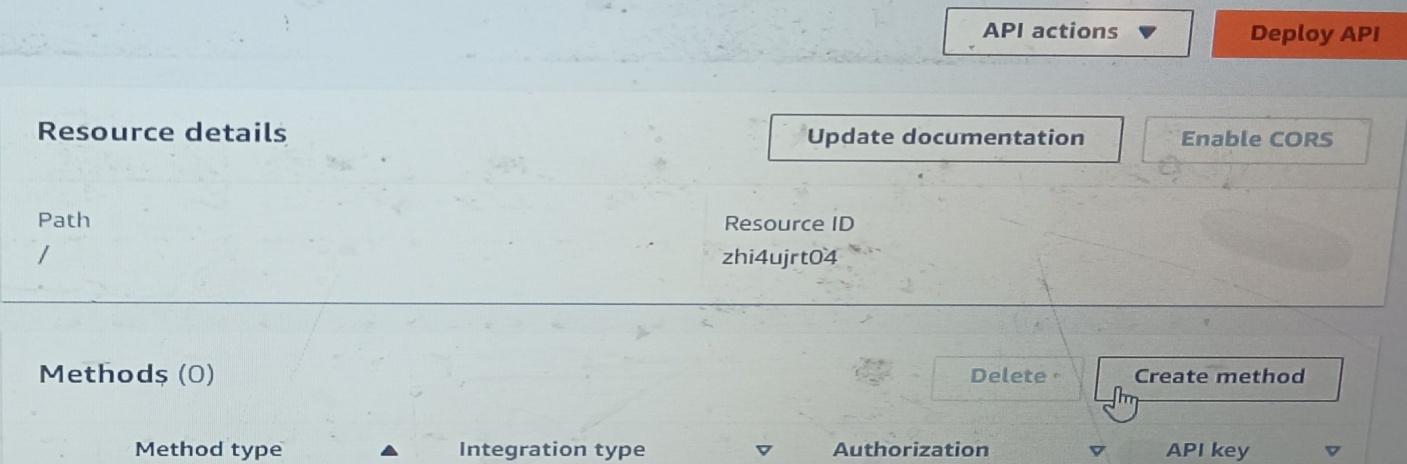
1) We would do this by invoking the Lambda function. So navigate to API Gateway console, we will be using a REST API.

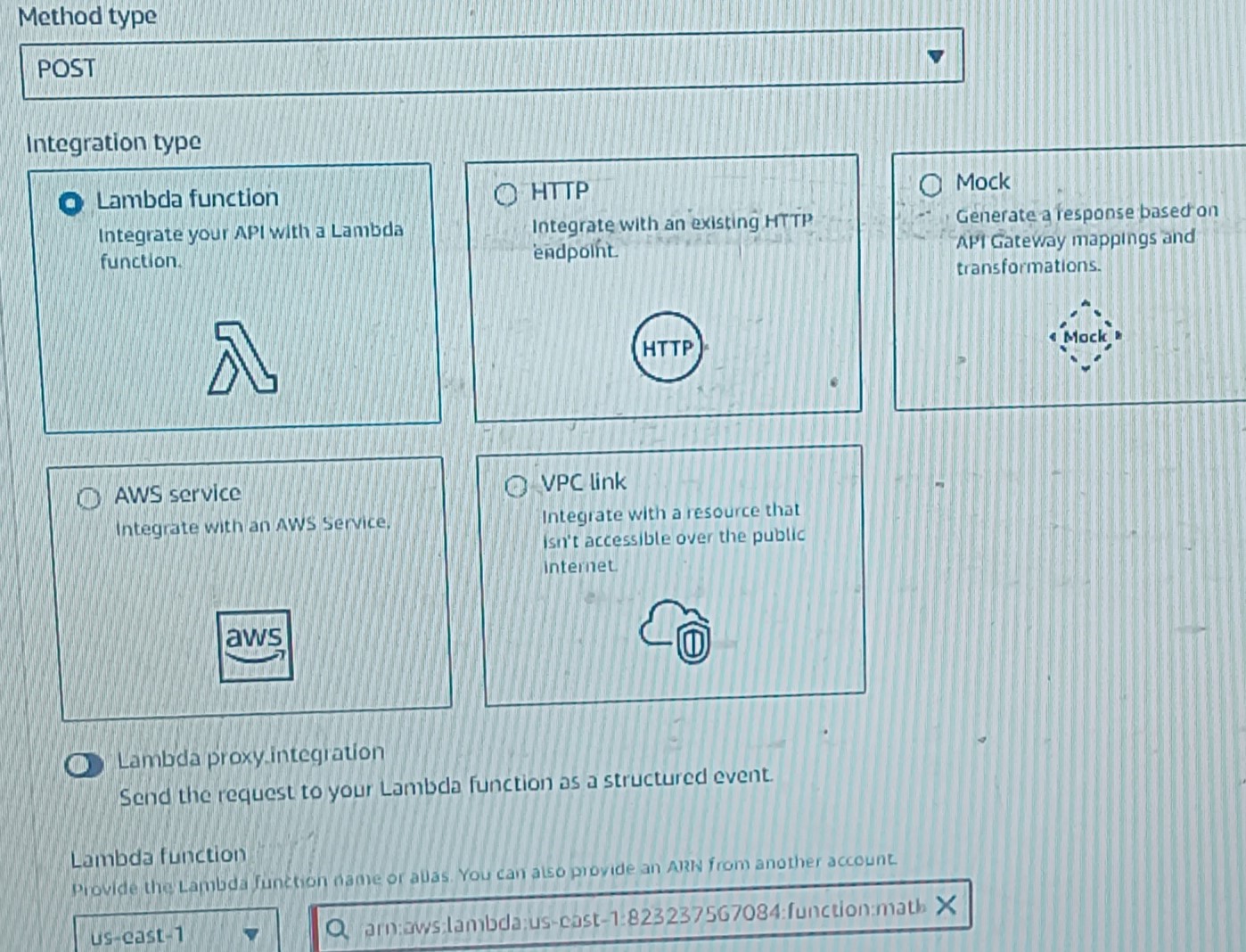
2)Click on RESTAPI New API Name the API keep everything default Create API.

****

****

3) Next create method. Method type should be ‘POST’. Integration type should be ‘Lambda function’ (Choose lambda function that ou have created). Create.

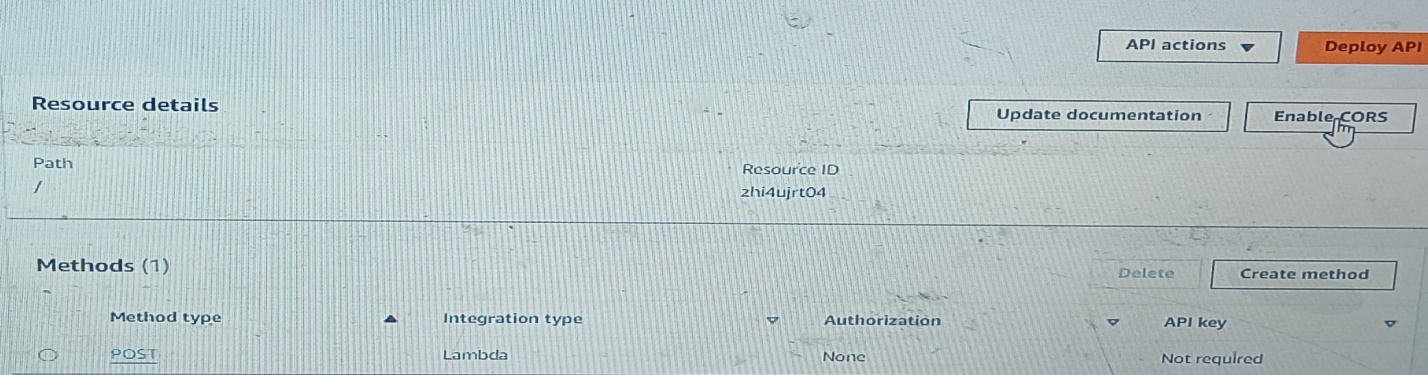


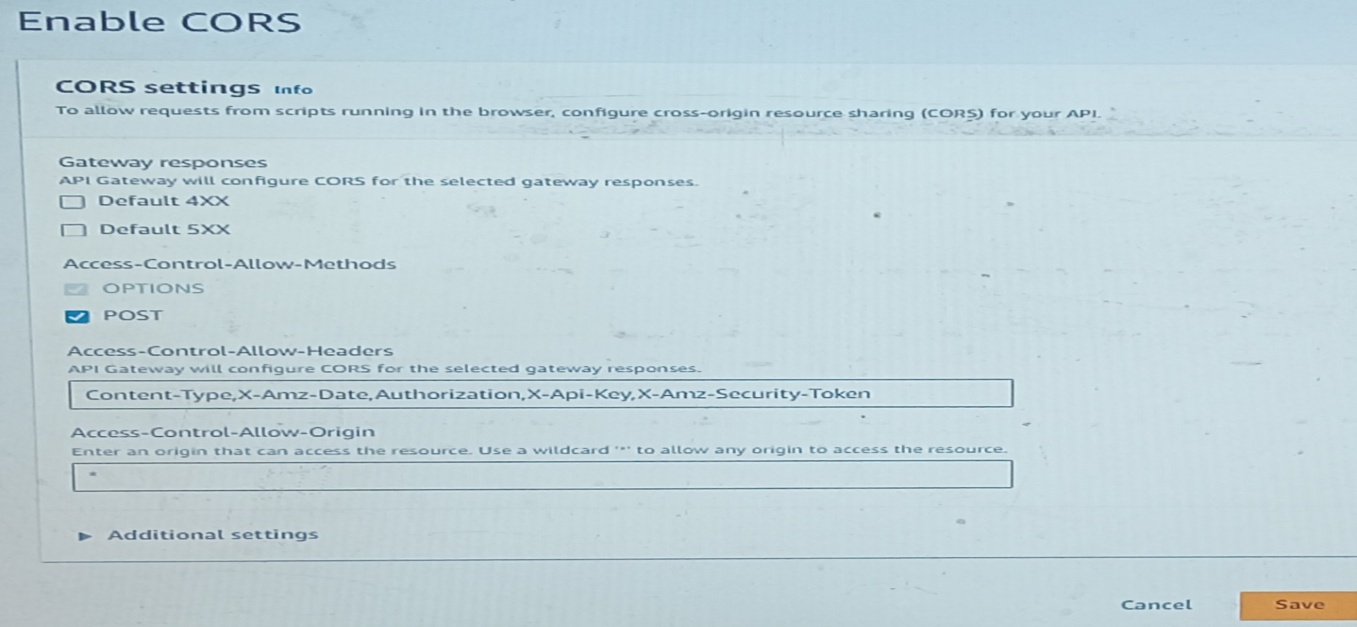


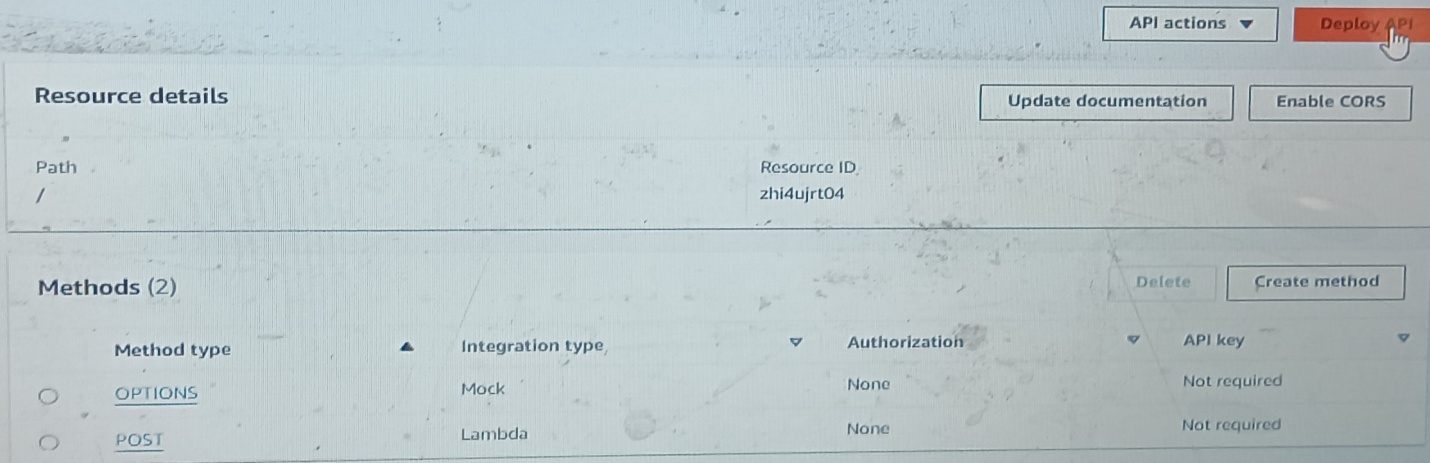
4) Next, we have to enable Cross Origin Resource Sharing (C.O.R.S). Enabling C.O.R.S allows a web application running in one origin or domain to be able to access resources on a different origin or domain. So because our web application is running in one domain on Amplify and our Lambda function will be running in another, we need to be able to work across these domains or origins that’s why we’re enabling C.O.R.S.

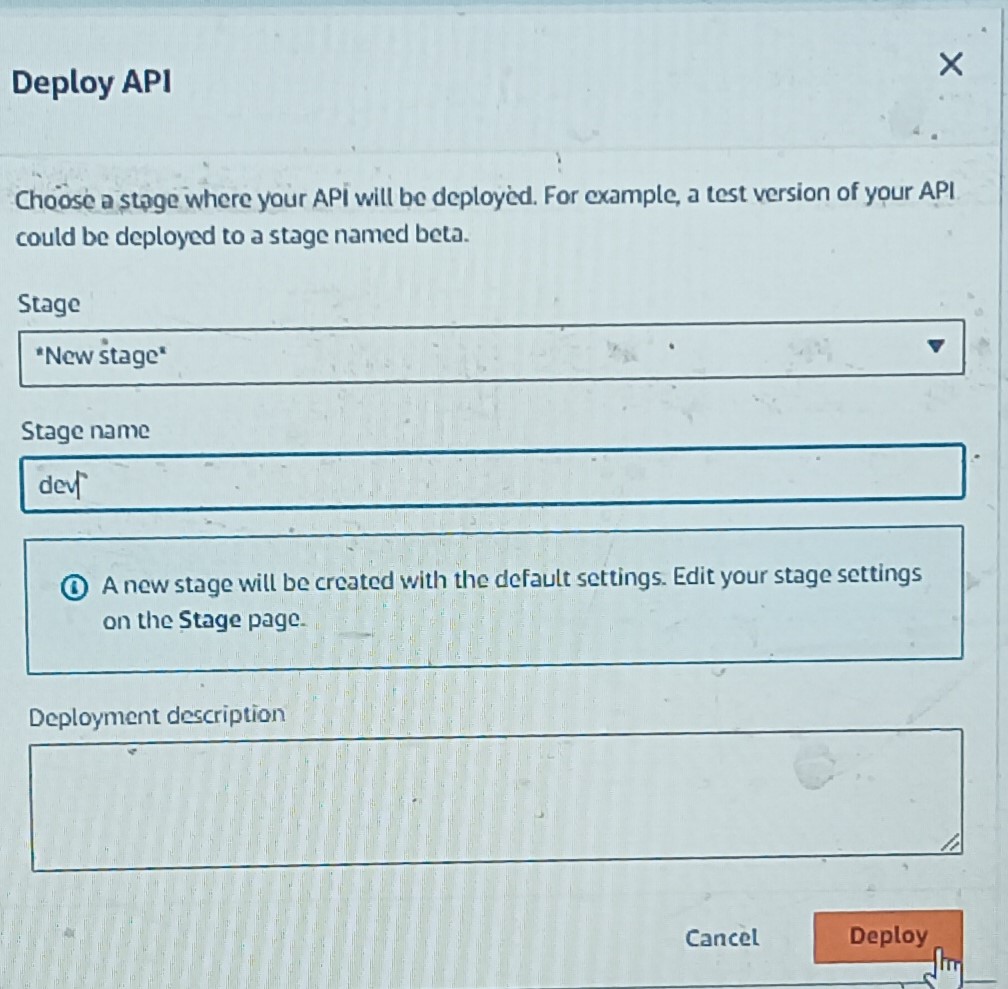
6) Select POST and leave the other settings as default and click save.

Now let’s deploy API so we can test it out.( Copy the Invoke URL and save it somewhere for later.)

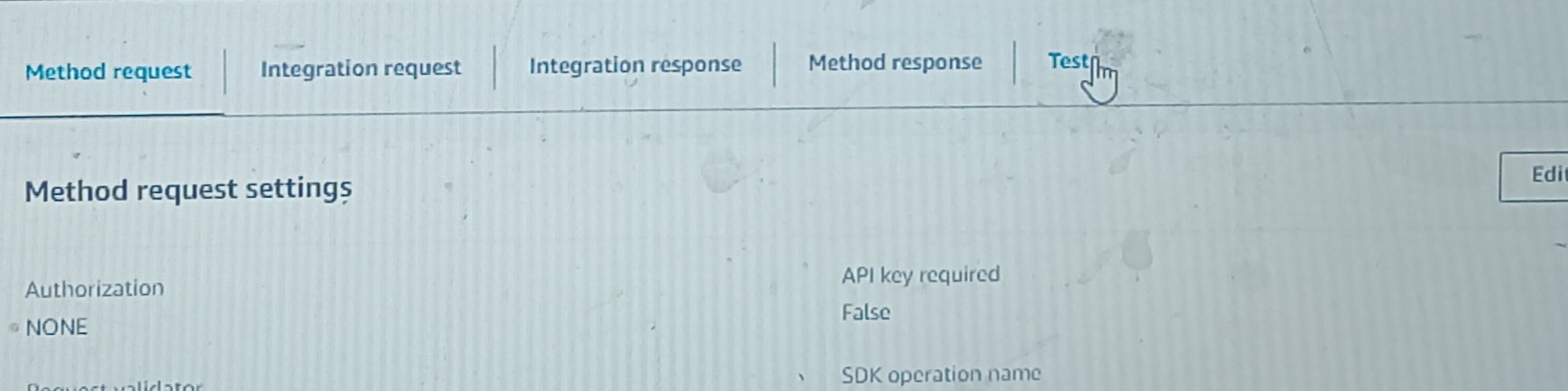


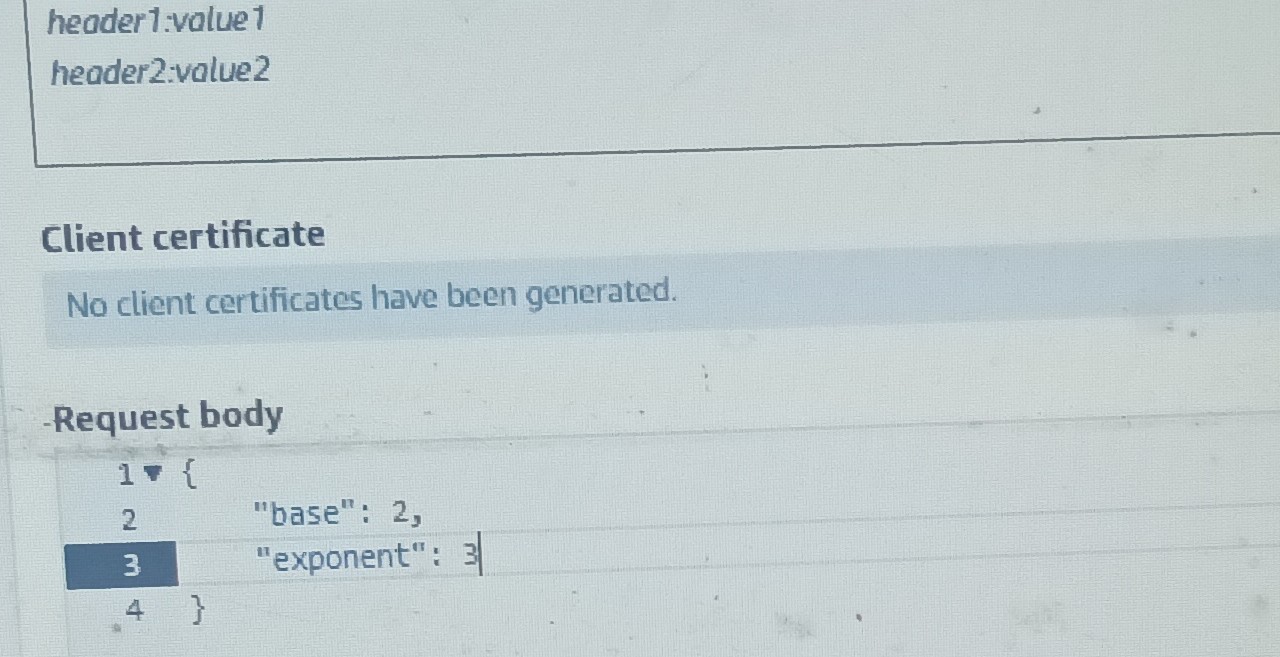


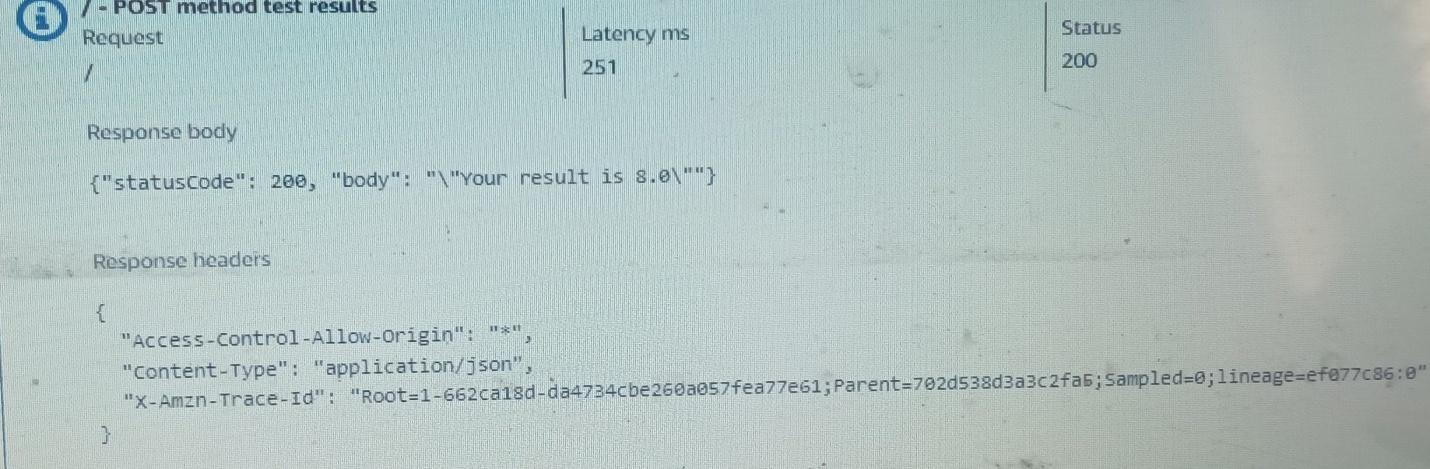




7) On the left hand side of your console click on “Resources”, then click on POST. Go to Test. “Request body” Enter values Click on “Test”.



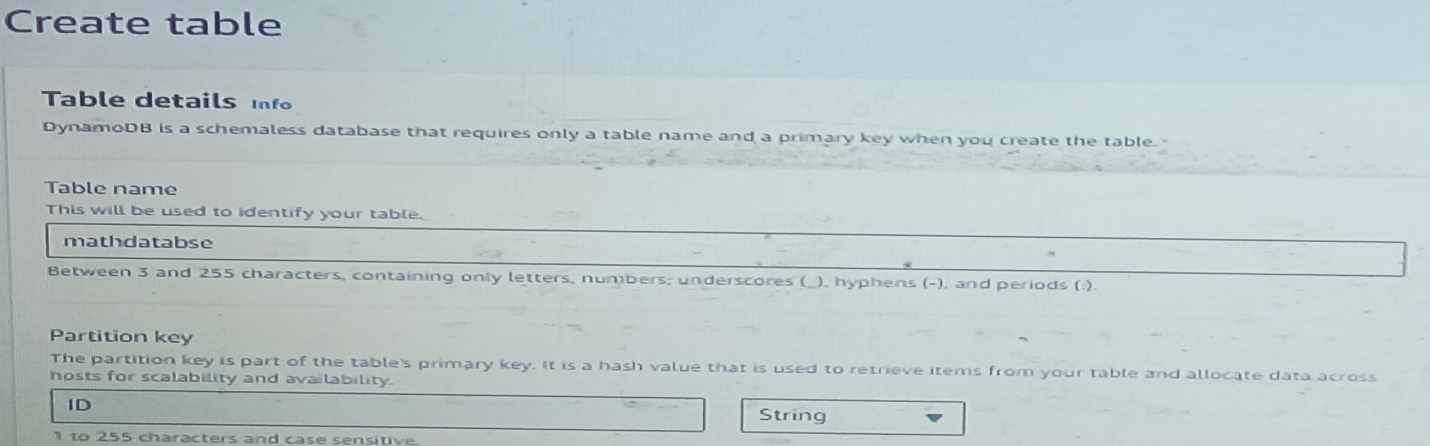




# ****Somewhere to store/return the math result****

1) On your AWS console, navigate to DynamoDB console and create a table.

2)Click on “Create Table” Name the table Partition key **/** ID**/** String Create Table.



3) Save the Amazon Resource Name (ARN) for the table. (Do so by clicking into the table, under General Information select Additional Info, copy the ARN and paste it into the same notepad or wherever you stored the other links.)

# ****A way to handle permission****

We have to give our Lambda function permission to write to our database table.

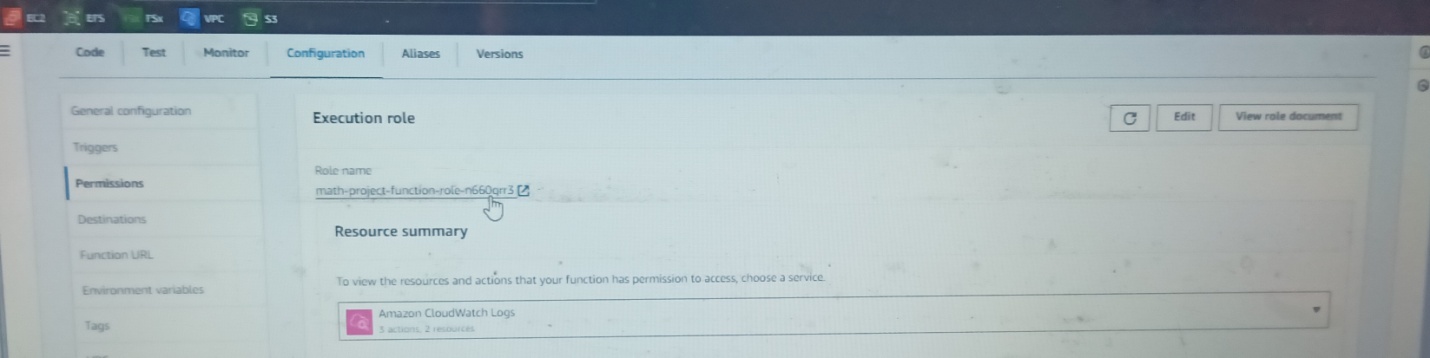
1)Go back to our Lambda function, go to the “Configuration” tab, click on “permissions”. Then click on the role name under Execution role.

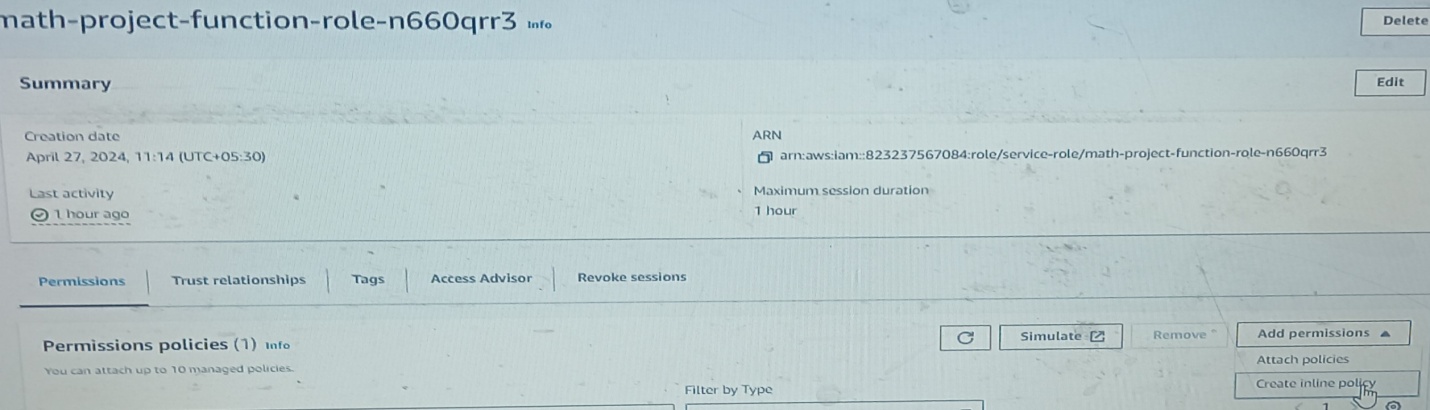
2) The execution role should open up in a new tab in your browser. We need to add some additional permissions to what the role has already, permissions related to DynamoDB. On the right click on Add permissions > Create inline policy. Policy editor should be JSON.

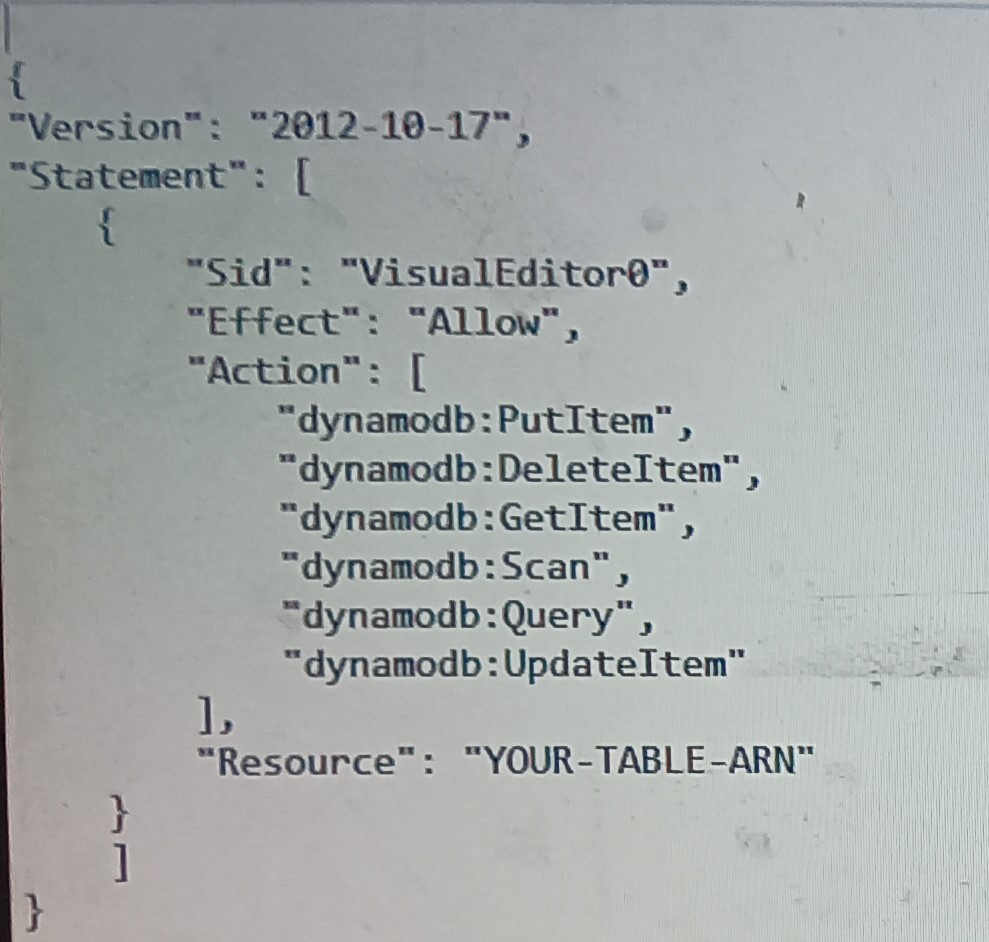
3) Paste this JSON Policy.

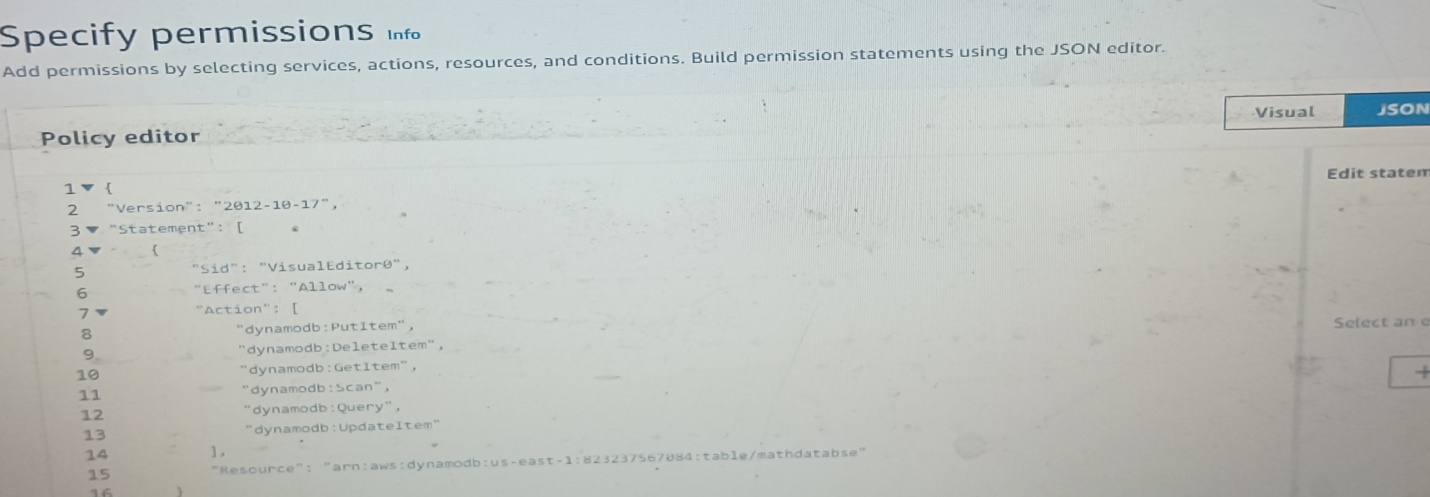
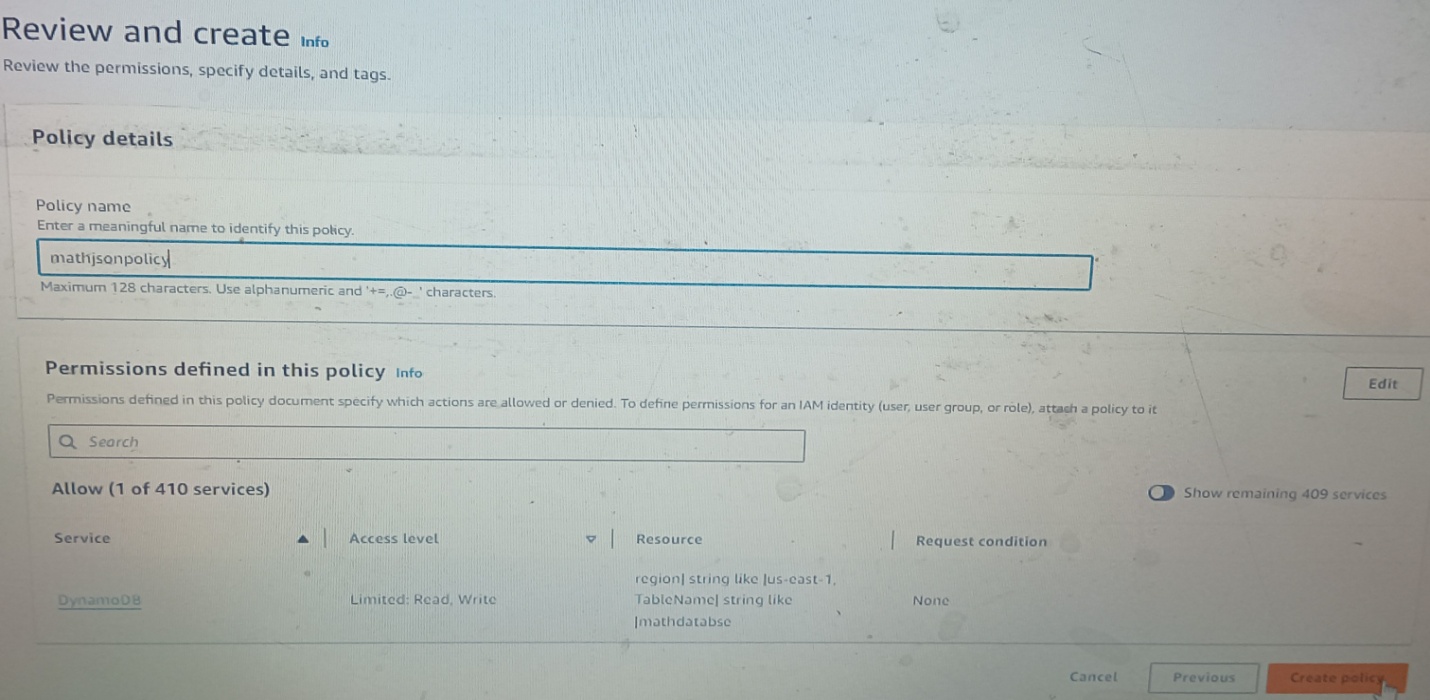
4) update the “Resource”: “YOUR-TABLE-ARN” with the table ARN you copied and saved a while ago.

5) After that scroll down and click on Next. Give your policy a name Then proceed to create the policy.





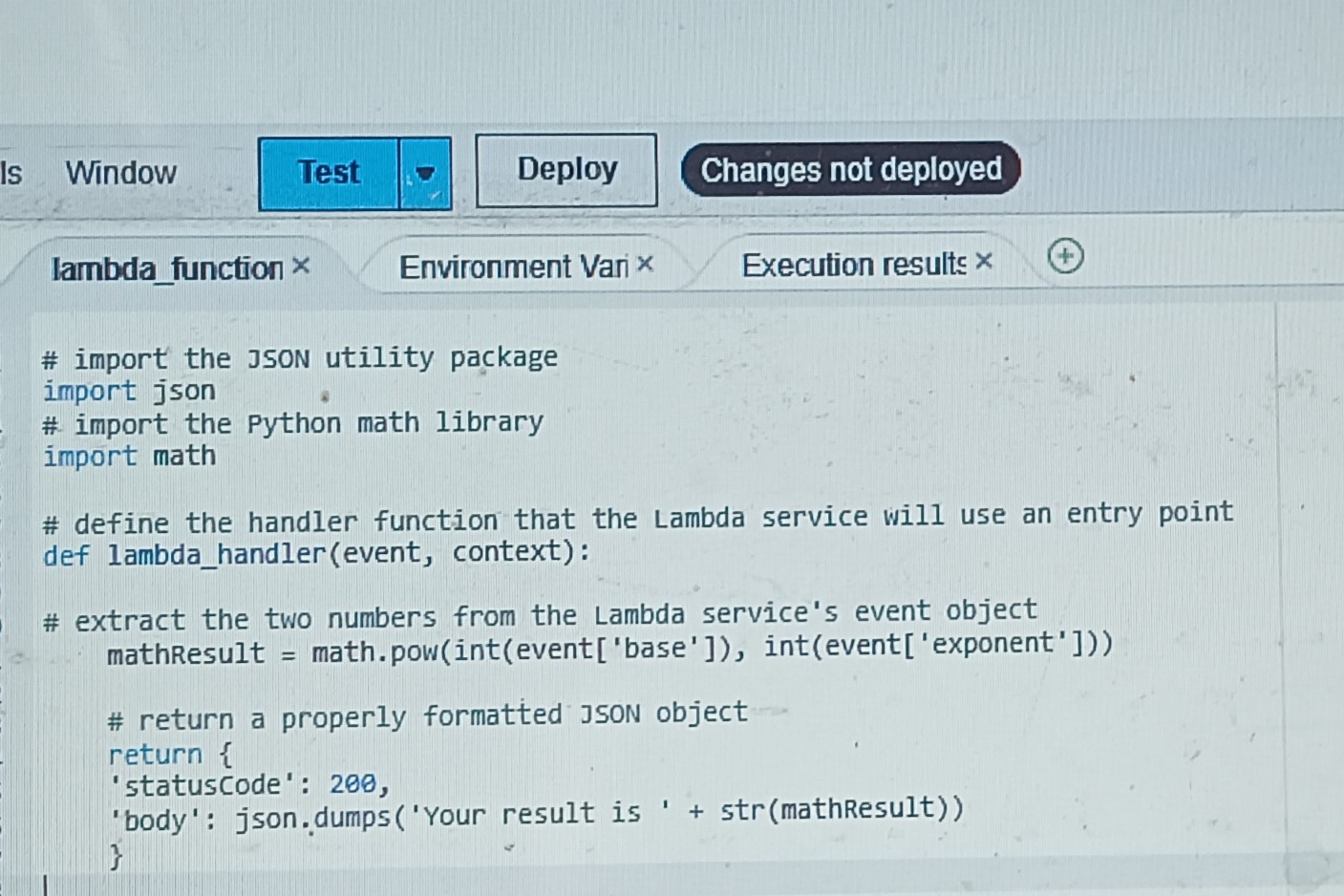


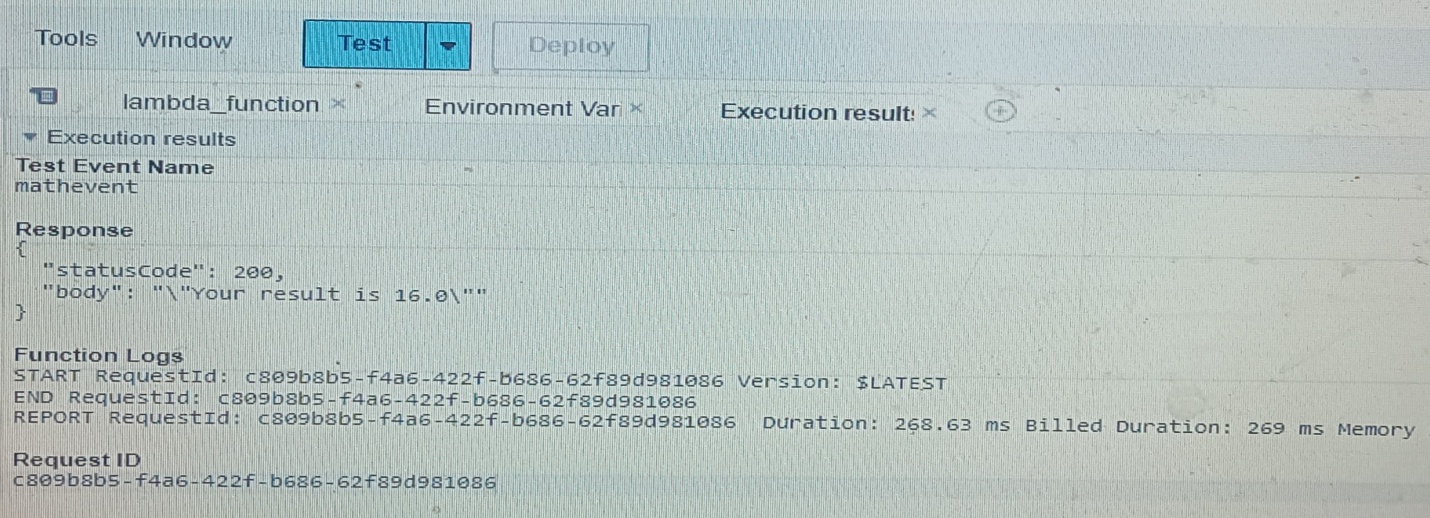
**let’s update the Lambda function to actually go write to the database.**

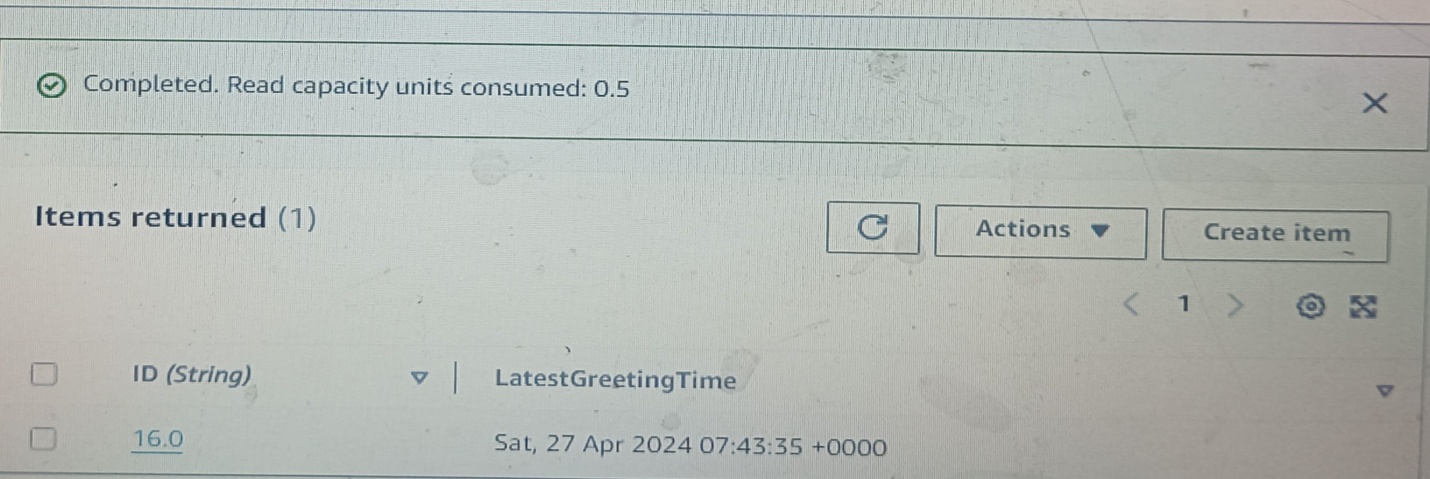
6)  Go back to your Lambda function, under the Code tab, replace the code and deploy.

7) After deploying click test. If your execution result is correct then you’re good to go.

8)Go to dynamodb table. Click into your DynamoDb table and click Explore table items (This will show you what has been stored on your database). we have our result and also the time occurrence.





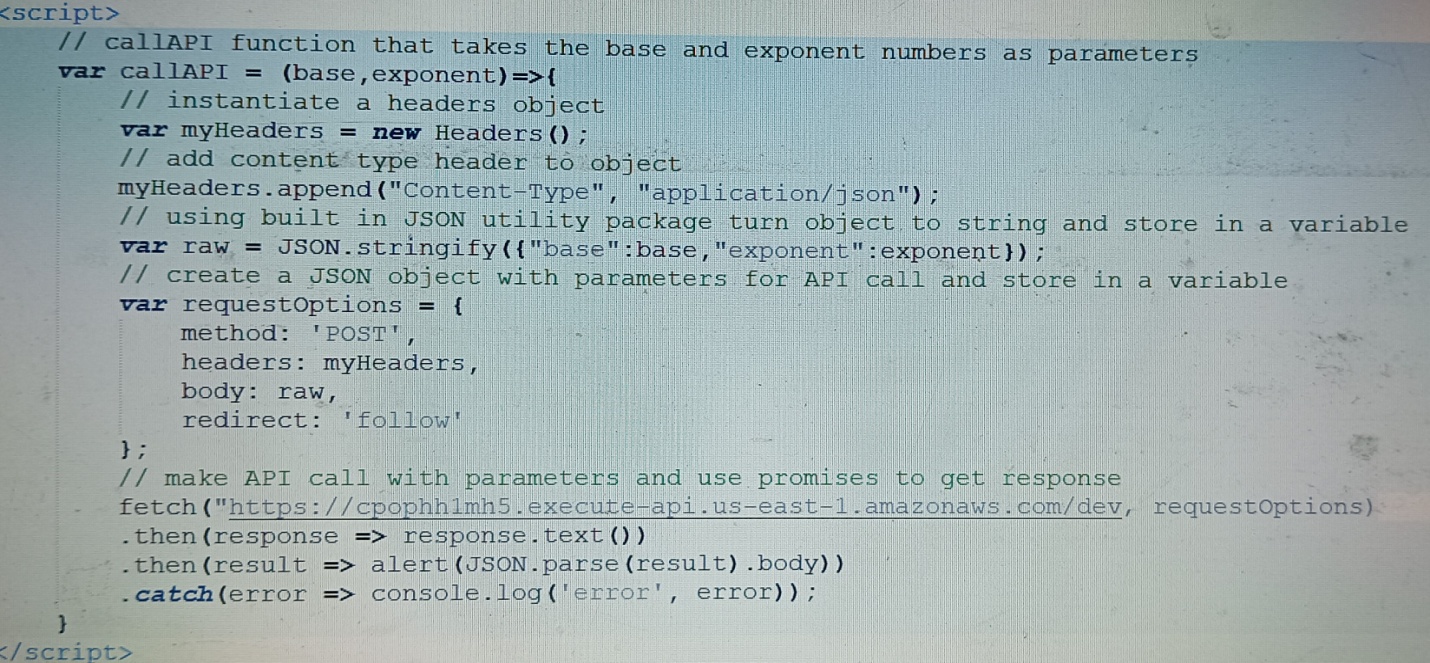


Finally we have to upload new index.html file to AWS Amplify.

In the script, you have to update your “API Gateway Endpoint”. So copy your Endpoint url from where you saved it earlier and update it.(In text editor).

create a new zip file from the updated one. Then redeploy it on Amplify.

After successfully deploying it, you can click the domain link to check out your web application and test it out.



API Gateway Endpoint

