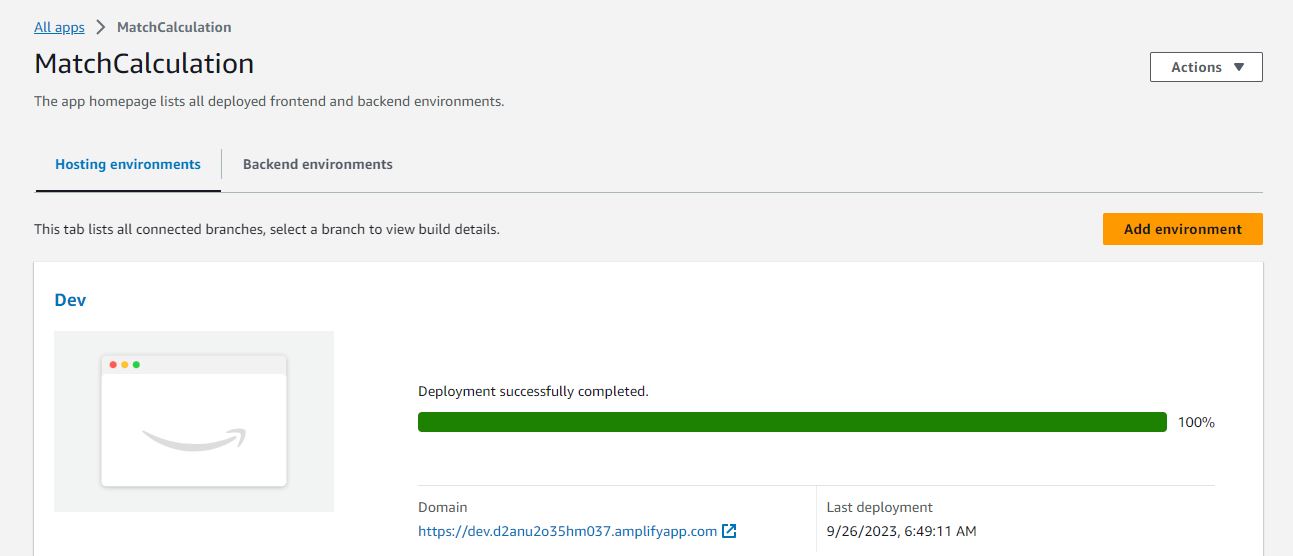
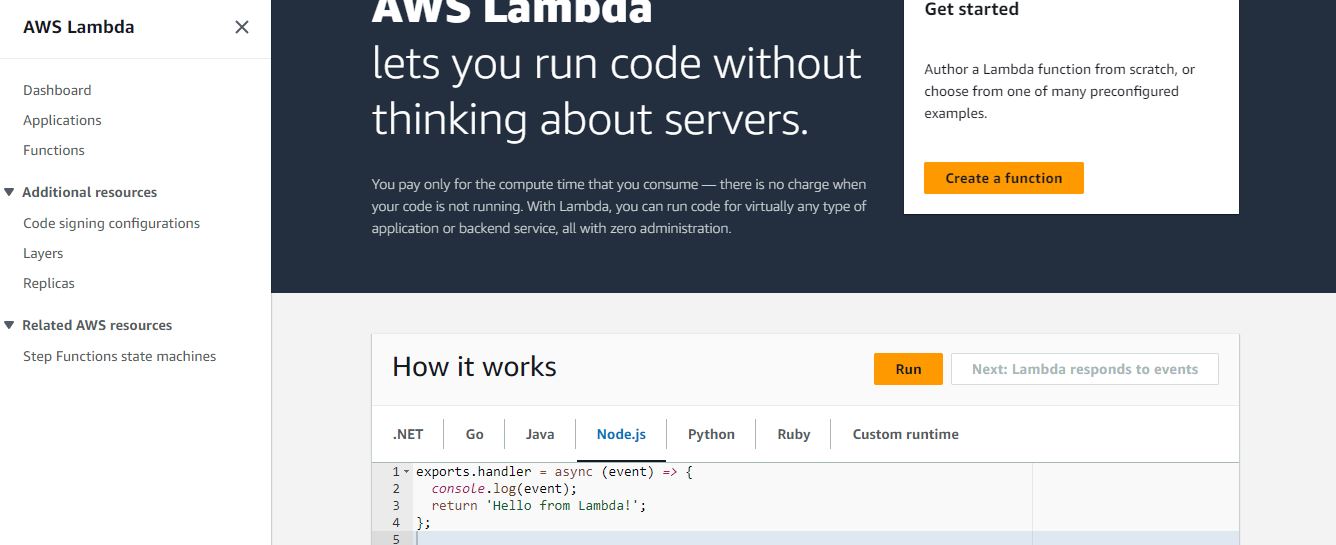
**Architect Build an End-to-End AWS Web Application**

In this project, I’m going to create and Architecture and Build an End-to-End end highly available fault tolerant AWS Web Application from Scratch. In this Project, I have used AWS services like AWS Lambda for serverless computing service, Identity and Access Management (IAM) this allows you to manage access to AWS resources securely, AWS API Gateway to create, publish, maintain, monitor, and secure APIs (Application Programming Interfaces) for this applications, AWS Amplify used for enables developers to build scalable and secure cloud-powered applications, and DynamoDB this is require a flexible and fully managed NoSQL database.

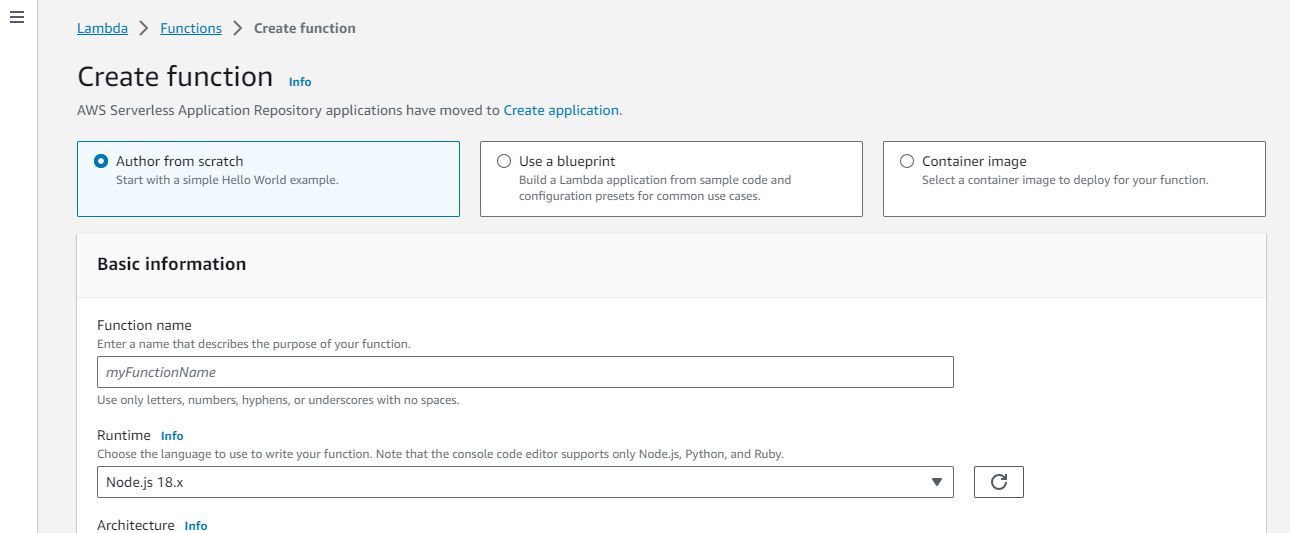
* To create a simple Html file, I’m used VScode.
* Login AWS account and search for AWS Amplify service.
* In Amplify Hosting Click on Get Started.
* Deploy without Git provider
* Uploaded simple html file
* https://github.com/Ruma-Akter-CSE/AWS\_Architect\_web\_Application



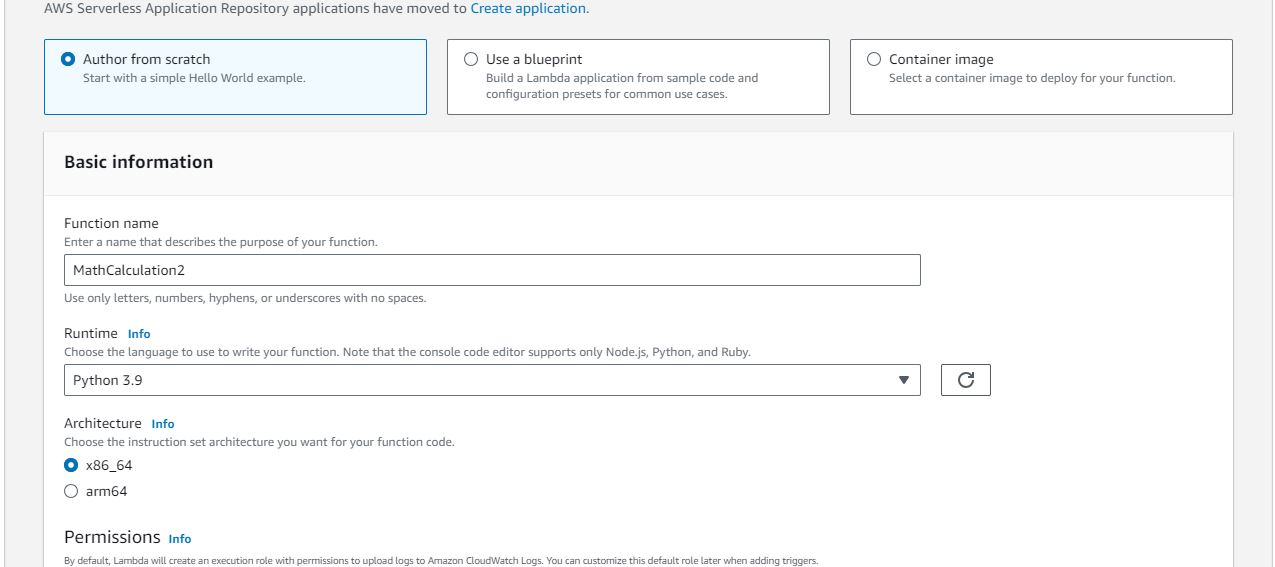
* After the deployment is successful, click on the Domain URL, it will redirect you to the new tab and you will be able to see the Webpage.
* Search AWS Lambda
* Create a function



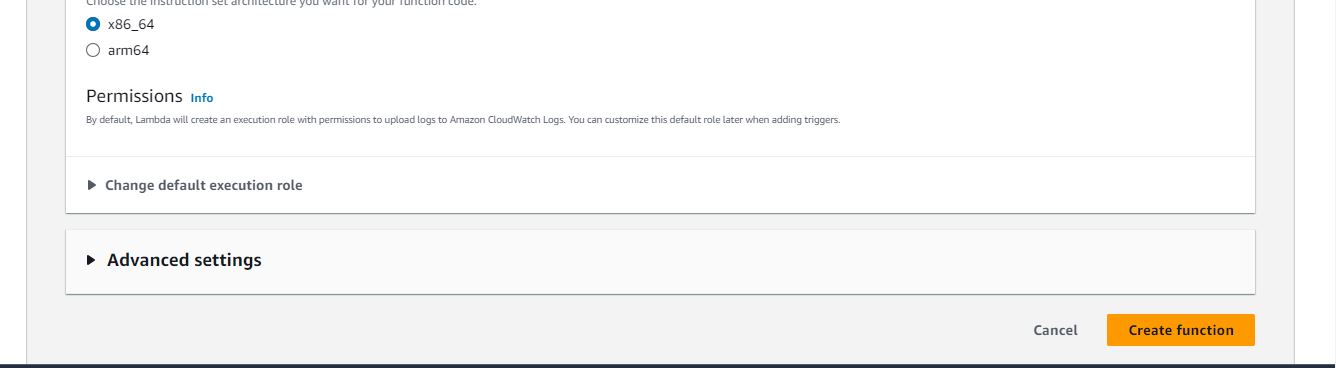
* Write a Function name



* Choose your preferable language
* Choose architecture x86\_64



* Click on Create Function



* Write a code on lambda\_function for function program check and press CTRL+s then click on deploy
* Code:

import json

import math

def lambda\_handler(event, context):

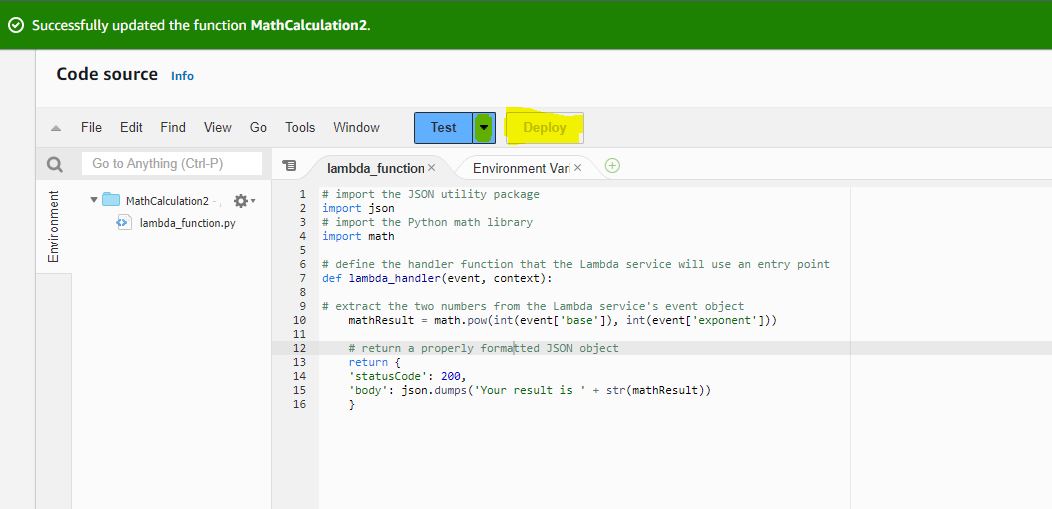
mathResult = math.pow(int(event['base']), int(event['exponent']))

return {

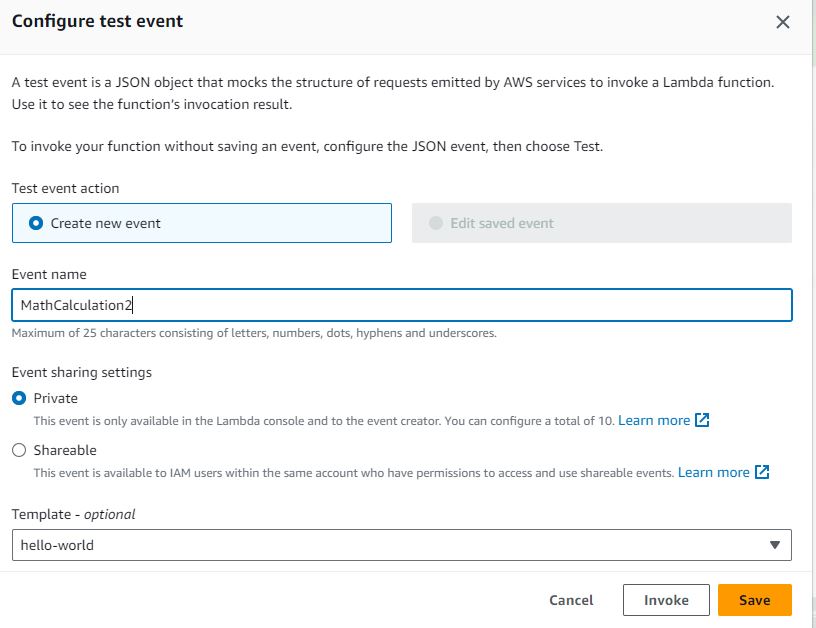
'statusCode': 200,

'body': json.dumps('Your result is ' + str(mathResult))

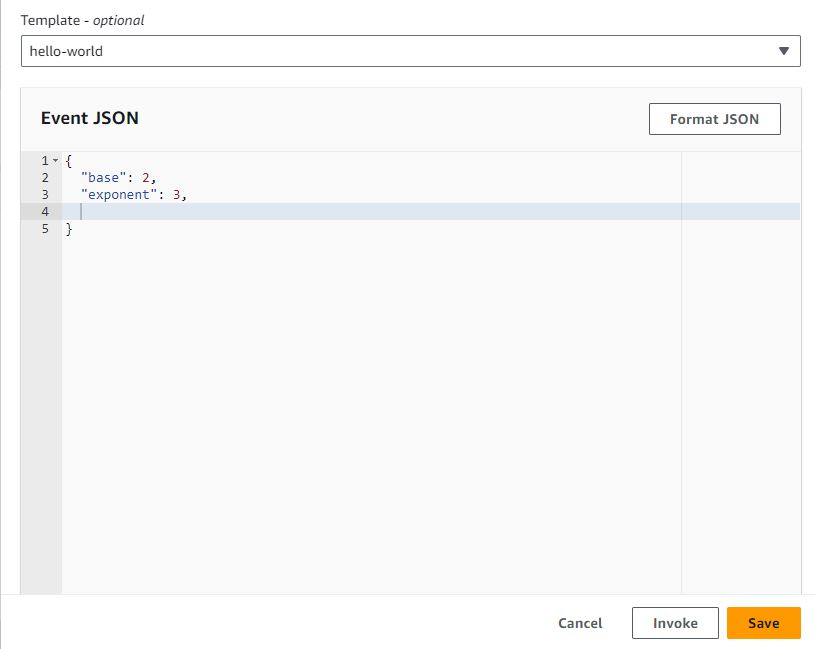
}



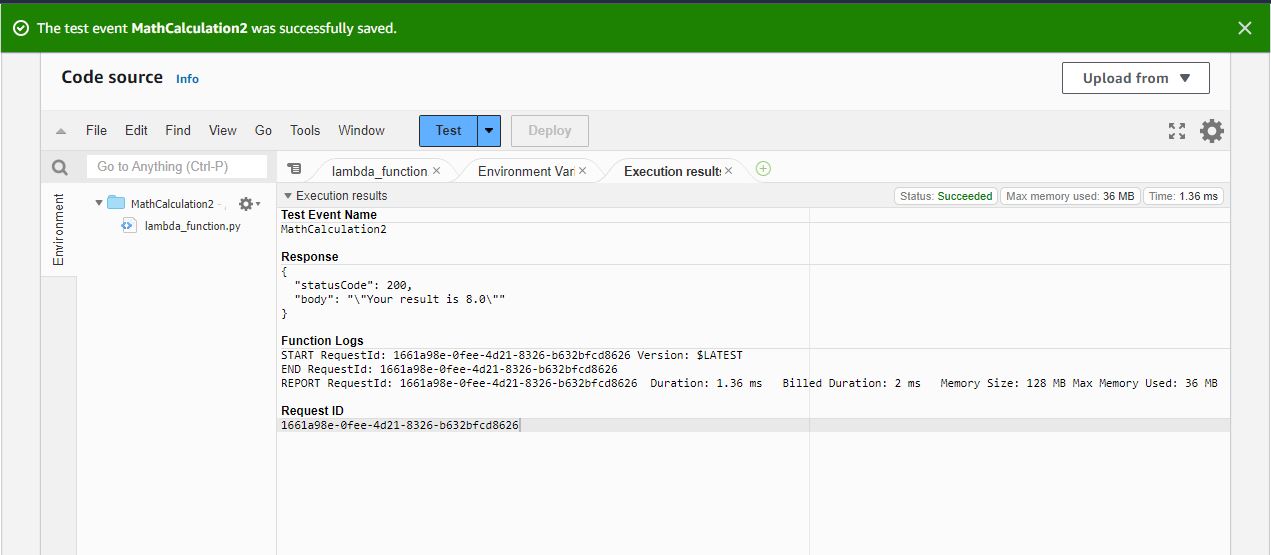
* After deploy click right on ‘Test’ and Configure test event
* Put a name of this even and leave all others as a default and click on Save



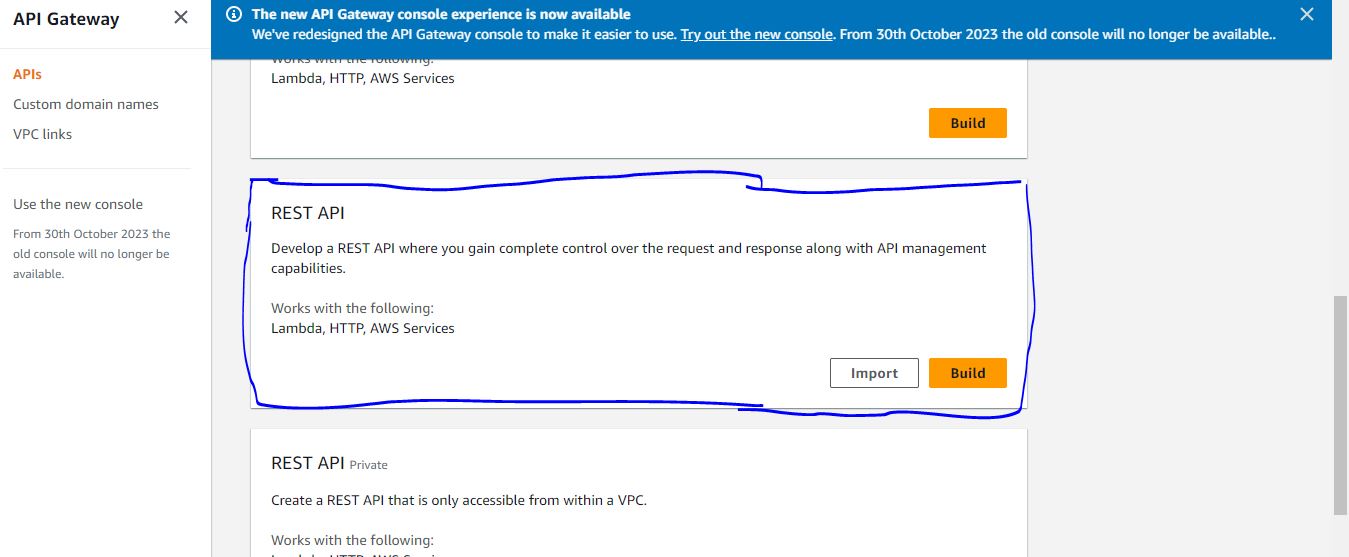
* Check your lambda function code



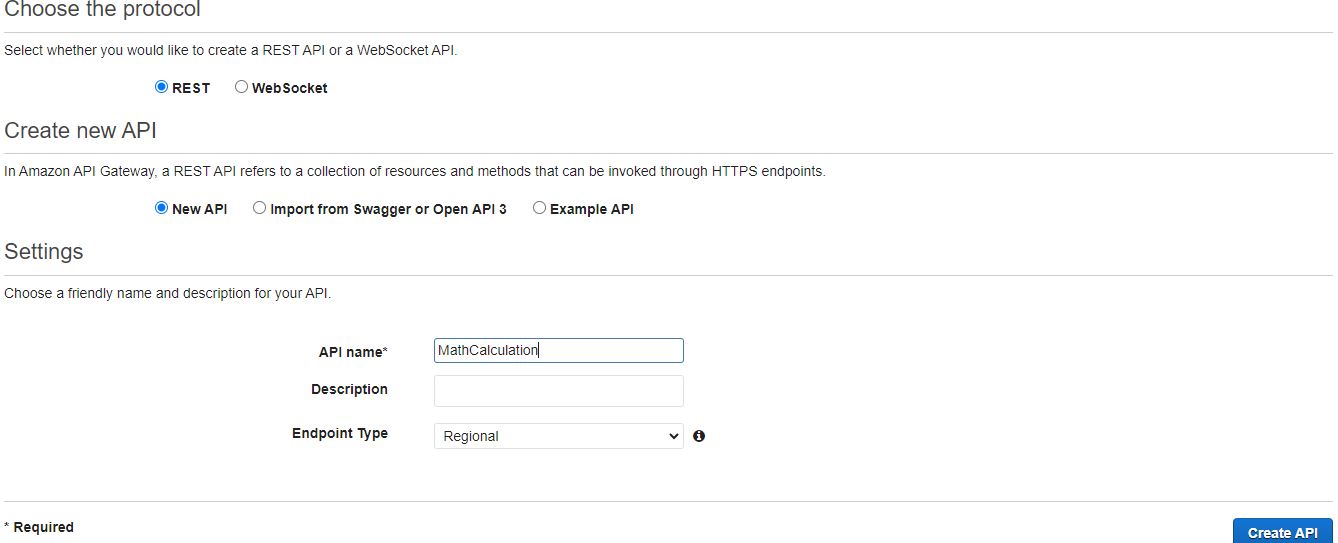
* Test event is successful



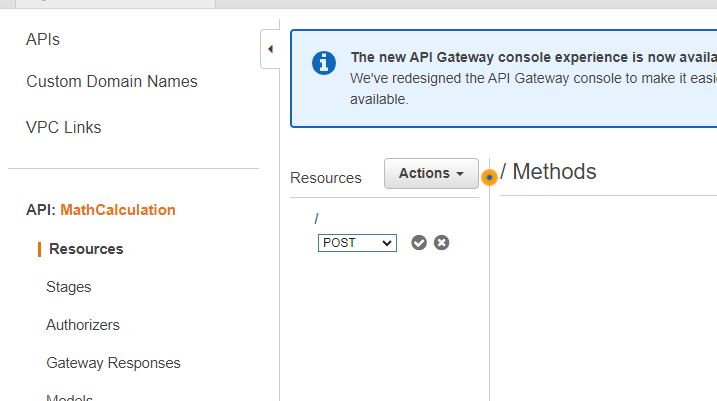
* Search AWS API
* Build a REST API



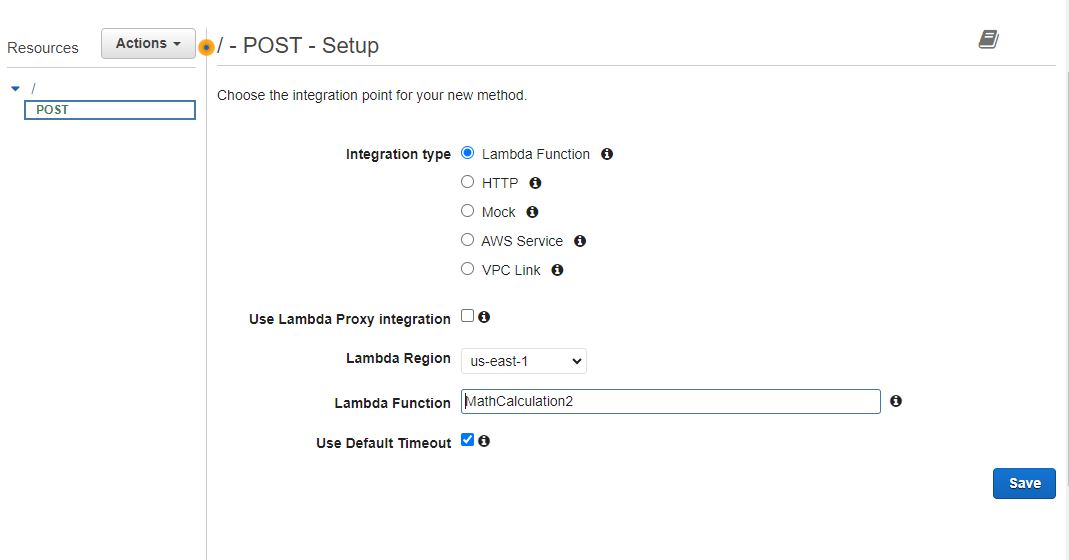
* Choose the protocol of REST and NEW API then click on Create API



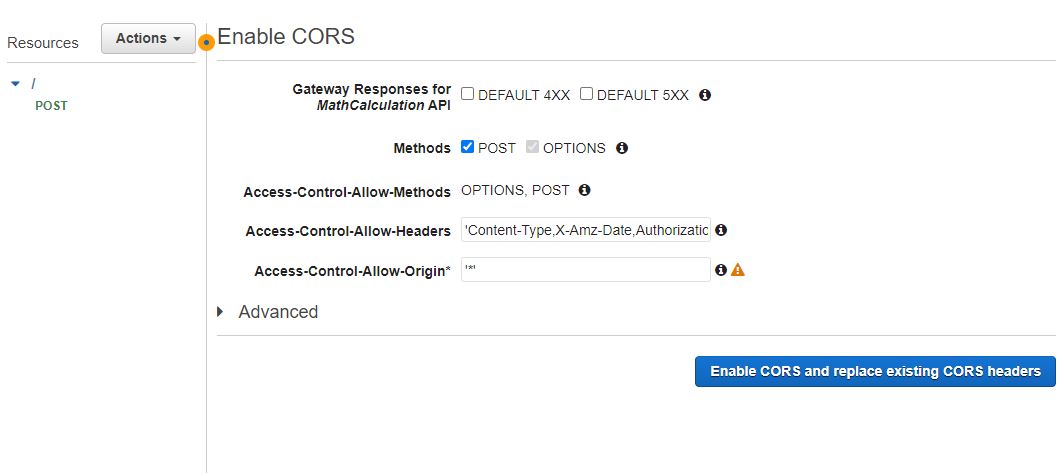
* Click on Resources and POST and choose the Action



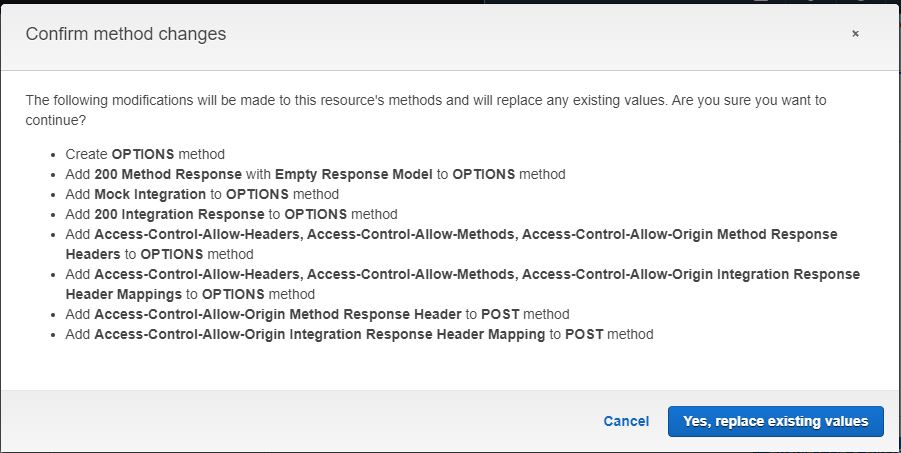
* Should be Integration type Lambda function
* Give a name of Lambda function then click on SAVE



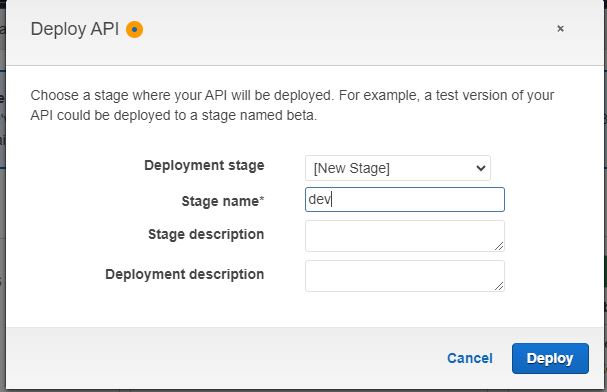
* Click on Enable CORS replace existing CORS headers



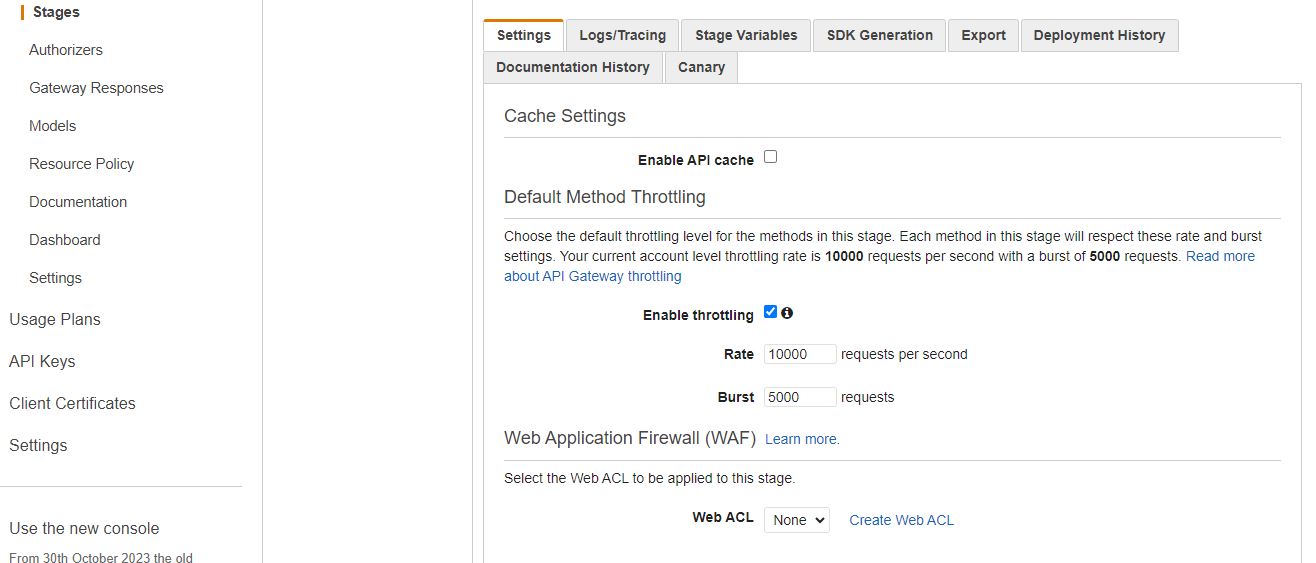
* Click on Yes, replace existing values



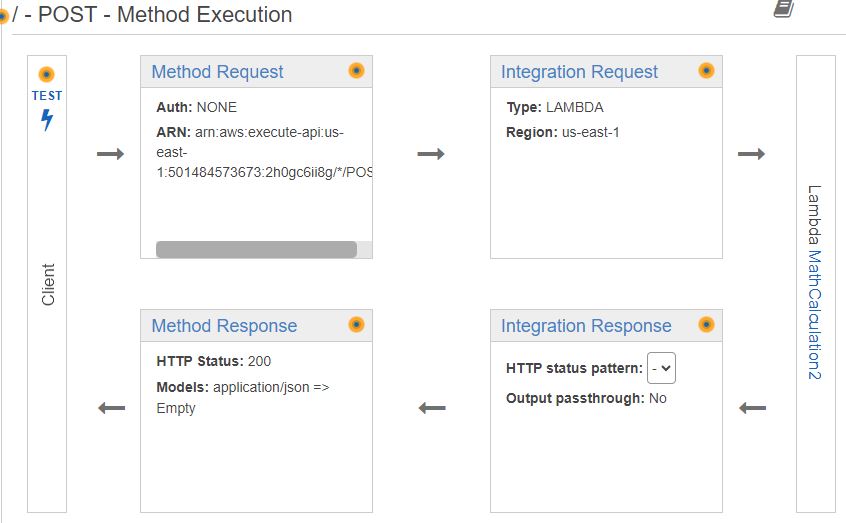
* Choose Deployment stage [New Stage]
* Give a Stage name and Deploy



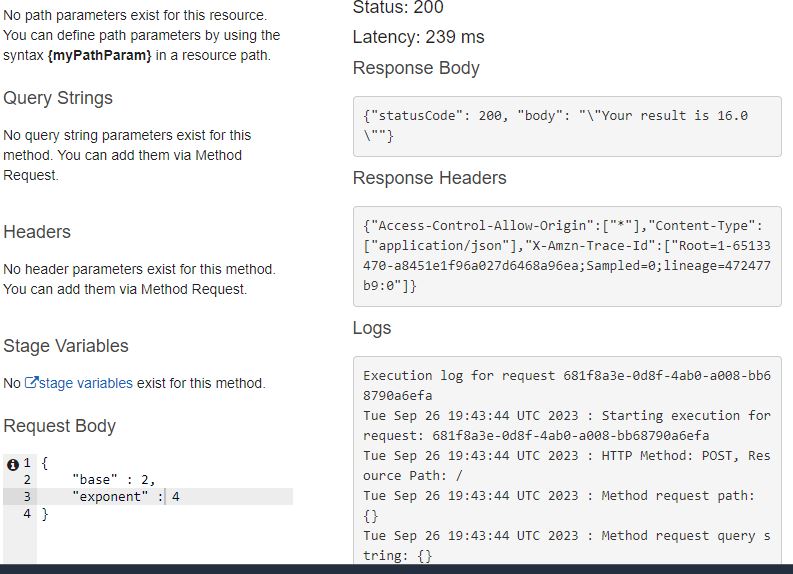
* Enable throtting



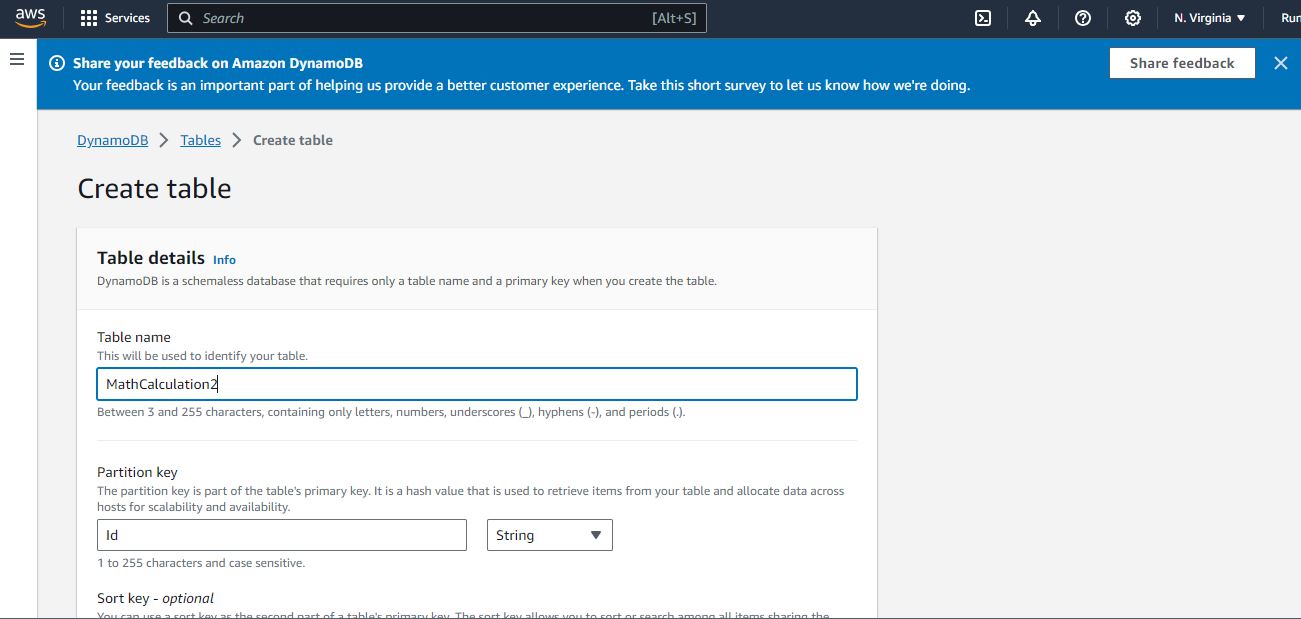
* In Method Execution click on the TEST



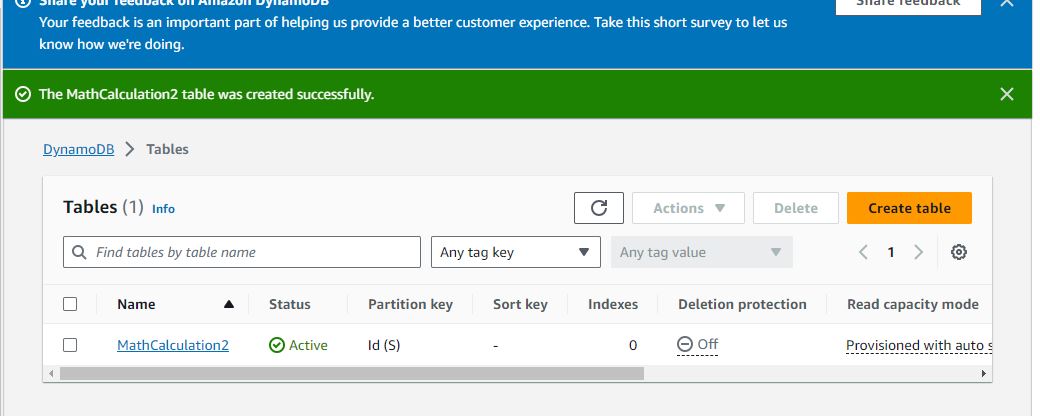
* And successfully Response body



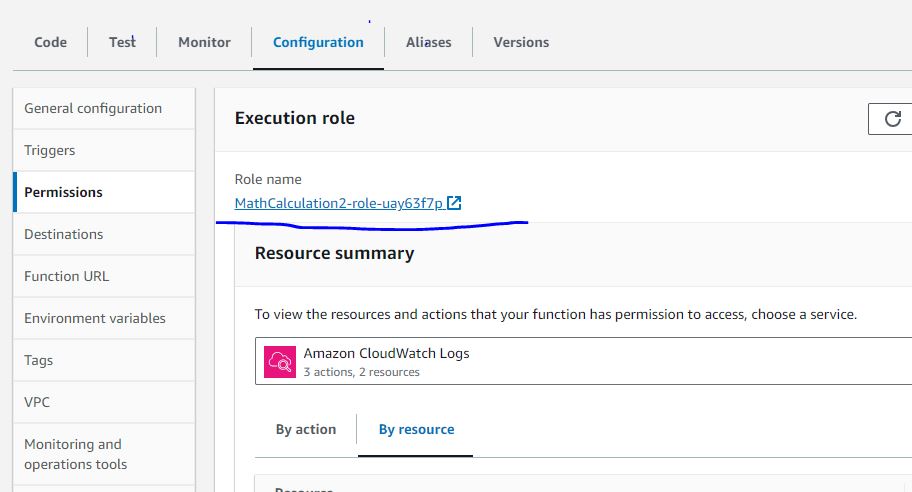
* Now search DaynamoDB and Create a new table



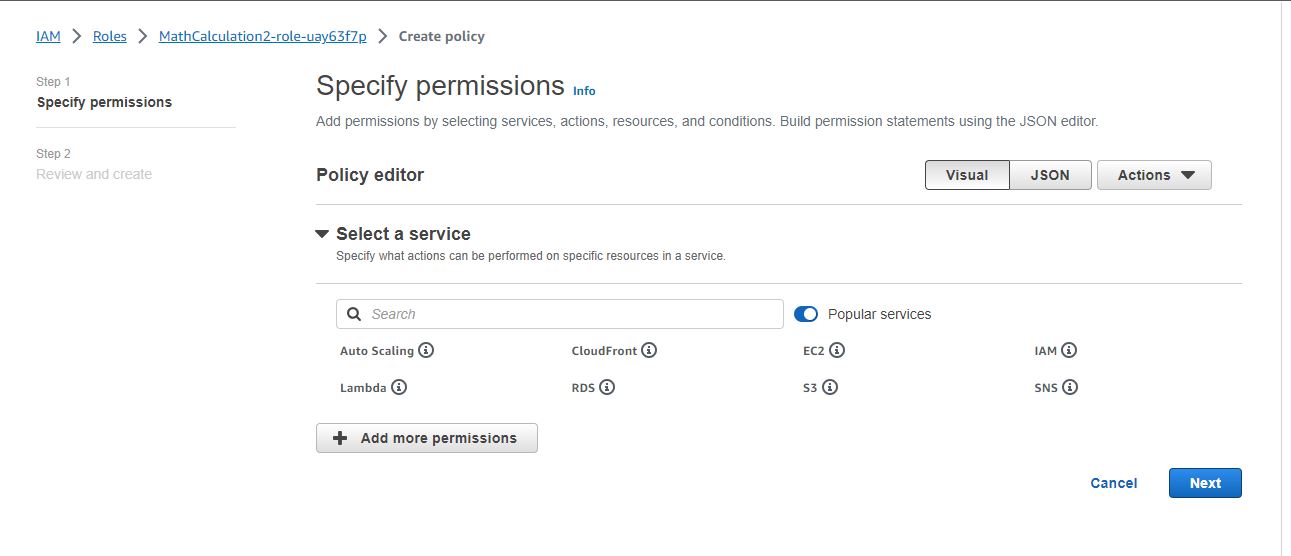
* Successfully crated table
* Collect your arn



* Go to the permission and click on the Role name



* Create a new policy



* Edit your policy
* Code:

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": [

"dynamodb:PutItem",

"dynamodb:DeleteItem",

"dynamodb:GetItem",

"dynamodb:Scan",

"dynamodb:Query",

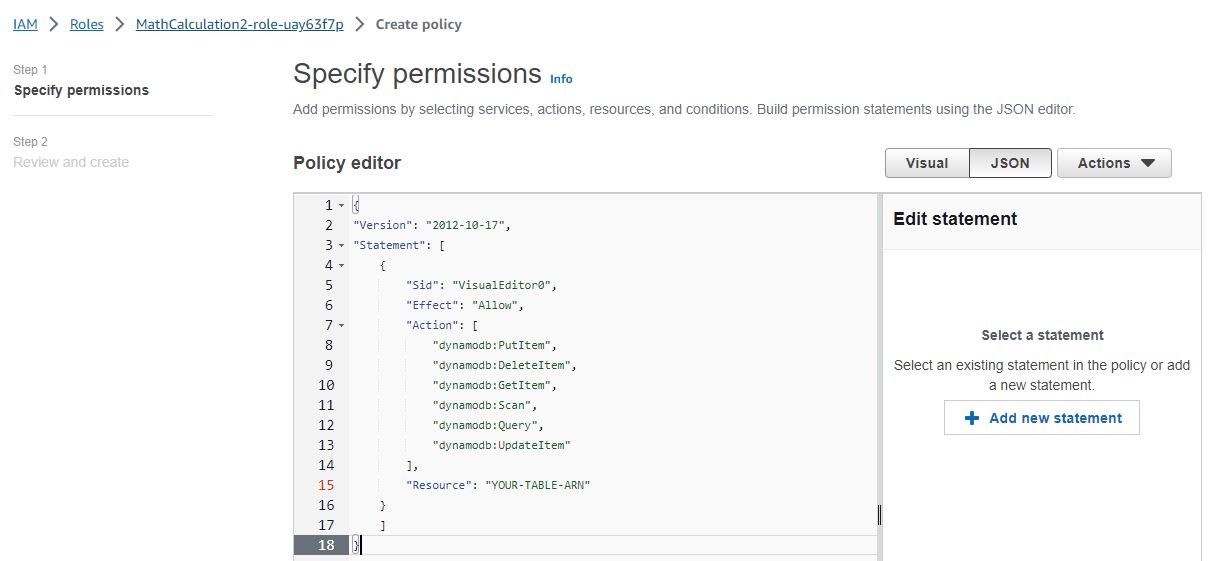
"dynamodb:UpdateItem"

],

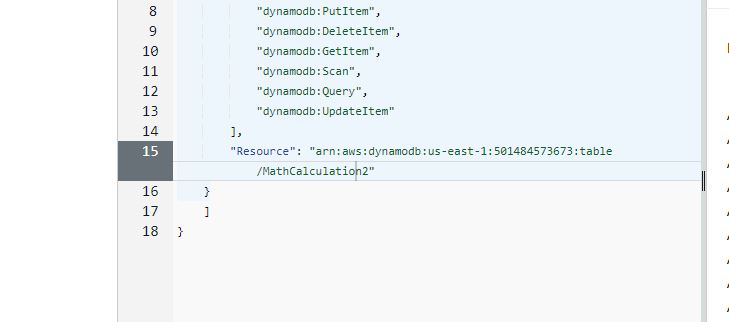
"Resource": "YOUR-TABLE-ARN"

}

]



* Given here your “arn from dynamoDB”



* Go back lambda function and write final code import needed library and dynamoDB table.

Code:

import json

import math

import boto3

from time import gmtime, strftime

dynamodb = boto3.resource('dynamodb')

table = dynamodb.Table('MathCalculation2')

now = strftime("%a, %d %b %Y %H:%M:%S +0000", gmtime())

def lambda\_handler(event, context):

mathResult = math.pow(int(event['base']), int(event['exponent']))

response = table.put\_item(

Item={

'id': str(mathResult),

'LatestGreetingTime':now

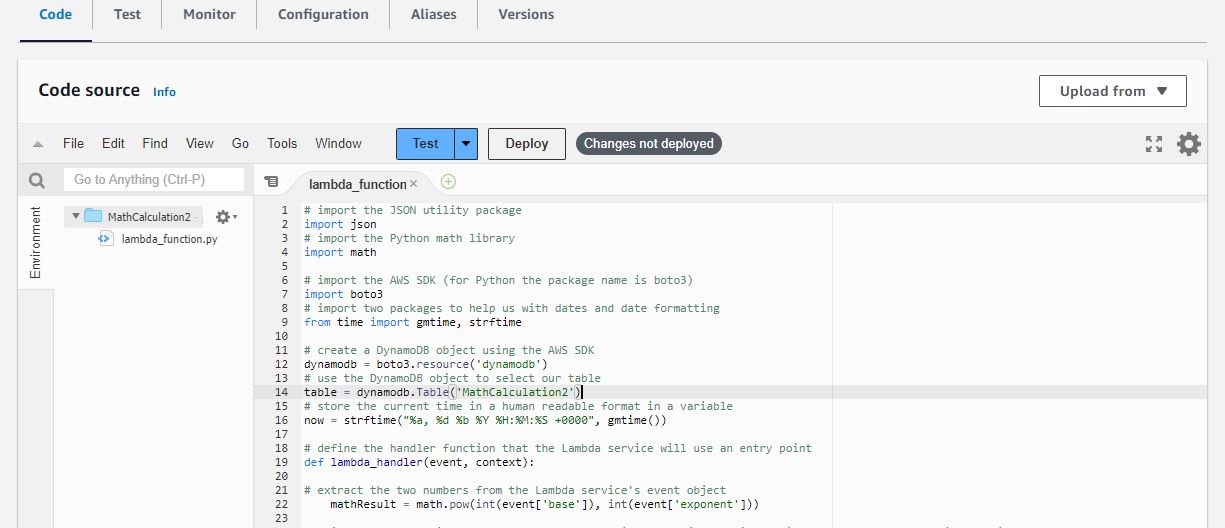
})

return {

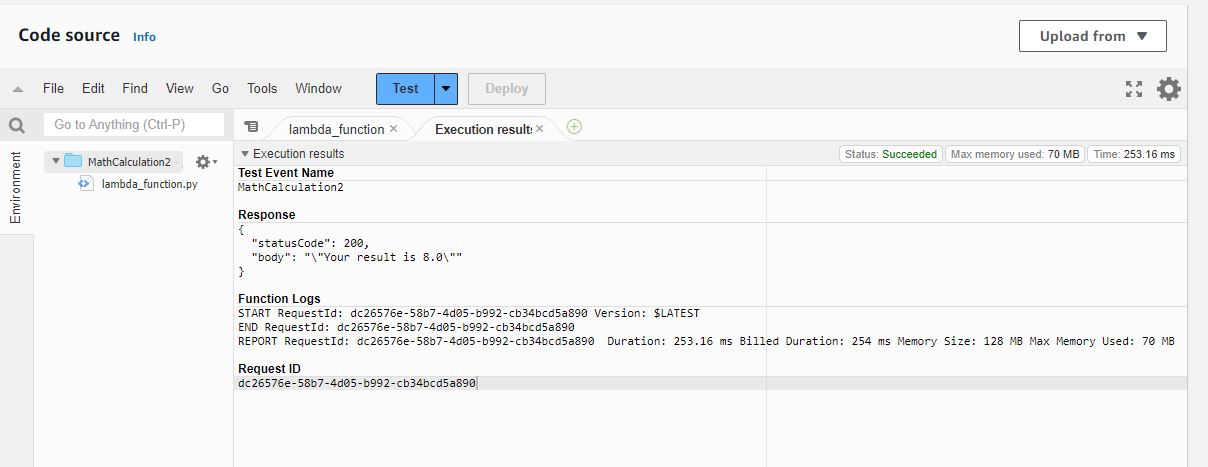
'statusCode': 200,

'body': json.dumps('Your result is ' + str(mathResult))

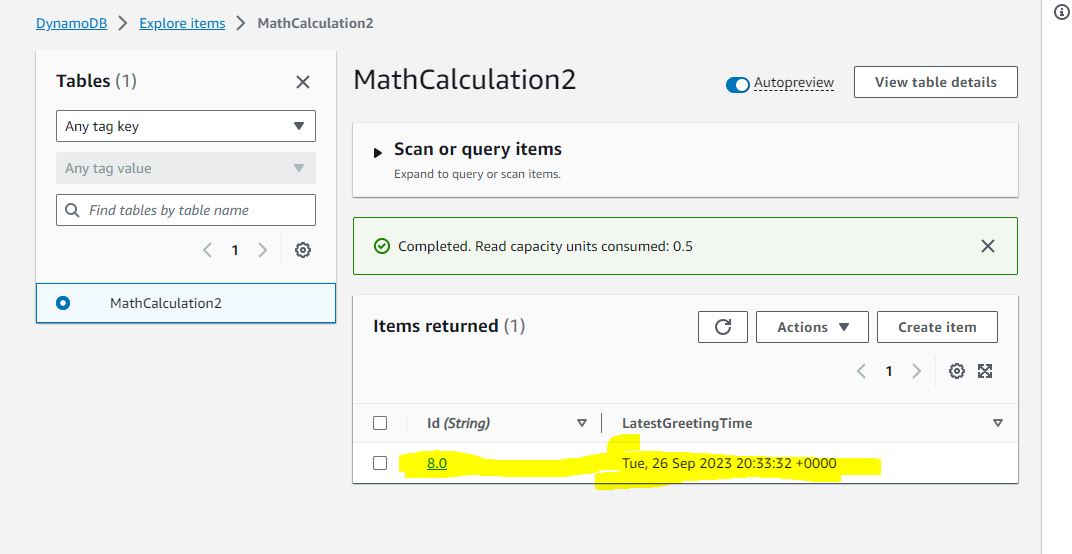
}



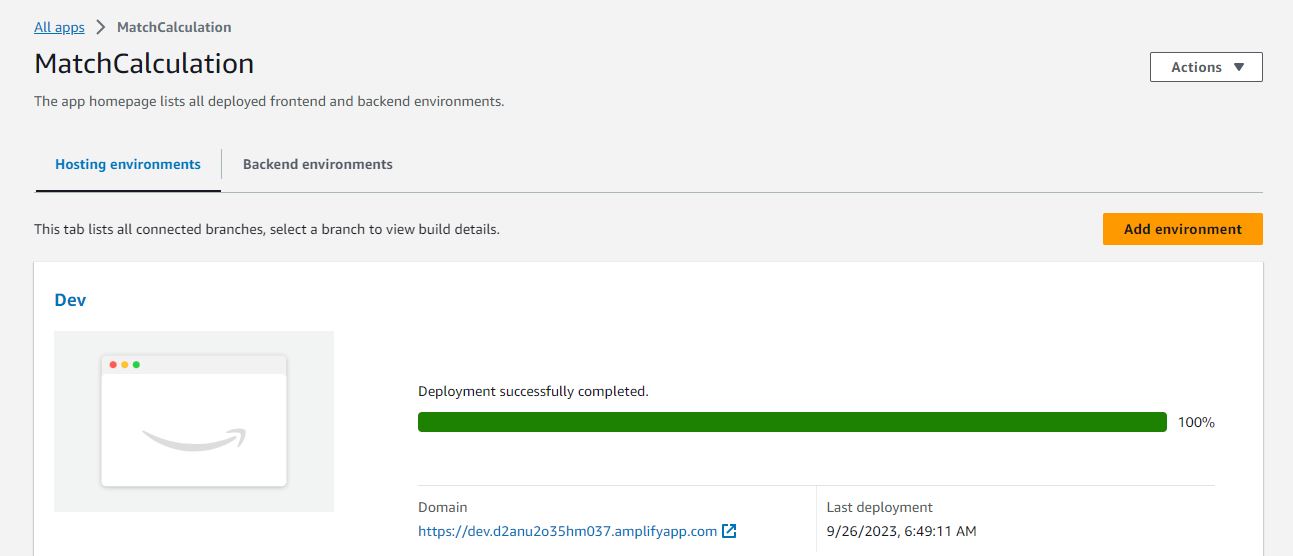
* Checked the code write or wrong



* Finally build architecture with multi-tier



* And after all of that drag the final code on AWS amplify
* Click on domain



* The final result of simple base calculator
* https://github.com/Ruma-Akter-CSE/AWS\_Architect\_web\_Application

