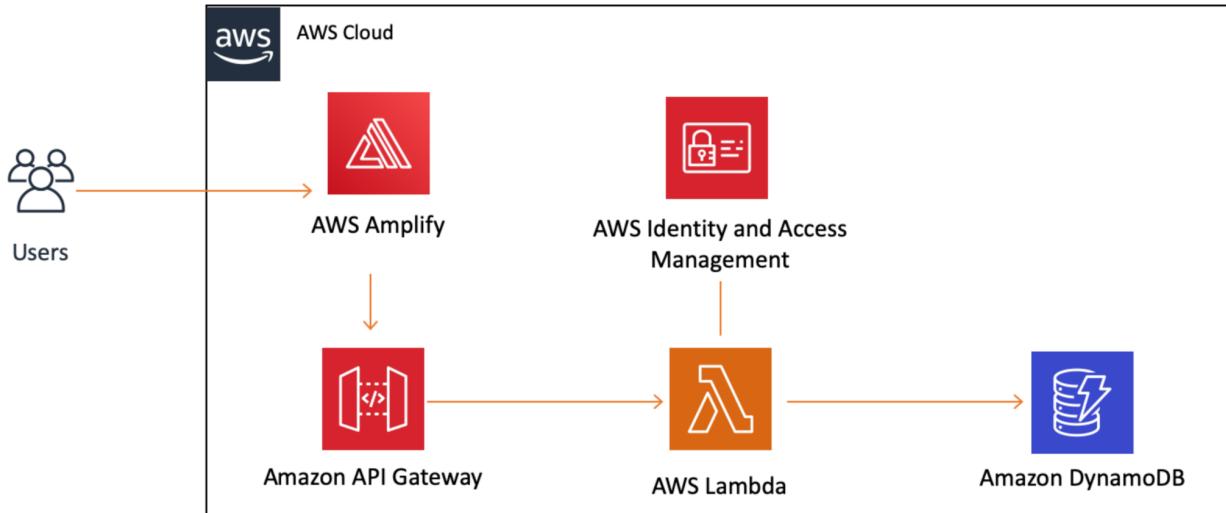


Project



Build a Basic Web Application

20.10.2023

—

Himanshu Nimje
Nagpur-440017

Overview

In this Lab, creating a simple web application, first build a web app, Then add how to add Functionality to the web app so the text that displays is based on a custom input I provide.

Goals

1. Enhance User Experience
2. Learn and Apply Technologies
3. Gain Development Experience
4. Explore Technologies and Tool

Benefits

1. Accessibility:- Global Reach:- A web application allows users to access service from anywhere with an internet connection, providing a global reach for content or services.
2. Community and Support:- Community Resources:- Leveraging cloud platforms connects you to a vast community of developers and resources, including forums, documentation, and tutorials.
3. Adaptability:- Adapt to Changes:- Web applications are flexible and can be easily adapted to changes in user requirements or technology trends.

Service Used

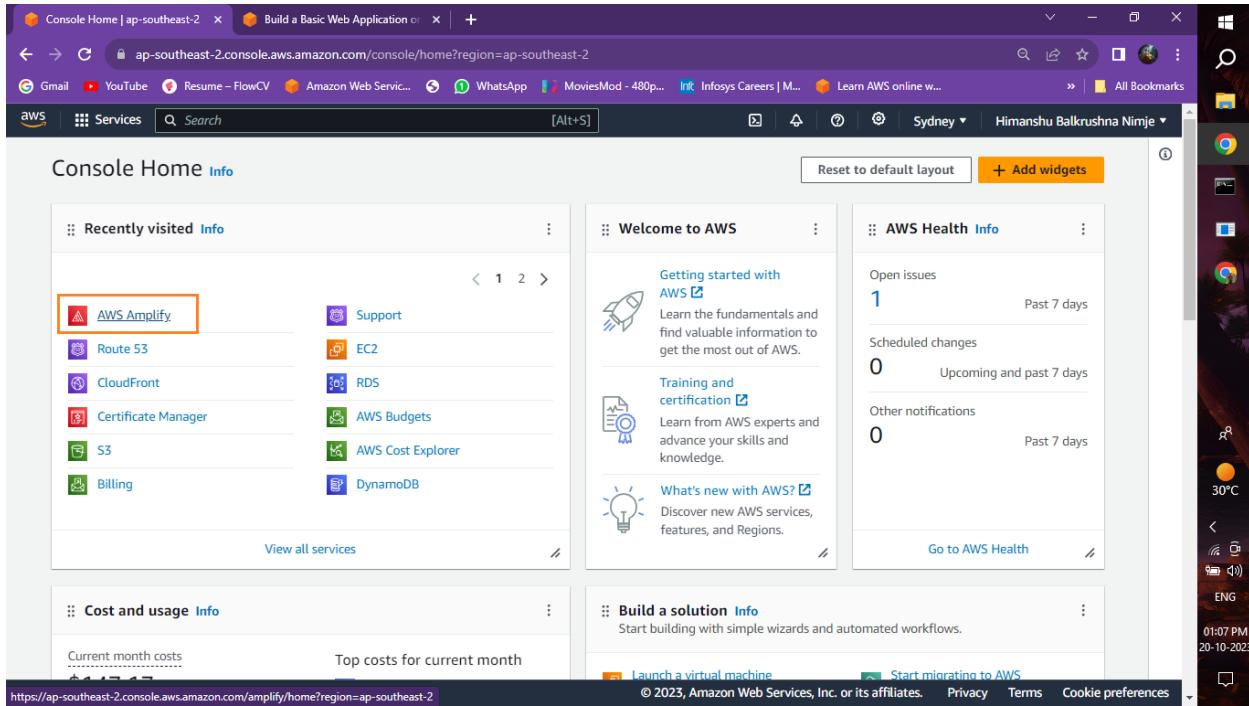
AWS Amplify, Lambda function, IAM Roles, Amazon DynamoDB,
Amazon API Gateway

Steps

1. Create a web app
2. Build a serverless function
3. Create a Lambda function using the AWS Lambda console
4. Link the serverless function to the web app
5. Create a Data table
6. Create a DynamoDB table using the DynamoDB console
7. Create an HTTP API using the API Gateway console
8. Test your API
9. Create IAM Roles (Select the type of trusted entity)

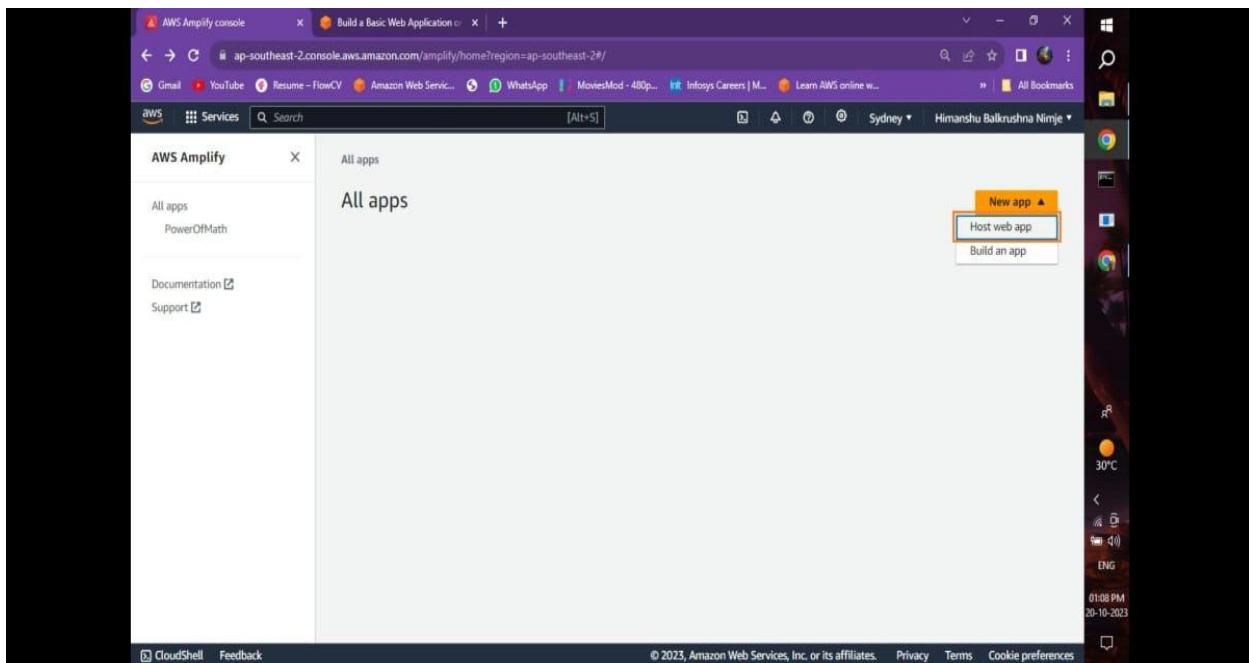
Implementation↓

Go to AWS Console Search AWS Amplify and click on



The screenshot shows the AWS Console Home page for the ap-southeast-2 region. The AWS Amplify service is highlighted with a red box. The page includes sections for Recently visited services (AWS Amplify, Route 53, CloudFront, Certificate Manager, S3, Billing), Welcome to AWS (Getting started with AWS, Training and certification, What's new with AWS?), AWS Health (Open issues, Scheduled changes, Other notifications), Cost and usage (Current month costs, Top costs for current month), and Build a solution (Launch a virtual machine, Start migrating to AWS). The URL in the address bar is <https://ap-southeast-2.console.aws.amazon.com/amplify/home?region=ap-southeast-2>.

Aws Amplify Dashboard, Go to New App select (Host web app)



The screenshot shows the AWS Amplify console. The 'New app' button is highlighted with a red box. The interface includes a sidebar with options for All apps (PowerOfMath), Documentation, and Support, and a main area titled 'All apps'.

Select → Deploy without Git Provider and click on Continue

The screenshot shows the AWS Amplify console interface. On the left, there's a sidebar with 'AWS Amplify' and 'PowerOfMath' selected. The main content area is titled 'Get started with Amplify Hosting'. It has a section for 'From your existing code' which includes options for GitHub, Bitbucket, GitLab, and AWS CodeCommit. Below this, there's a section for 'Deploy without Git provider' which is highlighted with an orange border. At the bottom right of this section is a 'Continue' button.

Naming the App (App name → Power of Math)

Upload index.zip file click on Save and Deploy

The screenshot shows the 'Start a manual deployment' page. It has fields for 'App name' (set to 'PowerOfMath') and 'Environment name' (set to 'dev'). Under 'Method', there are three options: 'Drag and drop' (selected and highlighted with an orange box), 'Amazon S3', and 'Any URL'. Below these methods is a large input field containing the file 'index.zip'. At the bottom right of the page is a 'Save and deploy' button, which is also highlighted with an orange box.

App has been Created

The screenshot shows the AWS Amplify console interface. On the left, a sidebar titled 'AWS Amplify' lists 'All apps' and 'PowerOfMath'. The main area is titled 'All apps' and shows a single item: 'PowerOfMath' with a status of 'Success' and a timestamp of '4 mins ago'. The entire 'PowerOfMath' card is highlighted with an orange border. The browser's address bar shows the URL: 'ap-southeast-2.console.aws.amazon.com/amplify/home?region=ap-southeast-2#/'. The system tray on the right indicates a weather of 30°C and a date/time of 01:09 PM 20-10-2023.

Go inside the App, click on the Domain

The screenshot shows the AWS Amplify console for the 'PowerOfMath' app. The left sidebar shows 'All apps' and 'PowerOfMath'. The main content area shows the 'PowerOfMath' app details, including its homepage listing deployed environments. Under 'Hosting environments', there is one entry: 'dev2350'. A deployment status bar shows 'Deployment successfully completed.' and a progress bar at 100%. Below this, the 'Domain' field is highlighted with an orange border, containing the URL 'https://dev2350.d2d1osjf84rei.amplifyapp.com'. The browser's address bar shows the URL: 'ap-southeast-2.console.aws.amazon.com/amplify/home?region=ap-southeast-2#/d2d1osjf84rei'. The system tray on the right indicates a weather of 30°C and a date/time of 01:09 PM 20-10-2023.

This tab lists all connected branches, select a branch to view build details.

dev2350

Deployment successfully completed.

Domain: <https://dev2350.d2d1osjf84srei.amplifyapp.com> Last deployment: 3, 1:05:18 PM

Open link in new tab (highlighted)

Open link in new window

Open link in incognito window

Open link as

Save link as...

Copy link address

Inspect

Drag and drop your project's build output directory or zip file here to update your app, or choose another method.

Choose files

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

TO THE POWER OF MATH!

Base number: ...to the power of: **CALCULATE**

30°C
ENG
01:10 PM
20-10-2023

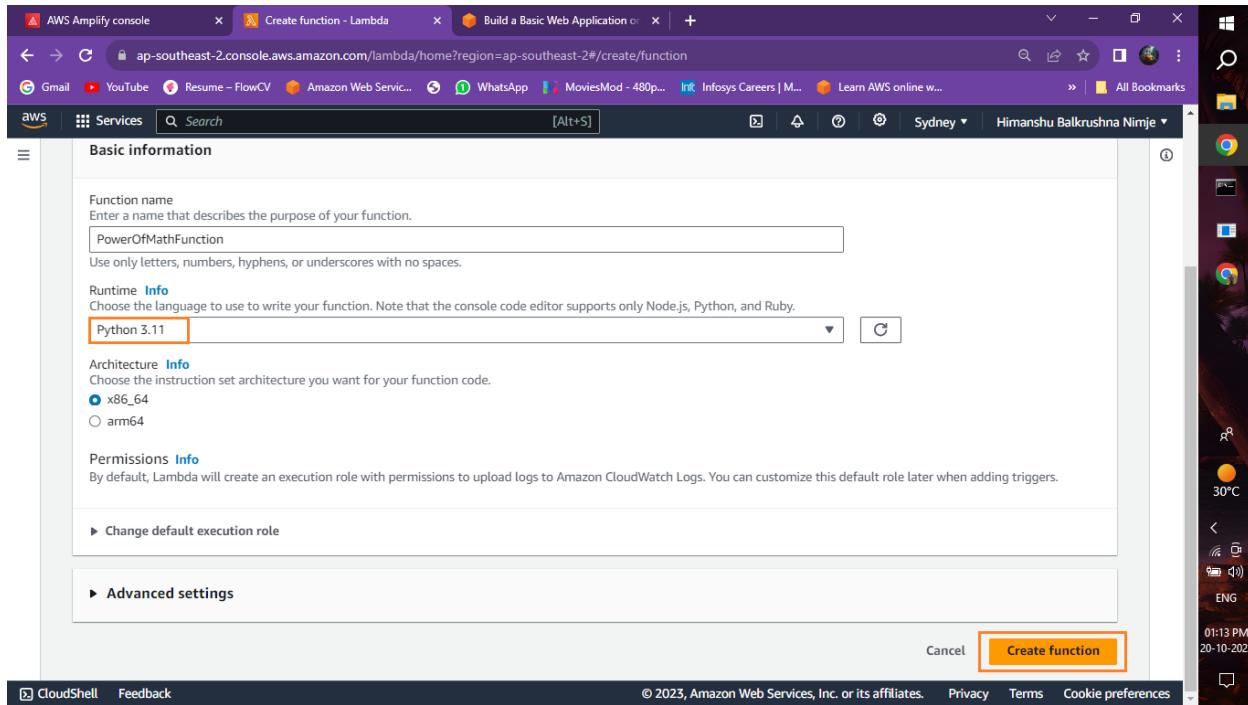
Open AWS console in New Tab, Search LAMBDA and click on

The screenshot shows the AWS Lambda search results page. The search bar at the top contains the query 'lambda'. The results are categorized under 'Services' and 'Features'. Under 'Services', the 'Lambda' service is highlighted with an orange border. Other services listed include CodeBuild, AWS Signer, and Amazon Inspector. Under 'Features', Local processing is listed as an IoT Core feature. On the right side of the page, there is a sidebar with 'Health' and 'Info' tabs, and a 'Go to AWS Health' button.

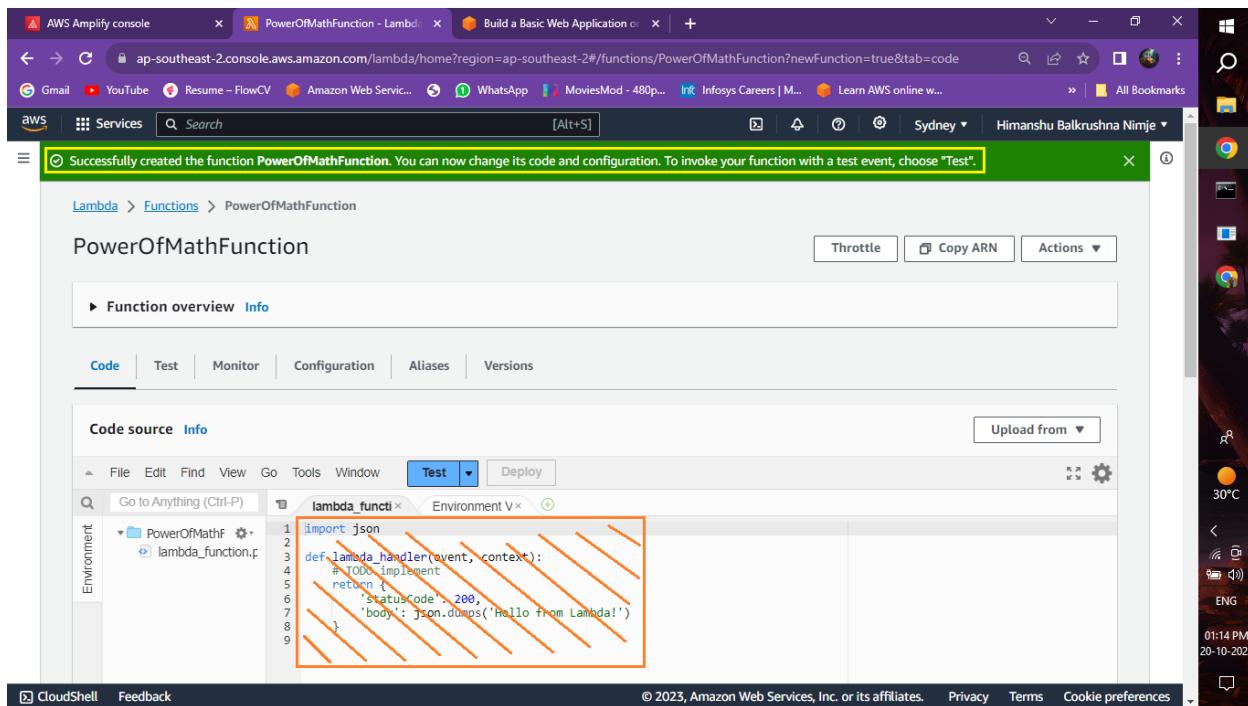
Create function

The screenshot shows the AWS Lambda Functions page. The left sidebar has 'AWS Lambda' selected. The main area displays a table titled 'Functions (0)' with columns for Function name, Description, Package type, Runtime, and Last modified. A message at the bottom of the table says 'There is no data to display.' An orange box highlights the 'Create function' button located at the top right of the table area.

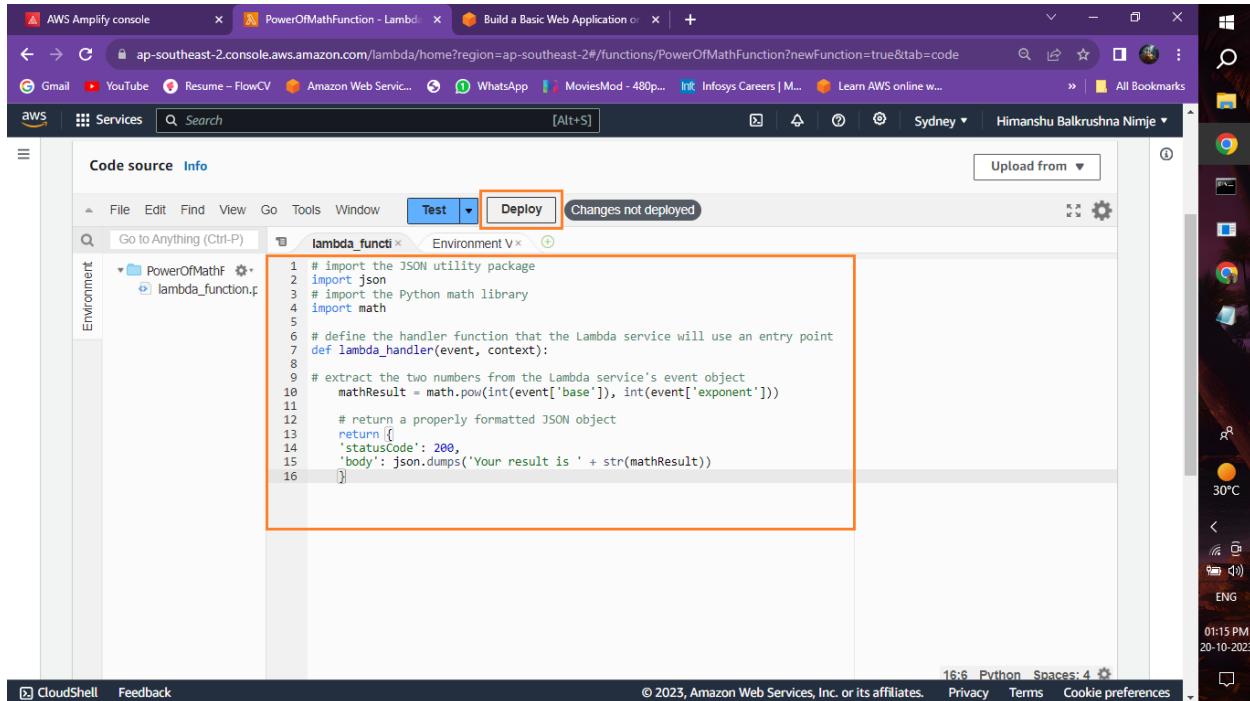
Naming the Function, Select Language, select Python and click on the Create Function



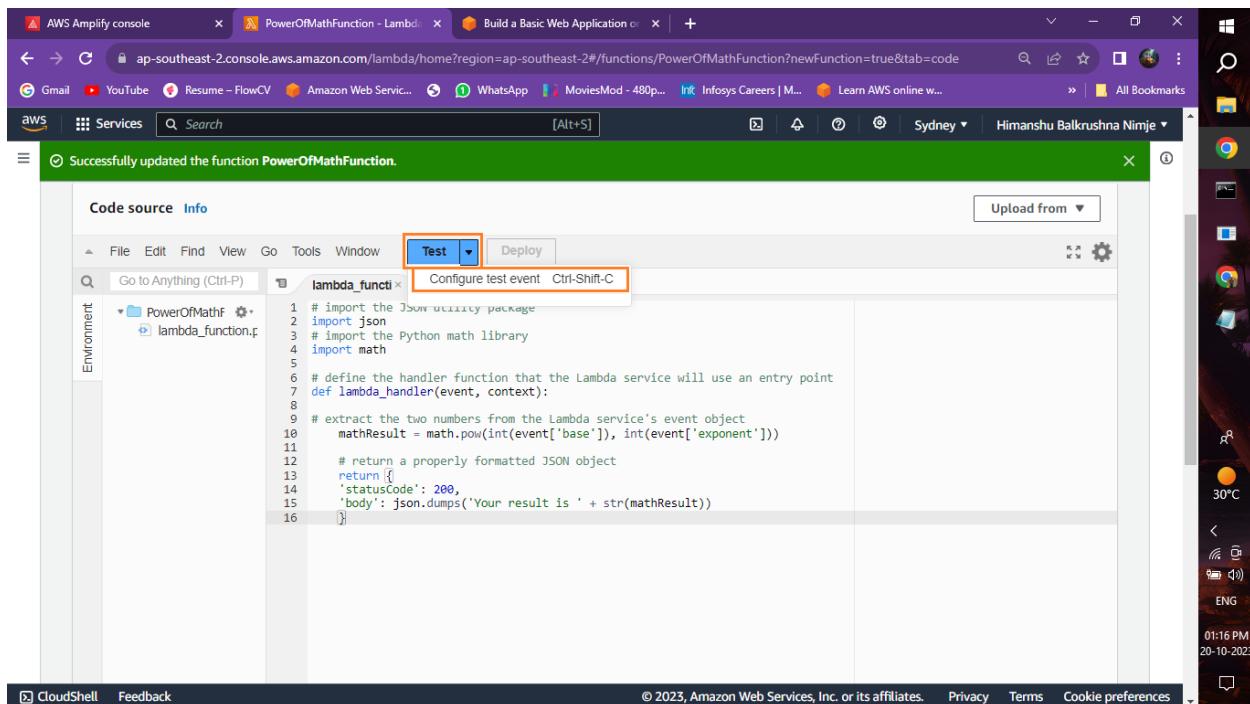
function is successfully created



In this function, Deploy Code, click on Deploy



Click on the Test



Response (status Code 200)and Result is 8.0 is Perfect

The screenshot shows the AWS Lambda console interface. A test execution for the 'PowerOfMathFunction' lambda function has been completed successfully. The response body is displayed as follows:

```

{
  "statusCode": 200,
  "body": "Your result is 8.0"
}
  
```

Open AWS console in New Tab, Search API GATEWAY and click on

The screenshot shows the AWS console search results for 'apigateway'. The 'API Gateway' service is highlighted with an orange box in the search results list. The search bar at the top contains 'apigateway'.

Click on the Get Started

The new API Gateway console experience is now available
We've redesigned the API Gateway console for REST APIs and WebSocket APIs. Try out the new console. As of 30th October 2023 the old console will no longer be available.

Amazon API Gateway

Amazon API Gateway helps developers to create and manage APIs to back-end systems running on Amazon EC2, AWS Lambda, or any publicly addressable web service. With Amazon API Gateway, you can generate custom client SDKs for your APIs, to connect your back-end systems to mobile, web, and server applications or services.

Get Started

Getting Started Guide

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Select REST API → Because This Provide (Control over the request and response along with API Management Capabilities)

The new API Gateway console experience is now available
We've redesigned the API Gateway console for REST APIs and WebSocket APIs. Try out the new console. As of 30th October 2023 the old console will no longer be available.

API Gateway

APIs Custom domain names VPC links

Use the new console
As of 30th October 2023 the old console will no longer be available.

REST API
Develop a REST API where you gain complete control over the request and response along with API management capabilities.
Works with the following:
Lambda, HTTP, AWS Services

REST API Private
Create a REST API that is only accessible from within a VPC.
Works with the following:
Lambda, HTTP, AWS Services

Import Build Import Build

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

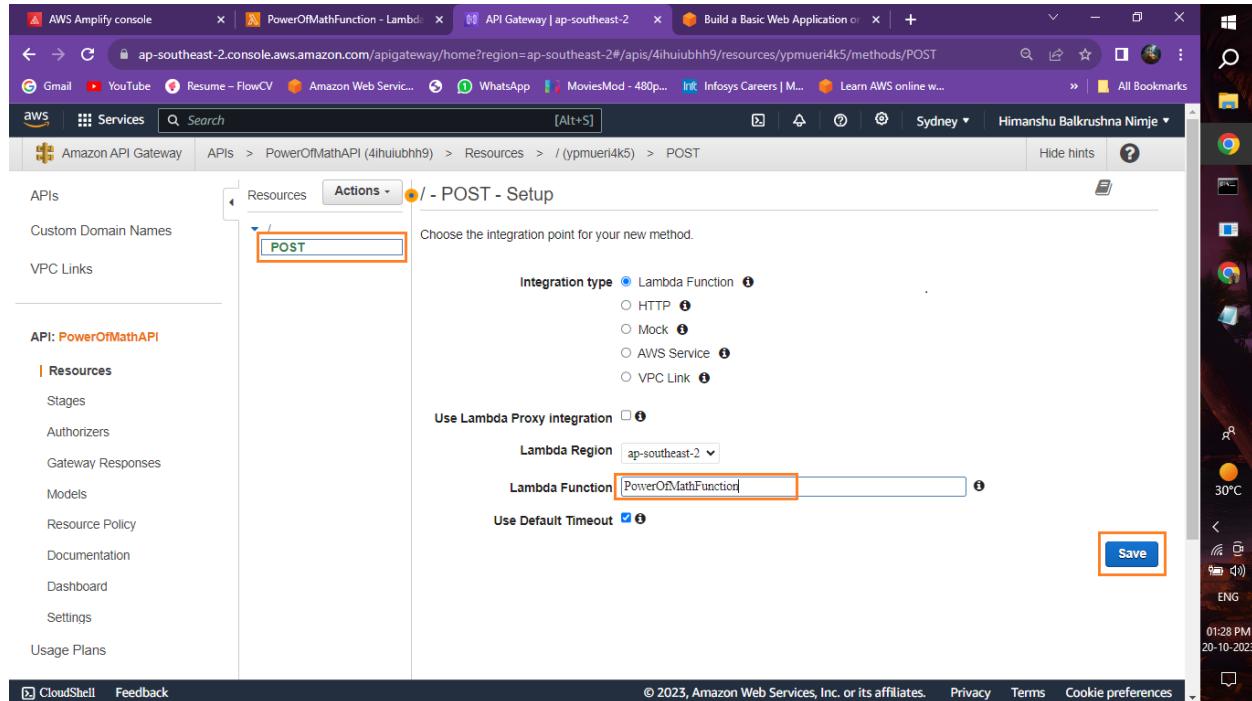
Choose protocol, Create new API, Assign API name and click on Create API

The screenshot shows the 'Create new API' interface in the AWS API Gateway console. The 'API name' field contains 'PowerOfMathAPI' and the 'Endpoint Type' dropdown is set to 'Regional'. The 'Create API' button is highlighted with a blue box.

Here PowerofMathAPI, Go to Resources, and click on Actions

The screenshot shows the 'Resources' page for the 'PowerOfMathAPI' API. The left sidebar shows the API name and various resource categories like Stages, Authorizers, and Models. The 'Actions' button in the top navigation bar is highlighted with an orange box.

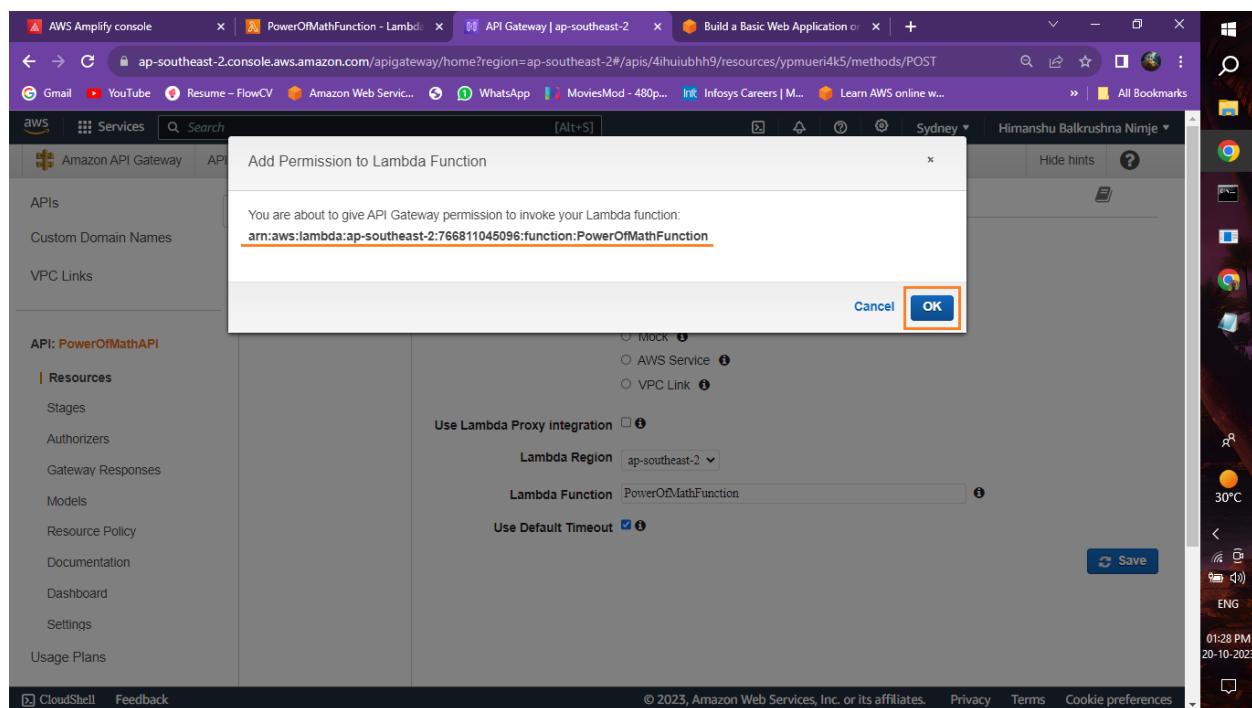
After click on Actions the post option will appear
 Select the integration type, select the lambda function, and click on save



The screenshot shows the AWS API Gateway console with the following details:

- APIs**: PowerOfMathFunction - Lambda
- Actions**: / - POST - Setup
- Integration type**: Lambda Function (selected)
- Lambda Region**: ap-southeast-2
- Lambda Function**: PowerOfMathFunction
- Use Default Timeout**: checked
- Save** button highlighted with a red box

This is the ARN link that allows the Lambda function



The screenshot shows the AWS API Gateway console with the following details:

- Add Permission to Lambda Function** dialog is open.
- Message: "You are about to give API Gateway permission to invoke your Lambda function:
`arn:aws:lambda:ap-southeast-2:766811045096:function:PowerOfMathFunction`
- OK** button highlighted with a red box
- Integration type**: Lambda Function (selected)
- Lambda Region**: ap-southeast-2
- Lambda Function**: PowerOfMathFunction
- Use Default Timeout**: checked
- Save** button visible at the bottom right

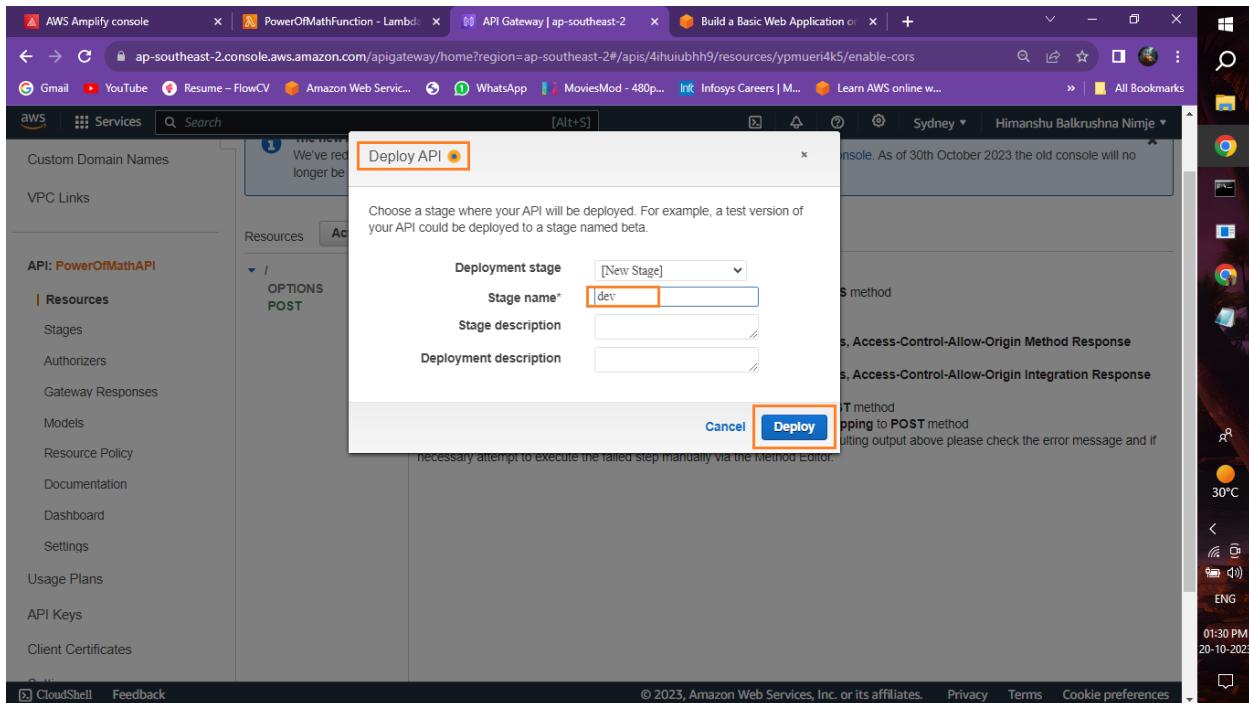
Click on Enable CORS

The screenshot shows the AWS API Gateway console for the 'PowerOfMathFunction - Lambda' API. The 'POST' method under the '/' resource is selected. In the 'Actions' dropdown, 'Enable CORS' is highlighted. The 'Access-Control-Allow-Origin' field contains the value '*' and has a warning icon. A blue button at the bottom right of the form says 'Enable CORS and replace existing CORS headers'. The status bar at the bottom indicates 'CloudShell Feedback'.

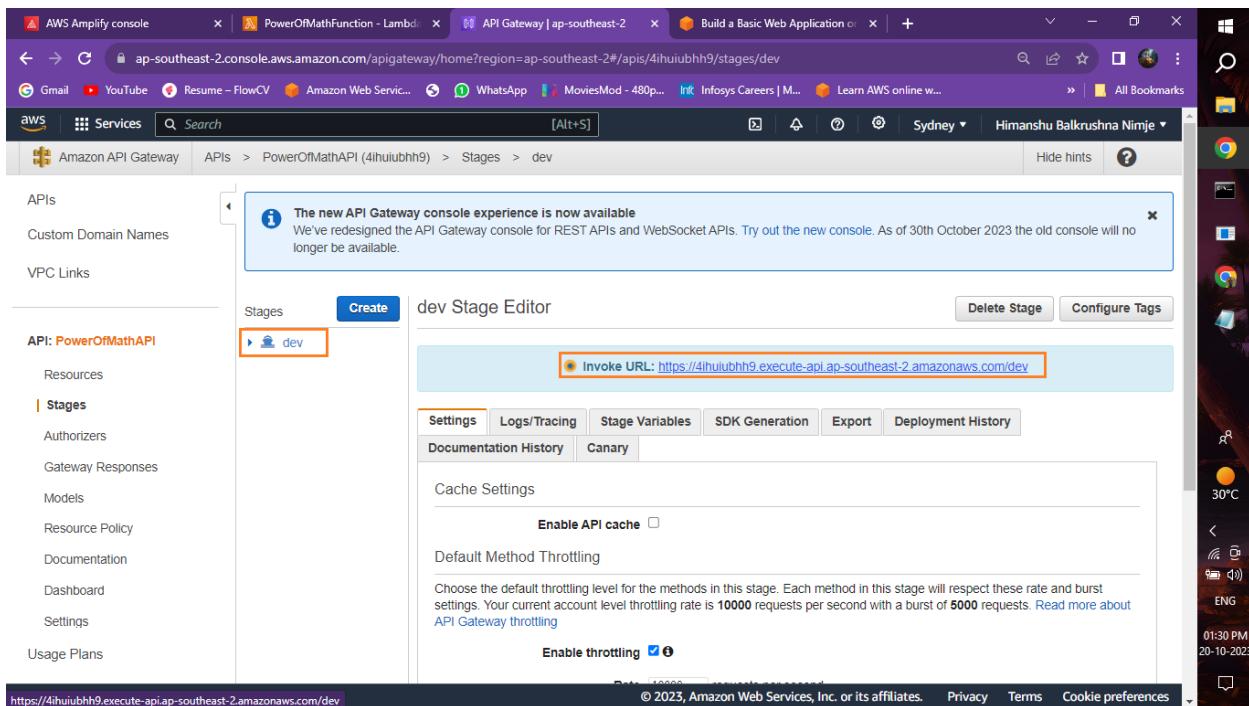
The screenshot shows the AWS API Gateway console after enabling CORS. A summary box lists the following changes:

- ✓ Create OPTIONS method
- ✓ Add 200 Method Response with Empty Response Model to OPTIONS method
- ✓ Add Mock Integration to OPTIONS method
- ✓ Add 200 Integration Response to OPTIONS method
- ✓ Add Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Method Response Headers to OPTIONS method
- ✓ Add Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Integration Response Header Mappings to OPTIONS method
- ✓ Add Access-Control-Allow-Origin Method Response Header to POST method
- ✓ Add Access-Control-Allow-Origin Integration Response Header Mapping to POST method

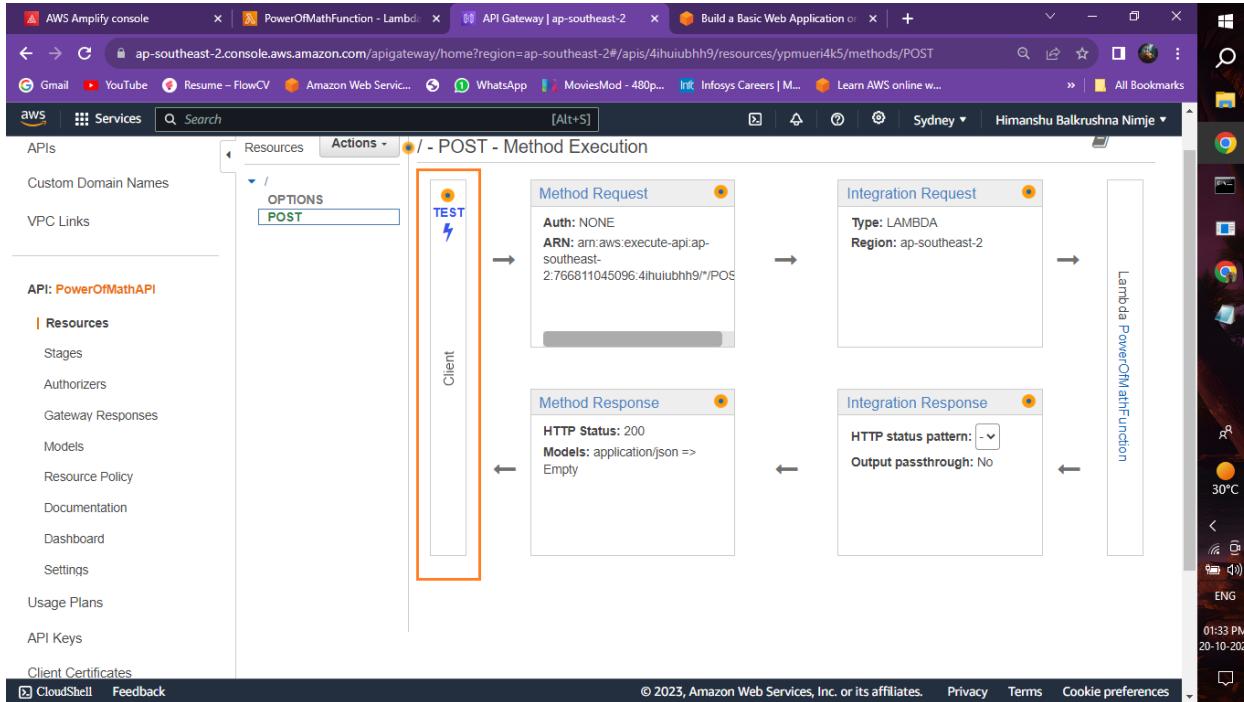
Deploy API, Set Stage, Click on the Deploy



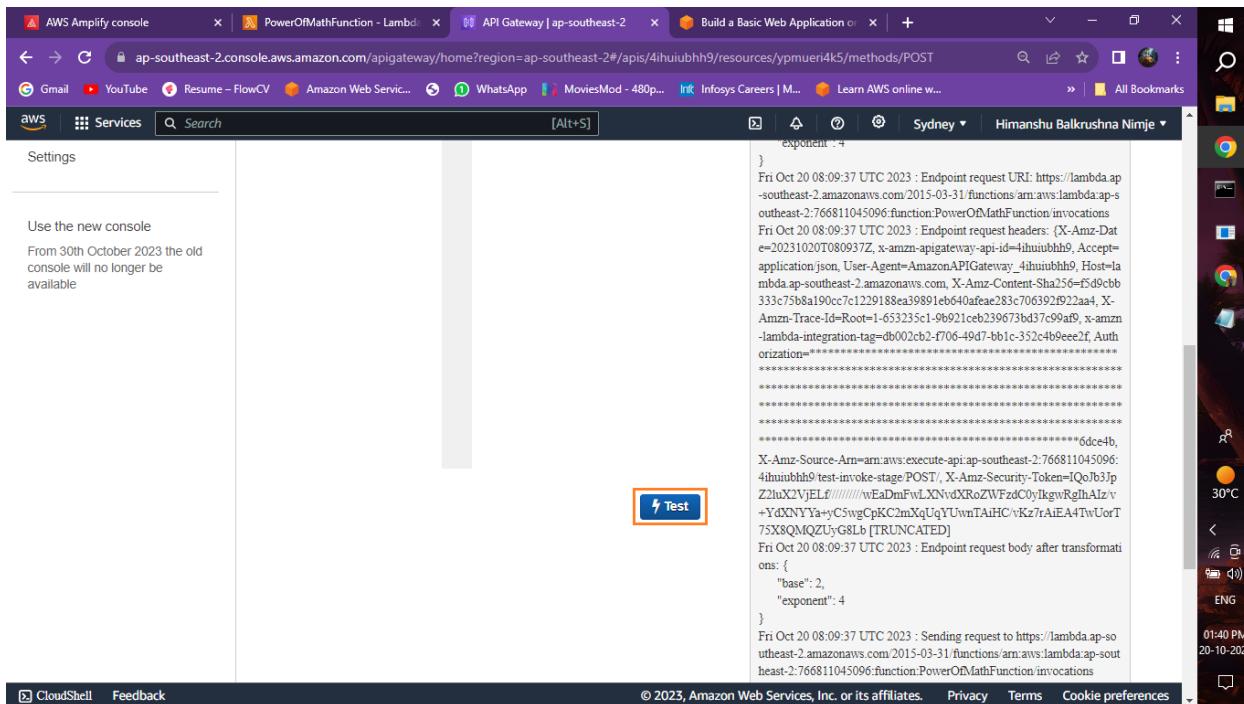
Here is API dev invoke URL



Test the Post



Click on TEST



Response should come in the POST Method test → Status 200, Body and Result is 16.0 is Perfect

The screenshot shows the AWS API Gateway console for the 'PowerOfMathFunction' Lambda function. In the 'Actions' dropdown, 'Method Execution' is selected, and the 'POST' method is chosen. The 'Method Test' section displays the following results:

- Request:** /
- Status:** 200
- Latency:** 40 ms
- Response Body:** `{"statusCode": 200, "body": "Your result is 16.0"}`
- Response Headers:** `{"Access-Control-Allow-Origin": ["*"], "Content-Type": ["application/json"], "X-Amzn-Trace-Id": ["Root=1-653235ea-9130ec367a9be0292b854582;Sampled=0;image=5a352b30.0"]}`
- Logs:** Execution log for request 53c946e5-43d6-4a2c-b05e-6dee9a0c7365

The screenshot shows the AWS API Gateway console for the 'PowerOfMathFunction' Lambda function. A context menu is open over the 'Actions' dropdown, with the 'Open link in new tab' option highlighted. A message box at the top right of the page states: 'The new API Gateway console experience is now available. We've redesigned the API Gateway console for REST APIs and WebSocket APIs. Try out the new console. As of 30th October 2023 the old console will no longer be available.'

The main interface shows the 'POST - Method Execution' section with the following flow diagram:

```

graph LR
    Client[Client] --> MethodRequest[Method Request]
    MethodRequest --> IntegrationRequest[Integration Request]
    IntegrationRequest --> Lambda[Lambda PowerOfMathFunction]
    Lambda --> MethodResponse[Method Response]
    MethodResponse --> IntegrationResponse[Integration Response]
    
```

The 'Method Request' panel shows:

- Auth: NONE
- ARN: arn:aws:execute-api:ap-southeast-2:2766811045096.4huiubhh9/*/POST

The 'Method Response' panel shows:

- HTTP Status: 200
- Models: application/json => Empty

The 'Integration Response' panel shows:

- HTTP status pattern:
- Output passthrough: No

Open AWS console in New Tab, Search DynamoDB and click on

The screenshot shows the AWS Console search results for 'Dynamodb'. The 'DynamoDB' service card, which is described as a 'Managed NoSQL Database', is highlighted with a red box. Other search results include 'Athena' (Serverless interactive analytics service) and sections for 'Features' (Settings, Clusters, Subnet Groups) and 'Documentation' (6,740 results). On the right side of the screen, there is a sidebar with various monitoring and health metrics for the service.

Create Table

The screenshot shows the Amazon DynamoDB service dashboard. The main heading is 'Amazon DynamoDB: A fast and flexible NoSQL database service for any scale'. Below it, a sub-section titled 'How it works' includes a link to 'What is Amazon DynamoDB?'. To the right, there is a 'Get started' box with a 'Create table' button, which is highlighted with a red box. The left sidebar lists various management options like 'Tables', 'Update settings', and 'Exports to S3'.

Assign Table Name

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The 'Table details' step is active. In the 'Table name' section, the input field contains 'PowerOffMathDatabase'. Below it, a note specifies that the name must be between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.). The 'Partition key' section shows 'ID' as the primary key of type 'String'. The 'Sort key - optional' section is empty. At the bottom, there's a 'Table settings' section and a footer with standard AWS links.

Click on Create Table

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The 'Table settings' step is active. It displays various configuration options: Capacity mode (Provisioned), Provisioned read capacity (5 RCU), Provisioned write capacity (5 WCU), Auto scaling (On), Local secondary indexes (-), Global secondary indexes (-), Encryption key management (Owned by Amazon DynamoDB), and Deletion protection (Off). Below these, the 'Tags' section indicates no tags are associated with the resource and provides a button to 'Add new tag'. At the bottom right, there are 'Cancel' and 'Create table' buttons, with the latter being highlighted.

Table is ready

The screenshot shows the AWS DynamoDB console interface. On the left, a sidebar menu includes 'Dashboard', 'Tables' (selected), 'Update settings', 'Explore items', 'PartiQL editor', 'Backups', 'Exports to S3', 'Imports from S3', 'Reserved capacity', and 'Settings'. Under 'Tables', there's a 'DAX' section with 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main content area displays a message: 'Creating the PowerOfMathDatabase table. It will be available for use shortly.' Below this, a table titled 'Tables (1) Info' lists one entry: 'PowerOfMathDatabase' with status 'Creating', partition key 'ID (S)', sort key '-', provisioned read capacity '0', and deletion protection 'Off'. The status 'Creating' is highlighted with an orange border.

Go inside to PowerofmathDatabase, The Table status is Active

The screenshot shows the AWS DynamoDB console interface for the 'PowerOfMathDatabase' table. The left sidebar is identical to the previous screenshot. The main content area shows the 'Overview' tab of the table details. A callout box highlights the 'Table status' field, which is set to 'Active' with a green checkmark icon. Other visible information includes the partition key 'ID (String)', sort key '-', capacity mode 'Provisioned', and point-in-time recovery (PITR) status 'Off'. The status 'Active' is also highlighted with an orange border.

Go to PowerofMathFunction

PowerOfMathFunction

Function overview Info

Code Test Monitor Configuration Aliases Versions

General configuration Triggers Permissions Destinations Function URL Environment variables Tags VPC Monitoring and operations tools

Execution role

Role name: PowerOfMathFunction-role-zche5t8g

Resource summary

To view the resources and actions that your function has permission to access, choose a service.

Amazon CloudWatch Logs 3 actions, 2 resources

By action By resource

Resource	Actions
arn:aws:logs:ap-southeast-2:766811045096:log-group:/aws/Lambda/PowerOfMathFunction*	Allow: logs>CreateLogGroup
arn:aws:logs:ap-southeast-2:766811045096:log-group:/aws/Lambda/PowerOfMathFunction*	Allow: logs>CreateLogStream

Create IAM role

The screenshot shows the AWS IAM Roles page. The left sidebar is collapsed. The main area displays the details for the 'PowerOfMathFunction-role-zche5t8g' role. The 'Summary' section includes the ARN: arn:aws:iam::766811045096:role/service-role/PowerOfMathFunction-role-zche5t8g. The 'Permissions' tab is selected, showing one managed policy attached. The 'Add permissions' button is visible. The bottom right corner shows a Windows taskbar with the date and time.

Create inline policy

The screenshot shows the same AWS IAM Roles page as the previous one, but with the 'Create inline policy' button highlighted in orange. This button is located in the 'Permissions policies' section of the 'Permissions' tab. The rest of the interface and the Windows taskbar are identical to the previous screenshot.

The screenshot shows the AWS IAM 'Create policy' interface. The left sidebar shows 'Step 1 Specify permissions' and 'Step 2 Review and create'. The main area is titled 'Specify permissions' with the sub-section 'Policy editor'. A JSON code editor contains the following policy:

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "VisualEditor0",
6       "Effect": "Allow",
7       "Action": [
8         "dynamodb:PutItem",
9         "dynamodb>DeleteItem",
10        "dynamodb:GetItem",
11        "dynamodb:Scan",
12        "dynamodb:Query",
13        "dynamodb:UpdateItem"
14      ],
15      "Resource": "YOUR-TABLE-ARN"
16    }
17  ]
18 }

```

A red box highlights the first few lines of the JSON code. To the right of the code editor is a panel titled 'Edit statement' with a 'paste it' button. Below the code editor, a status bar indicates '9951 of 10240 characters remaining'.

Click on the Next

The screenshot shows the same AWS IAM 'Create policy' interface. The JSON code editor now includes a 'Resource' field with the value 'arn:aws:dynamodb:ap-southeast-2:766811045096:table/PowerOfMat'. A red box highlights the 'Resource' field. The status bar at the bottom of the code editor shows '9951 of 10240 characters remaining'. At the bottom right of the screen, there is a 'Next' button, which is also highlighted with a red box.

Naming the policy name, and select required policies

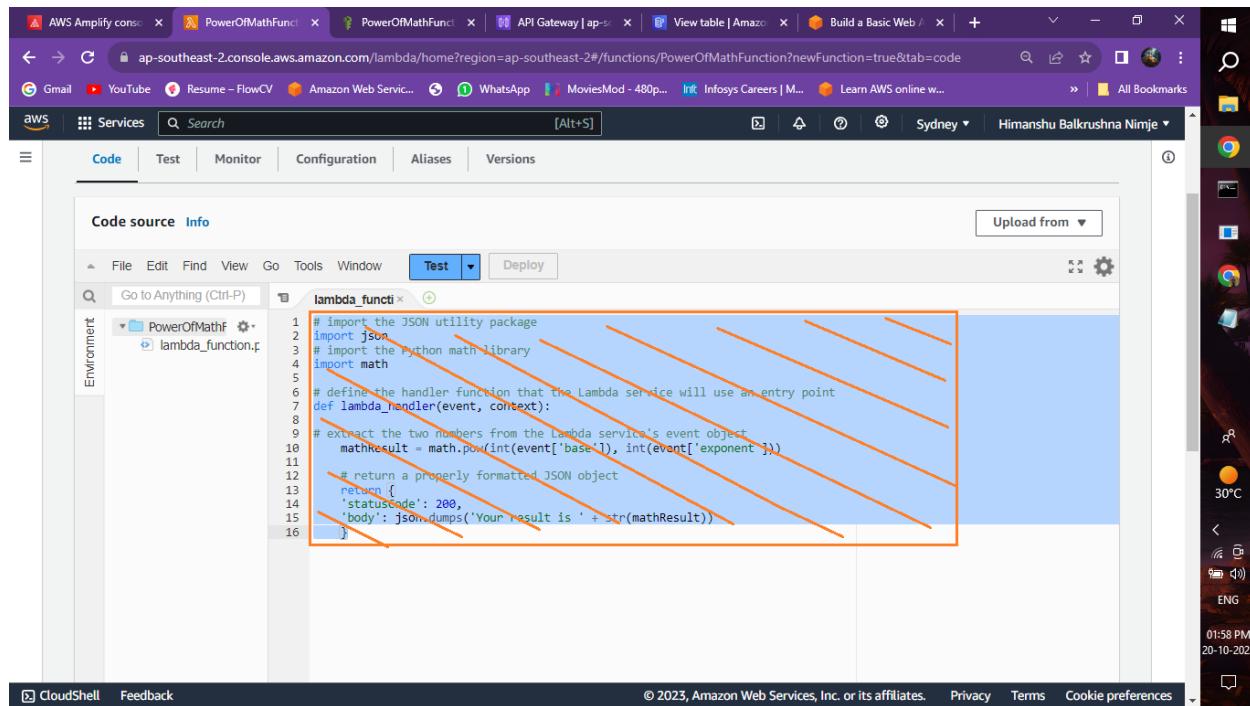
The screenshot shows the 'Review and create' step of creating a new IAM policy. The 'Policy details' section has a 'Policy name' input field containing 'PowerOfMathDynamoPolicy'. Below it, a table lists a single permission: 'Allow (1 of 384 services)' for 'DynamoDB' with 'Limited: Read, Write' access and 'None' request condition. The table has columns for Service, Access level, Resource, and Request condition.

Service	Access level	Resource	Request condition
DynamoDB	Limited: Read, Write	region string like ap-southeast-2, TableName string like PowerOfMathDatabase	None

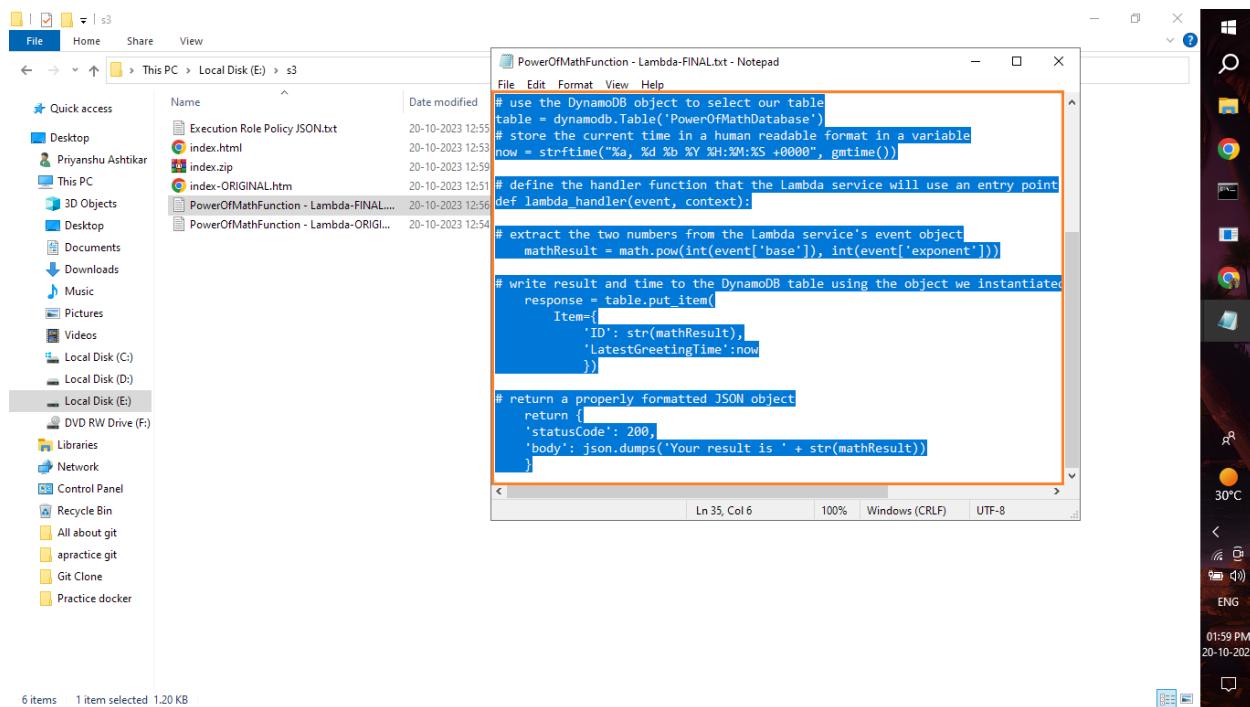
And click on Create policy

The screenshot shows the same 'Review and create' step as before, but the 'Create policy' button at the bottom right is highlighted in orange, indicating it is the next action to be taken.

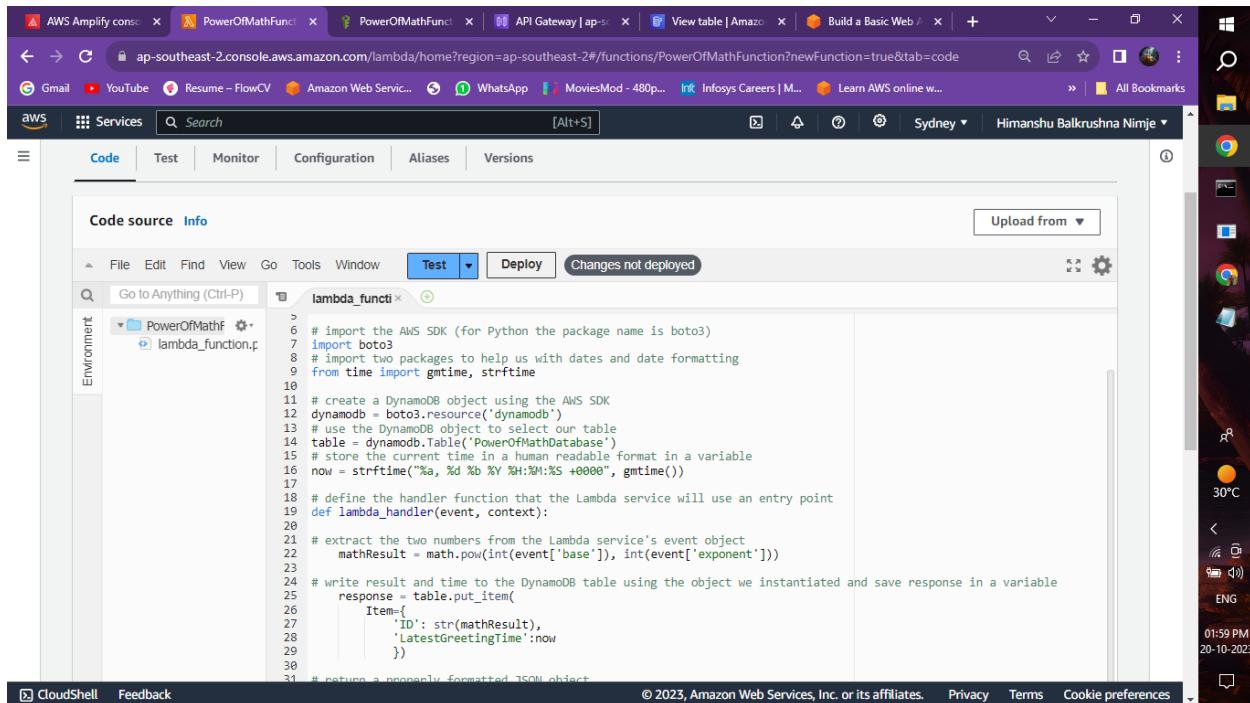
Go inside to the lambda function



copy the code



Paste on the LAMBDA→ Code (Code Source) option



```

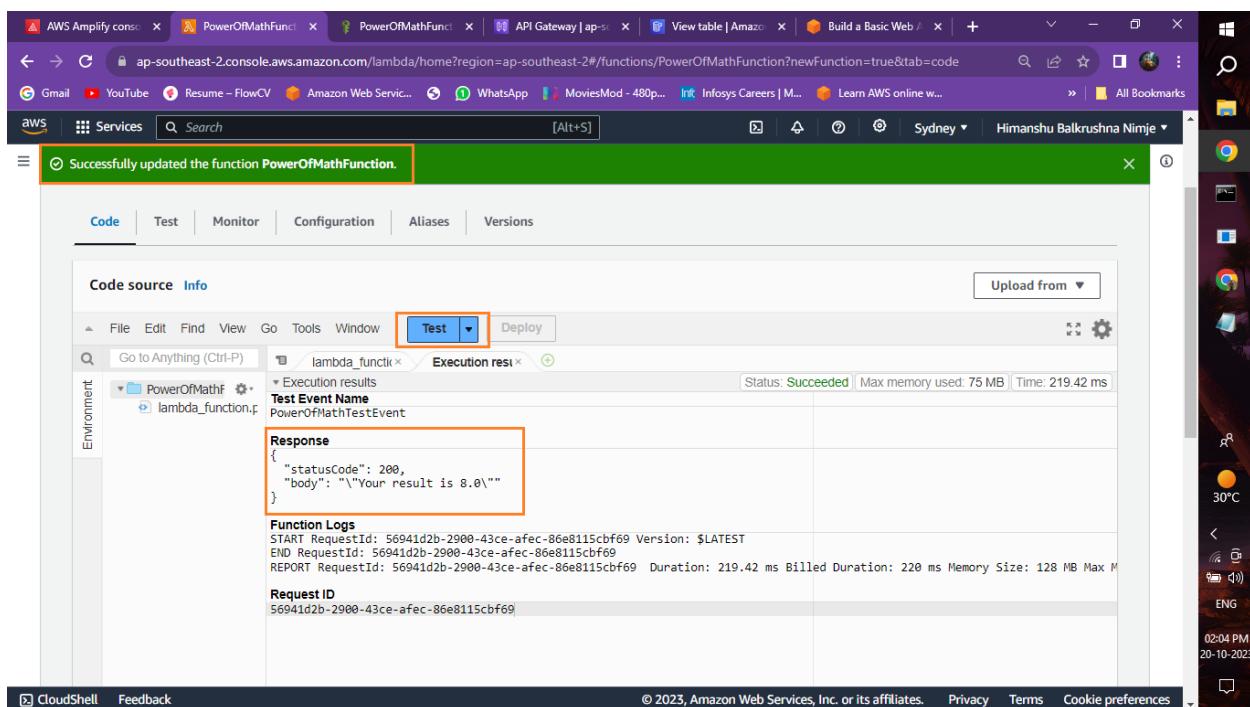
AWS Amplify console | PowerOfMathFunc | PowerOfMathFunc | API Gateway | ap-southeast-2 | View table | Amazon Web Services | Build a Basic Web | + | - | X
Gmail YouTube Resume – FlowCV Amazon Web Servic... WhatsApp MoviesMod - 480p... Infosys Careers | M... Learn AWS online w...
Services Search [Alt+S] Sydney Himanshu Balkrushna Nimje
Code Test Monitor Configuration Aliases Versions
Code source Info Upload from
File Edit Find View Go Tools Window Test Deploy Changes not deployed
Go to Anything (Ctrl-P)
lambda_funcx
1>
2# import the AWS SDK (for Python the package name is boto3)
3import boto3
4# import two packages to help us with dates and date formatting
5from time import gmtime, strftime
6
7# create a DynamoDB object using the AWS SDK
8dynamodb = boto3.resource('dynamodb')
9# use the DynamoDB object to select our table
10table = dynamodb.Table('PowerOfMathDatabase')
11# store the current time in a human readable format in a variable
12now = strftime("%a, %d %b %Y %H:%M:%S +0000", gmtime())
13
14# define the handler function that the Lambda service will use as an entry point
15def lambda_handler(event, context):
16
17    # extract the two numbers from the Lambda service's event object
18    mathResult = math.pow(int(event['base']), int(event['exponent']))
19
20    # write result and time to the DynamoDB table using the object we instantiated and save response in a variable
21    response = table.put_item(
22        Item={
23            'ID': str(mathResult),
24            'LatestGreetingTime': now
25        }
26    )
27
28    # return a properly formatted JSON object
29
30
31# return a properly formatted JSON object

```

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on Test

Here is status show Succeeded, status code is 200 and Result is 8.0



Successfully updated the function PowerOfMathFunction.

Code source Info Test Deploy

Execution results Status: Succeeded Max memory used: 75 MB Time: 219.42 ms

Test Event Name PowerOfMathTestEvent

Response

```
{
  "statusCode": 200,
  "body": "Your result is 8.0"
}
```

Function Logs

```
START RequestId: 56941d2b-2900-43ce-afec-86e8115cbf69 Version: $LATEST
END RequestId: 56941d2b-2900-43ce-afec-86e8115cbf69
REPORT RequestId: 56941d2b-2900-43ce-afec-86e8115cbf69 Duration: 219.42 ms Billed Duration: 220 ms Memory Size: 128 MB Max Memory Used: 75.00 MB
```

Request ID 56941d2b-2900-43ce-afec-86e8115cbf69

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

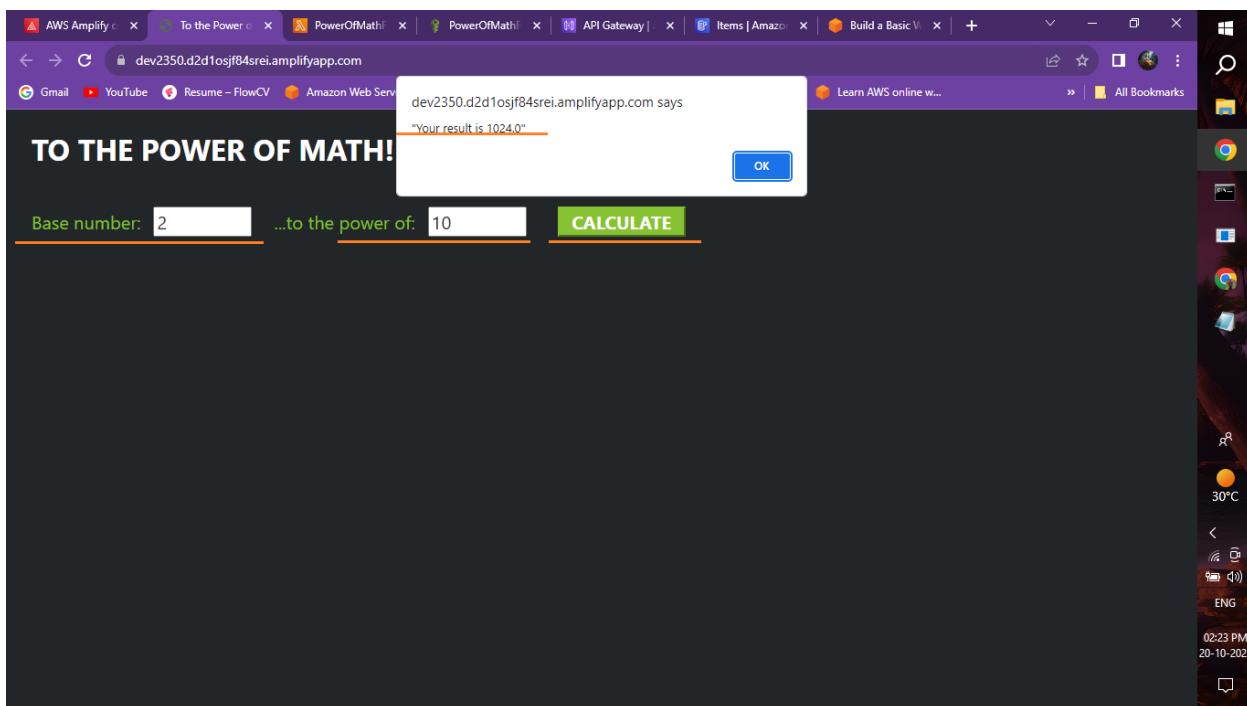
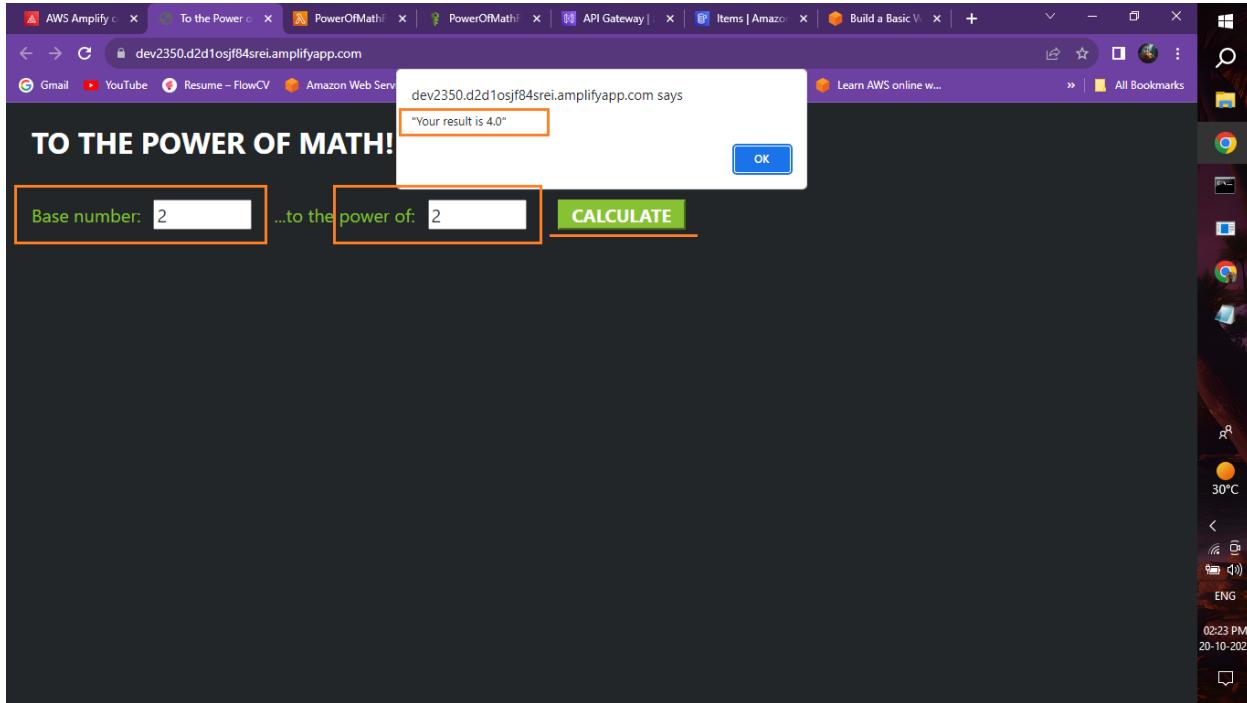
Go to DynamoDB, Go to PowerofMathDatabase

The screenshot shows the AWS DynamoDB console interface. On the left, a sidebar menu includes 'Dashboard', 'Tables' (selected), 'Update settings' (highlighted with a red box), 'Explore items', 'PartiQL editor', 'Backups', 'Exports to S3', 'Imports from S3', 'Reserved capacity', and 'Settings'. Under 'Tables', 'PowerOfMathDatabase' is selected. The main panel displays the 'PowerOfMathDatabase' table details. The 'General information' section shows a Partition key 'ID (String)', Sort key '-', Capacity mode 'Provisioned', and Table status 'Active'. The 'Additional info' section shows Table class 'DynamoDB Standard', Indexes '0 globals, 0 locals', DynamoDB stream 'Off', Time to Live (TTL) 'Off', Replication Regions '0 Regions', Encryption 'Owned by Amazon', Date created 'October 20, 2023, 13:45:14 (UTC+05:30)', and Deletion protection 'Off'. The URL in the browser bar is <https://ap-southeast-2.console.aws.amazon.com/dynamodbv2/home?region=ap-southeast-2#item-explorer?maximize=true&table=PowerOfMathDatabase>.

Go to Explore items and Click on the Run Option

The screenshot shows the 'Scan or query items' interface in the AWS DynamoDB console. The sidebar menu is identical to the previous screenshot. The main panel has 'Scan' selected under 'Scan or query items' and the 'Run' button highlighted with a red box. A message box at the bottom says 'Completed. Read capacity units consumed: 0.5'. Below it, the 'Items returned (1)' section shows a single item with ID '8.0' and LatestGreetingTime 'Fri, 20 Oct 2023 08:33:54 +0000'. The URL in the browser bar is <https://ap-southeast-2.console.aws.amazon.com/dynamodbv2/home?region=ap-southeast-2#item-explorer?maximize=true&table=PowerOfMathDatabase>.

Here the APP has been successfully launched
Put Base number and Power, click on calculate



What I Learned

1. AWS Amplify:
Frontend Deployment: I have learned how to use AWS Amplify to deploy and host web application's frontend. Amplify simplifies the process of setting up and managing the frontend infrastructure, providing features like continuous deployment, hosting, and scalability.
2. IAM Roles (Identity and Access Management):
Security Management: I have gained an understanding of IAM roles and how they are used to securely control access to AWS services and resources. IAM allows you to manage permissions and access policies for users, groups, and roles.
3. Amazon DynamoDB:
NoSQL Database: I have learned how to use DynamoDB as a NoSQL database to store and retrieve data for your application. DynamoDB is a fully managed, scalable, and highly available database service provided by AWS.
4. Lambda Function:
Serverless Computing: I have learned how to create and deploy AWS Lambda functions. Lambda allows you to run your code without provisioning or managing servers, making it a key component of serverless architecture.
5. Amazon API Gateway:
API Management: I have learned how to set up and configure API Gateway to create, publish, and manage APIs for your application. API Gateway enables you to create RESTful APIs that can integrate with various AWS services and external systems.

By combining these services, I have created a scalable and serverless architecture for web applications, leveraging the strengths of each service to build a robust and efficient system. I have also gained insights into managing security, databases, serverless computing, and API management on the AWS platform.