AWS Project

1. We need to create and host a web page
2. A way to invoke math functions
3. A way to do some math
4. To store and return the math result
5. To handle the permissions

AWS Services

1. Amplify – used to build and host websites

First, we need to create our web page from the visual studio code. Then go to the AWS Amplify service and then click on

Host a web page, then enter the app name and branch name and then upload the file.

You will get a domain after uploading the app where you get to view your app in a web page.

1. Lambda - To run code or function (serverlessly) upon some trigger.

First, go to the lambda and create a function with name PowerOfMath and set Python 3.9 as Runtime.

Enter this code in Lambda

# import the JSON utility package

import json

# import the Python math library

import math

# define the handler function that the Lambda service will use an entry point

def lambda\_handler(event, context):

# extract the two numbers from the Lambda service's event object

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

# return a properly formatted JSON object

return {

'statusCode': 200,

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

}

The above code is to find the power of a number; to test the above function we need to create an event and give the key and value in JSON

Key should be same as the variable given in the Code.

Base: 2.

Exponent: 3

1. API Gateway: used to build HTTP, REST and WebSocket APIs. To invoke the function

Create a REST API with the name of PowerOfMath

Go to Resources with backslash on the and click on create a method and select the method name as POST and then give the permission to the Lambda Function (linking the API to the lambda function)

POST method is used

The POST method is typically used to send data to the server to create a new resource. For example, when a user submits a form on a website, the data is often sent to the server using a POST request to create a new entry in a database.

Then in the Resource Sharing enable **Cross-Origin Resource Sharing (CORS)**

Enabling CORS allows the server to specify which domains are permitted to access its resources, thereby allowing cross-origin requests.

1. Dynamo DB: (Key- Value NO SQL Database or Relational Database)

To store the result in a database

Create a database in DynamoDB and name the partition key as ID

(The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.)

After creating Table, copy the ARN (Amazon Resource Name)

Go back to Lambda Function and then go to permissions and click on Execution Role

1. IAM: (Set Permissions on the execution role for Lambda)

After you go to the tab execution role has popped we can see permissions and then you can add inline policy and then edit the specify permissions in the JSON and then copy paste the table ARN to add permissions to the table.

Then review the policy and then click on create policy

Now we are set that the lambda function can directly access the database.

Rewrite the code in the Lambda to update the values in the table and then test it.

After you test the code, the result gets stored in the DynamoDB table.

After editing the index.html file with the API gateway then redeploy the file.

A diagram of a company

Description automatically generated