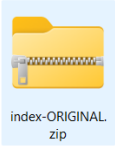
AWS Project: Architect and Build an End-to-End AWS Web Application from Scratch

For this AWS project, I will create a simple calculator to compute powers using a given base and exponent. The purpose of this project is to piece together 5 different AWS Services to run and deploy a simple math application. This application needs a way to create and host a webpage (AWS Amplify), invoke the math functionality (API Gateway), do the calculation (Lambda), store and return the results (DynamoDB), and handle permissions (IAM).

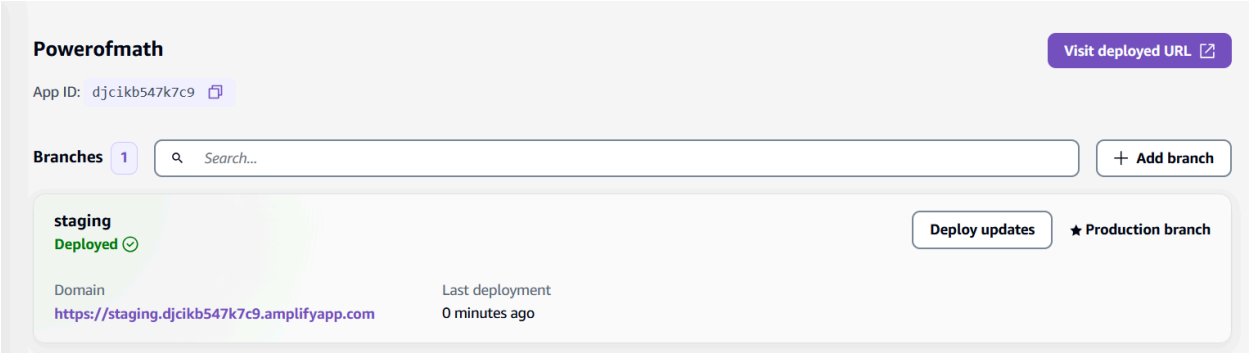
Create and Host a Webpage

To create and host a webpage for the application, AWS Amplify will be used. AWS Amplify is a managed service used to build and host websites easily. I initially created a basic, bare-bones html file:

Then compressed the html file into a ZIP.

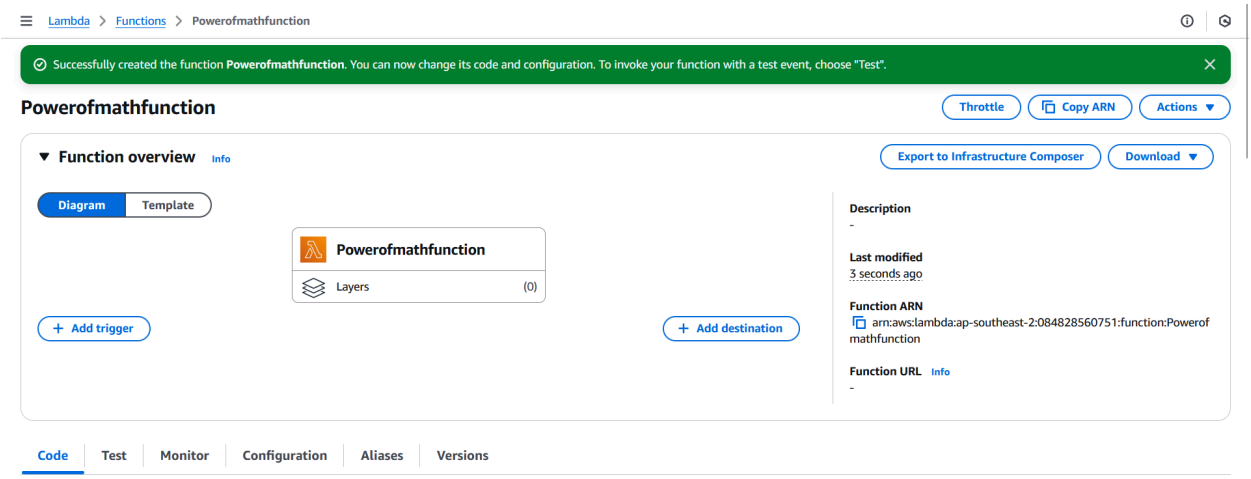


On AWS Amplify, I uploaded the ZIP file, named it Powerofmath, and deployed the application.

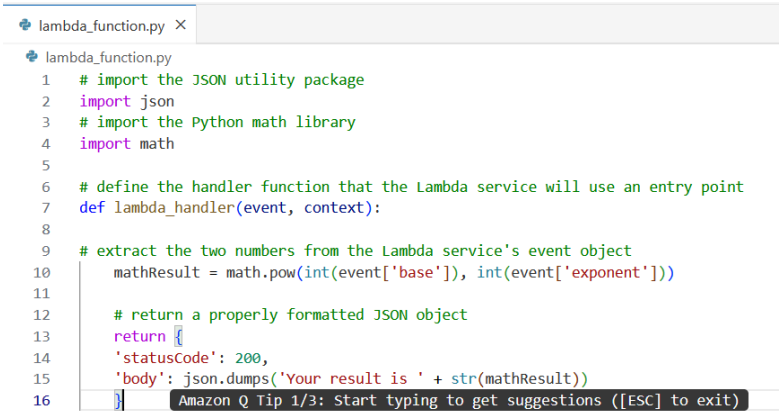


Calculate the Result

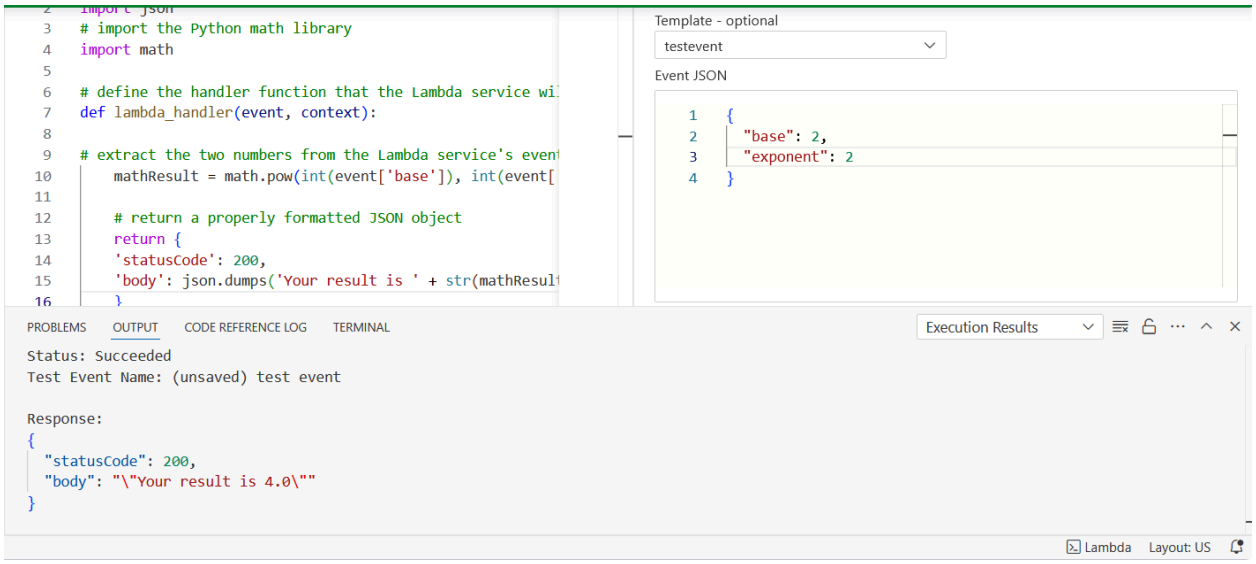
AWS Lambda will be used to execute the underlying calculations of the application. I first created the “Powerofmath” Lambda function to handle this.



For the logic of th lambda function, I specified the following:

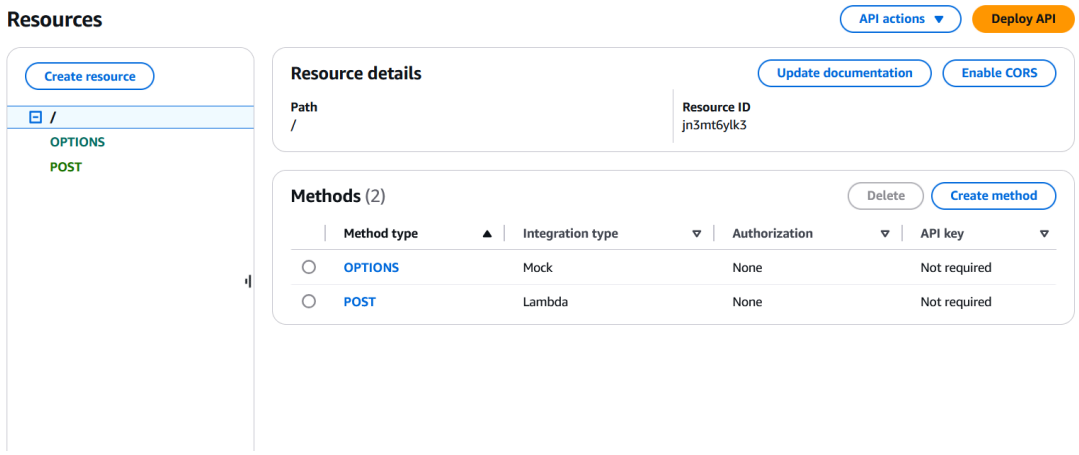


Once that was complete, I tested out the function by passing in test values and validated if the output was correct.

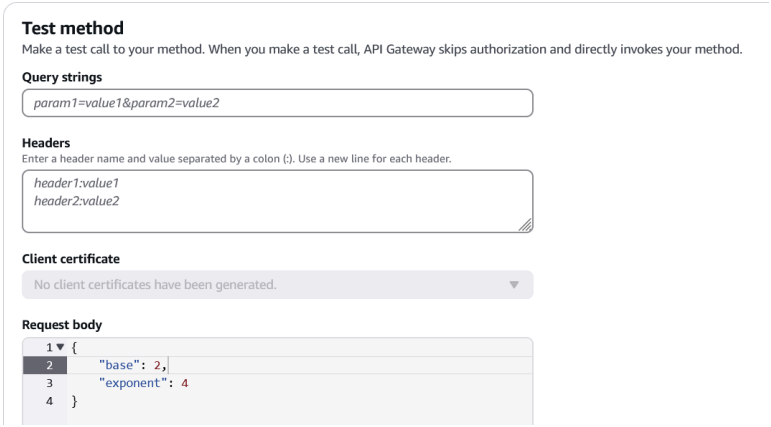


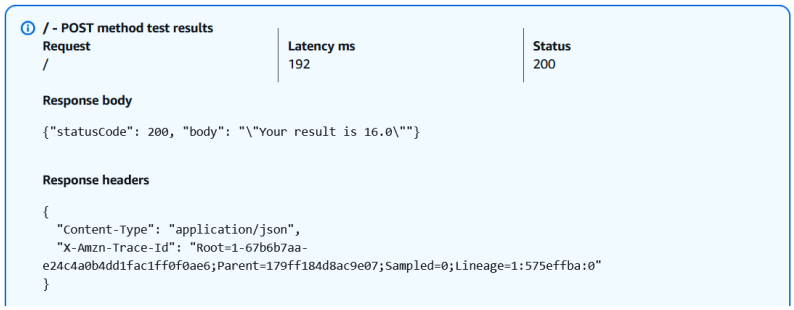
Invoke the Lambda Function

A public endpoint or URL is needed for the users to invoke and trigger the lambda function. API Gateway is used to build API's, which can be used to invoke Lambda functions. I created a "PowerofMath" API and enabled CORS (Cross Origin Resource Sharing) which allows for different domains to access resources from another origin or domain. In this case Amplify which has its own domain will access resources from Lambda, which has its own domain.



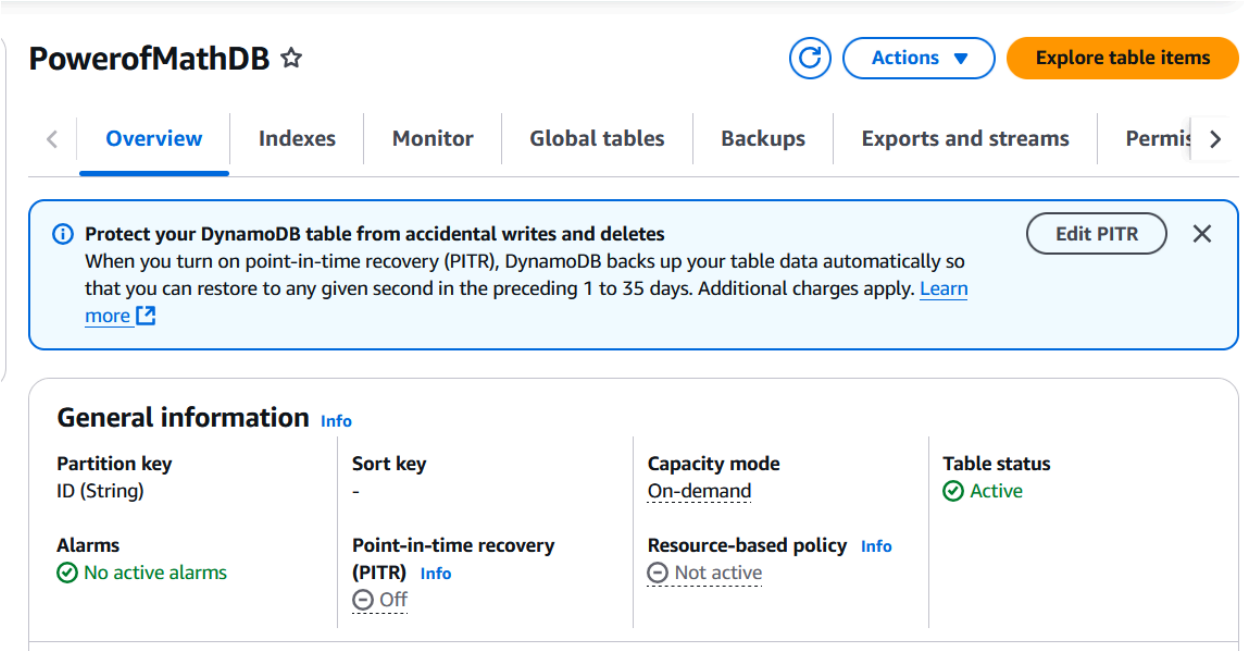
Testing the API and Validating its results:





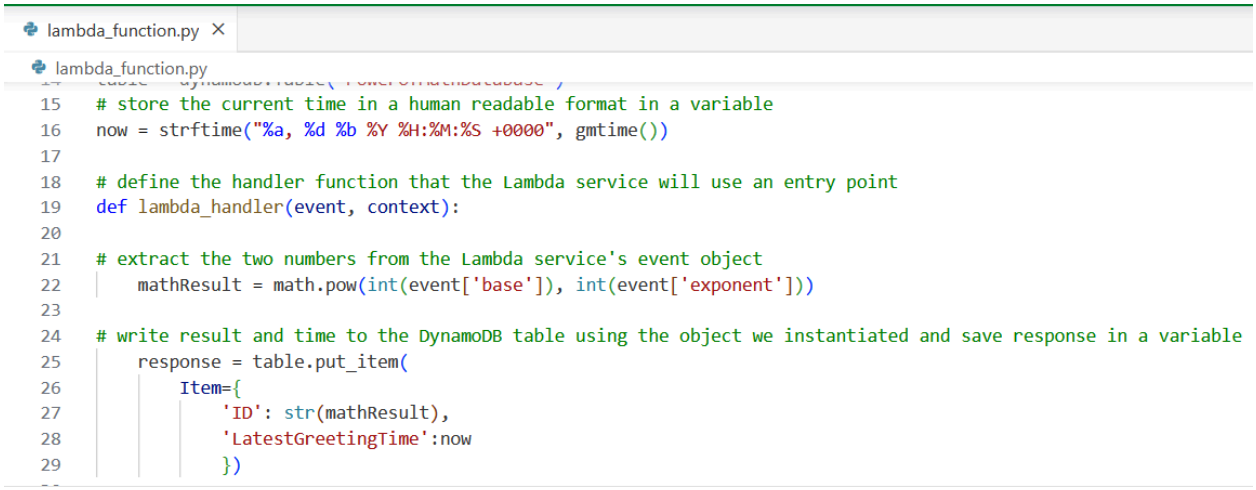
Store and Return the Results

DynamoDB, a lightweight, Key-Value, NoSQL database was used for to store and return results.



Back at the Lambda Function, an inline policy was created to allow it to write to the DynamoDB as seen below.

The Code was then edited again to incorporate access to the Database into the code itself, and also some styling choices were also added.



Final Result: