TO CREATE AN END-TO-END APPLICATION

Name: Y. Meghana

Roll No.: 20A91A04C4

Branch: ECE

AWS-DEVOPS (Afternoon)

Aditya Engineering College (AEC).

We are going to create an end-to-end application using various tool in Amazon Web Services (AWS). Here end-to-end describes a process that takes a system or service from beginning to end and delivers a complete functional solution, usually without needing to obtain anything from a third party.

So, in order to create this application, we are going to use five services in AWS:

* AWS Amplify
* AWS Lambda
* API Gateway
* DynamoDB

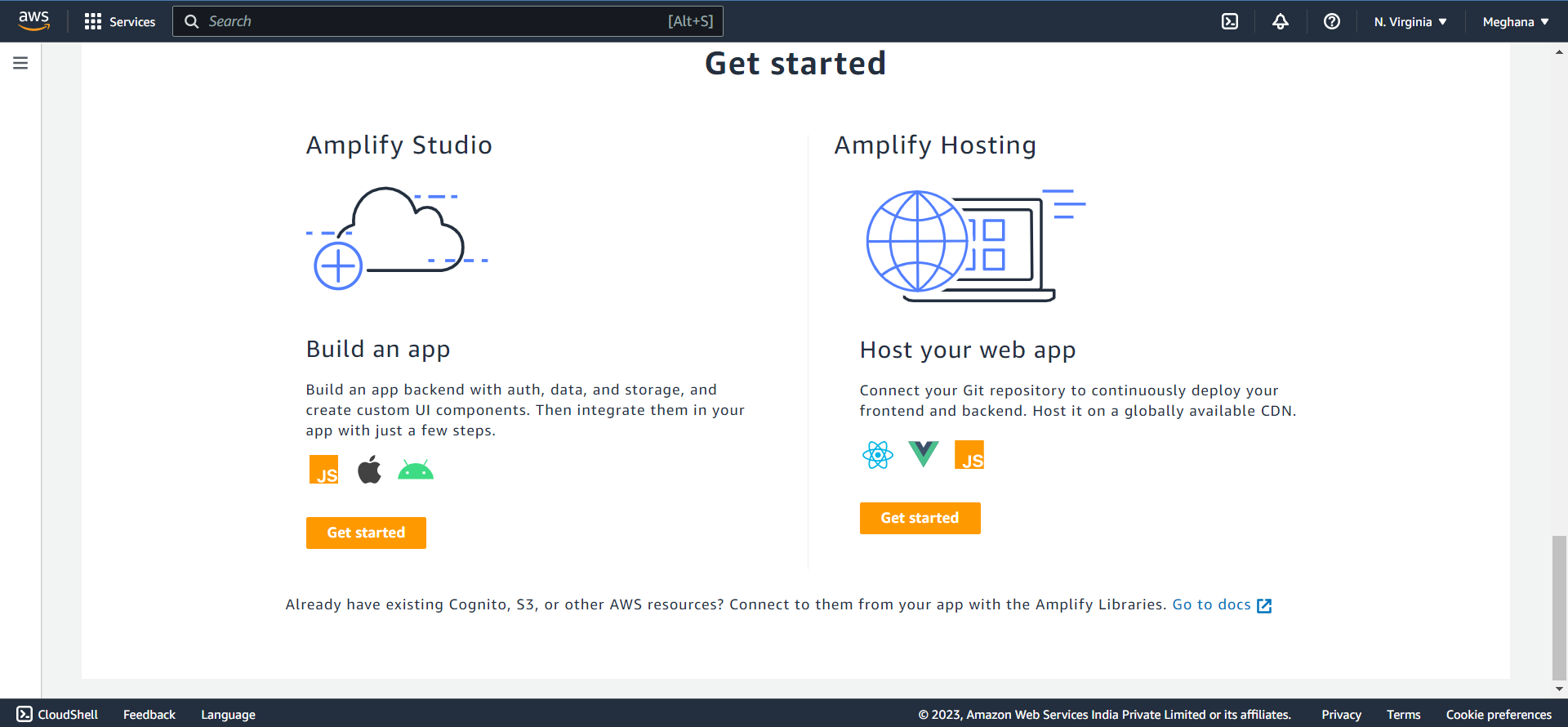
Resulting in fully-functional math web application

Starting we need to

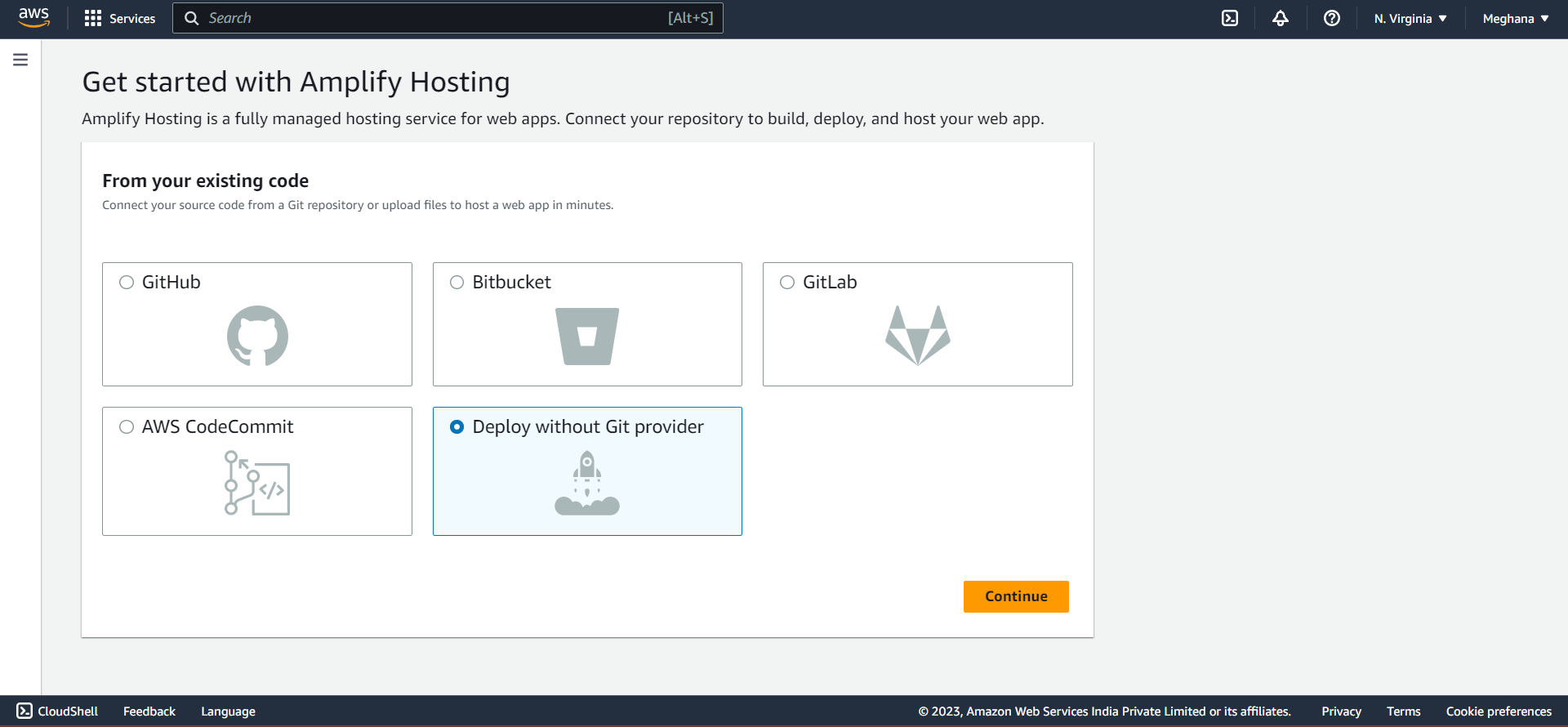
* Create/host a webpage
* Invoke Math functionality
* Do some math
* Store the result
* Handle permissions

Step-1: we are going to build and host a website using Amplify:

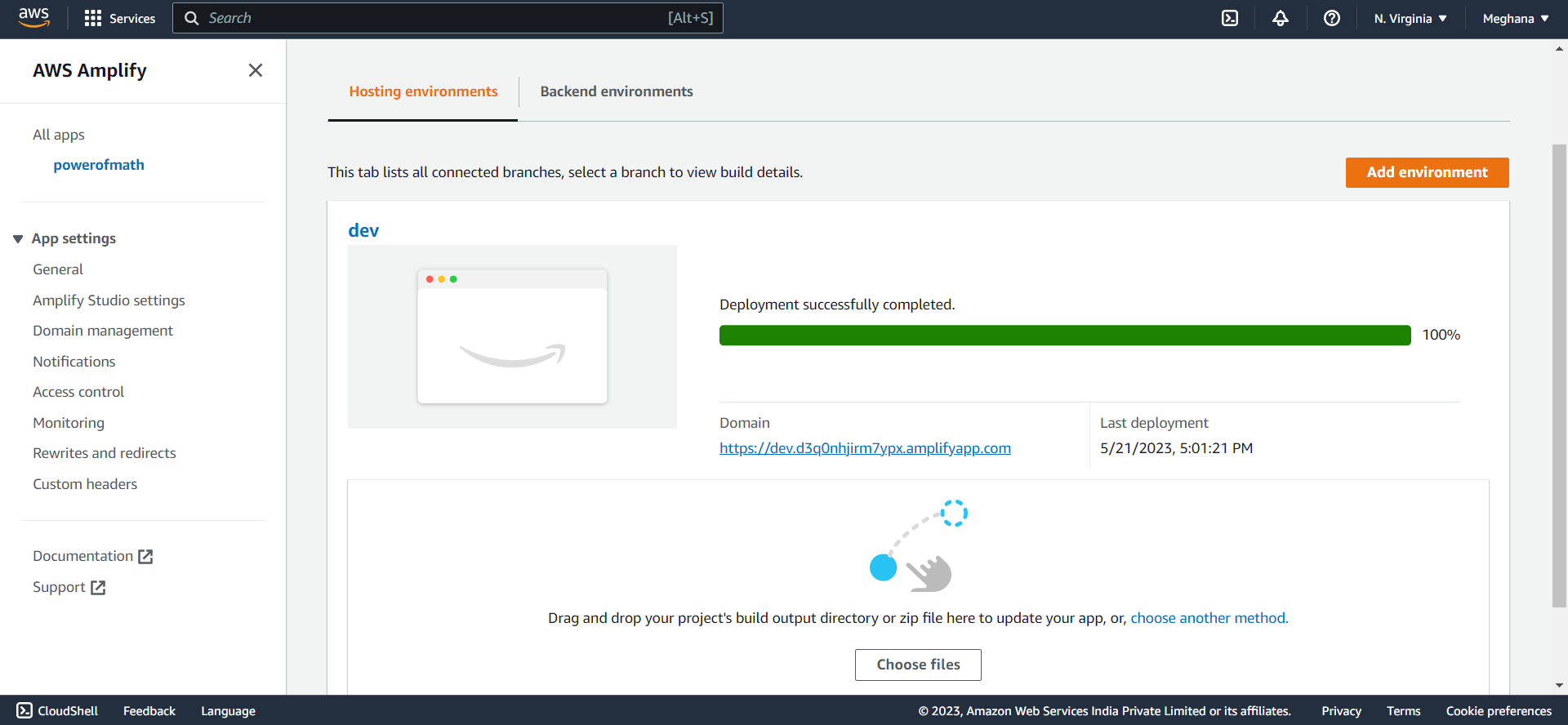
**AWS Amplify** is a set of product and tools that enable mobile and front-end web developers to build and deploy secure, scalable full-Stack application. Pick Amplify Hosting



Trying without Git provider:

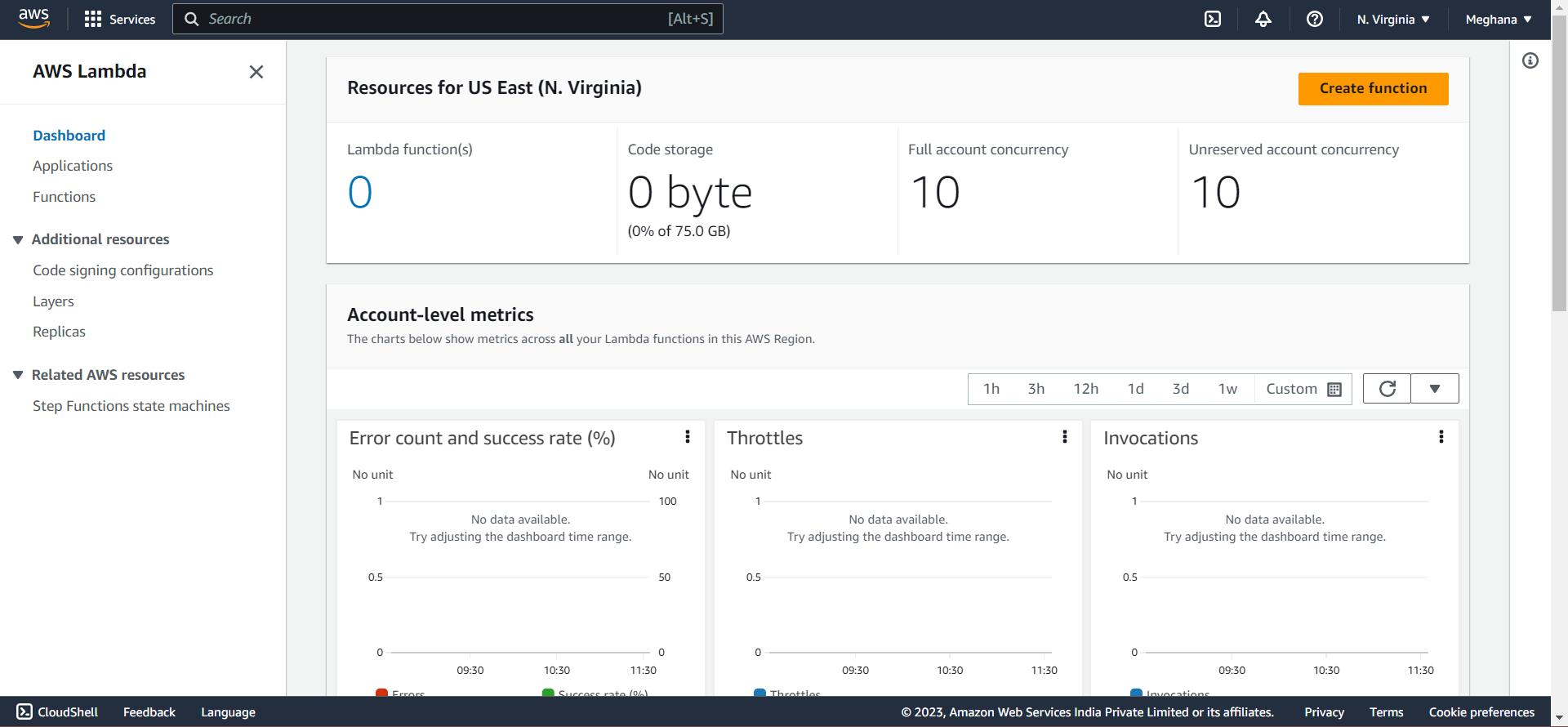


A Sample website is created a website:

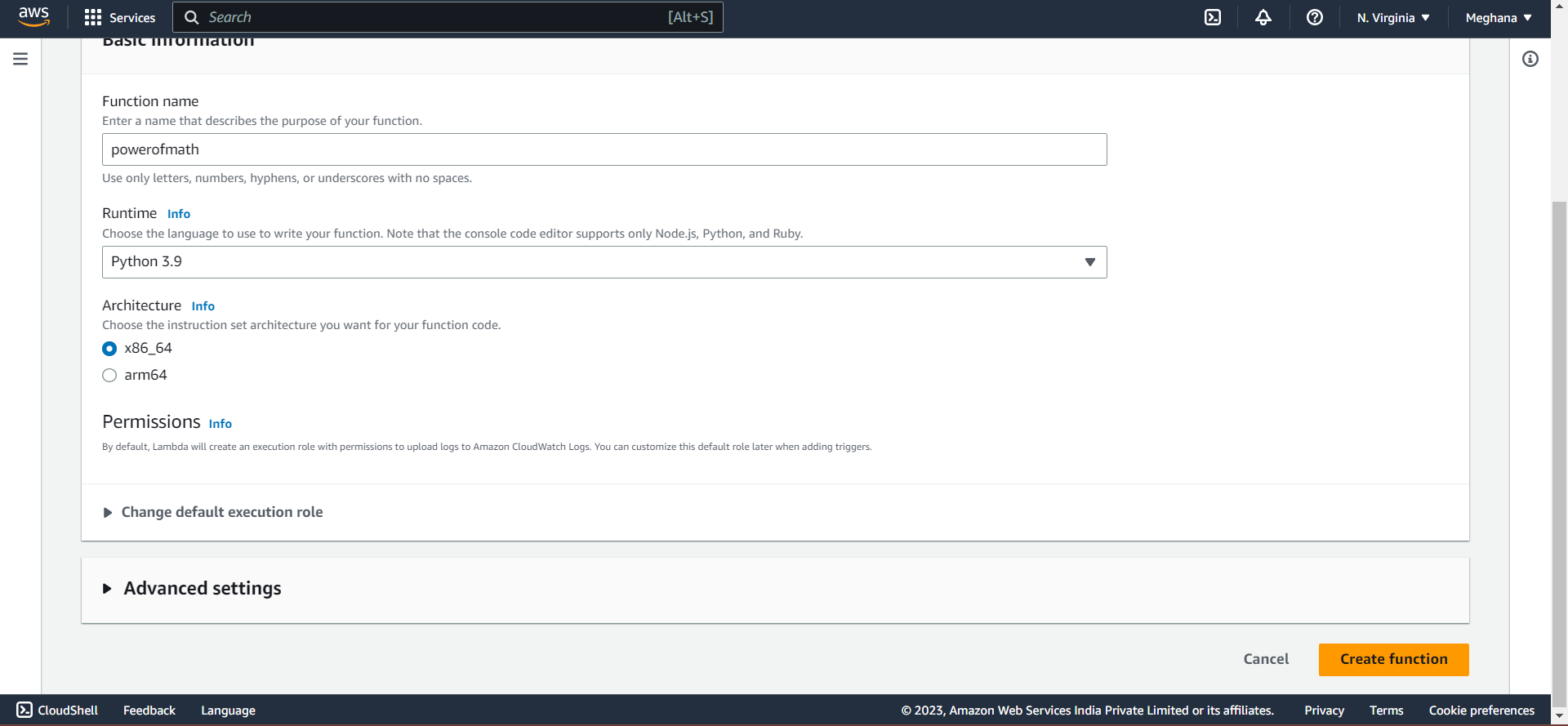


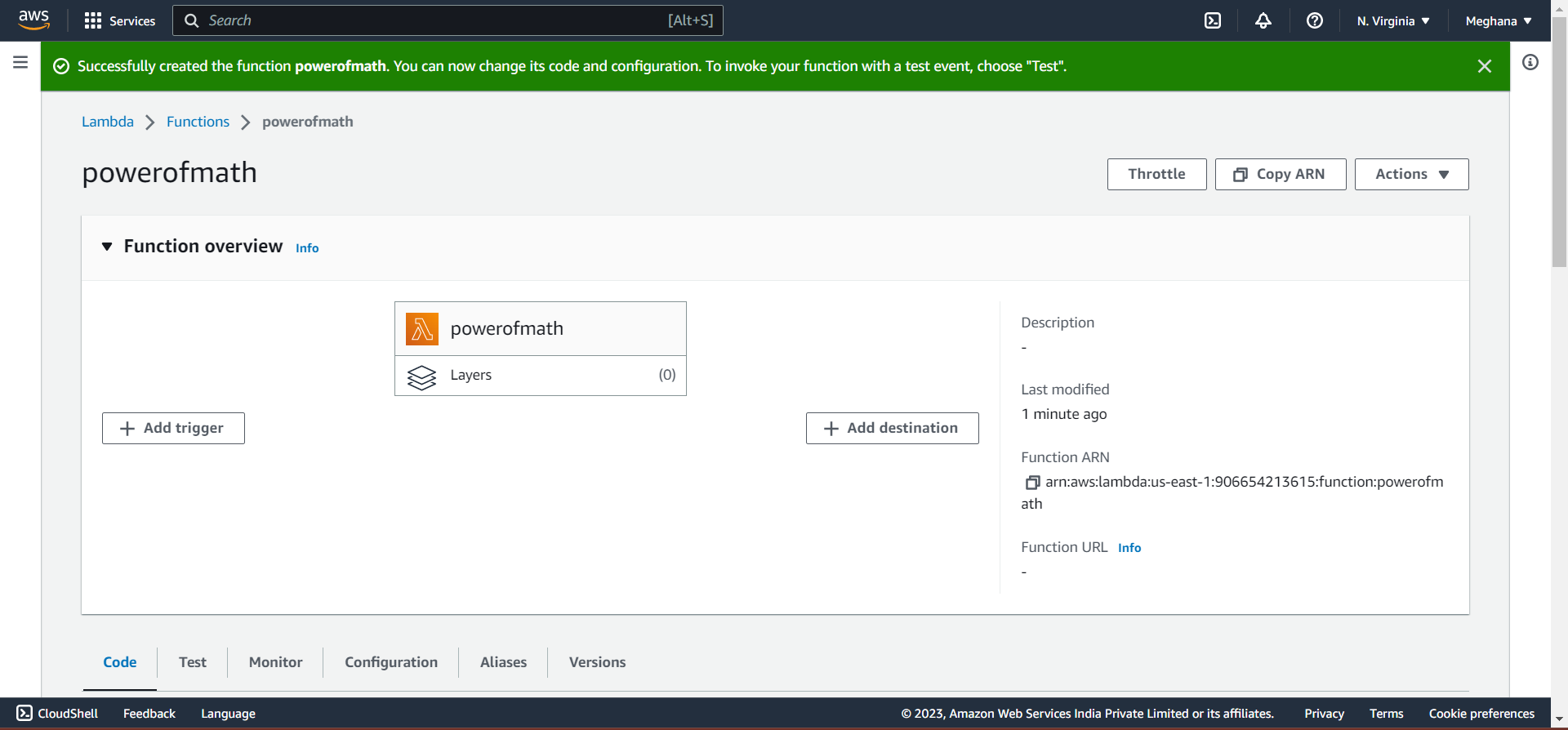
Step 2: So now we are building some math information Using AWS Lambda:

**AWS Lambda** is a compute service that lets you run code without provisioning or managing servers. The Lambda service runs your function only when needed and scales automatically.

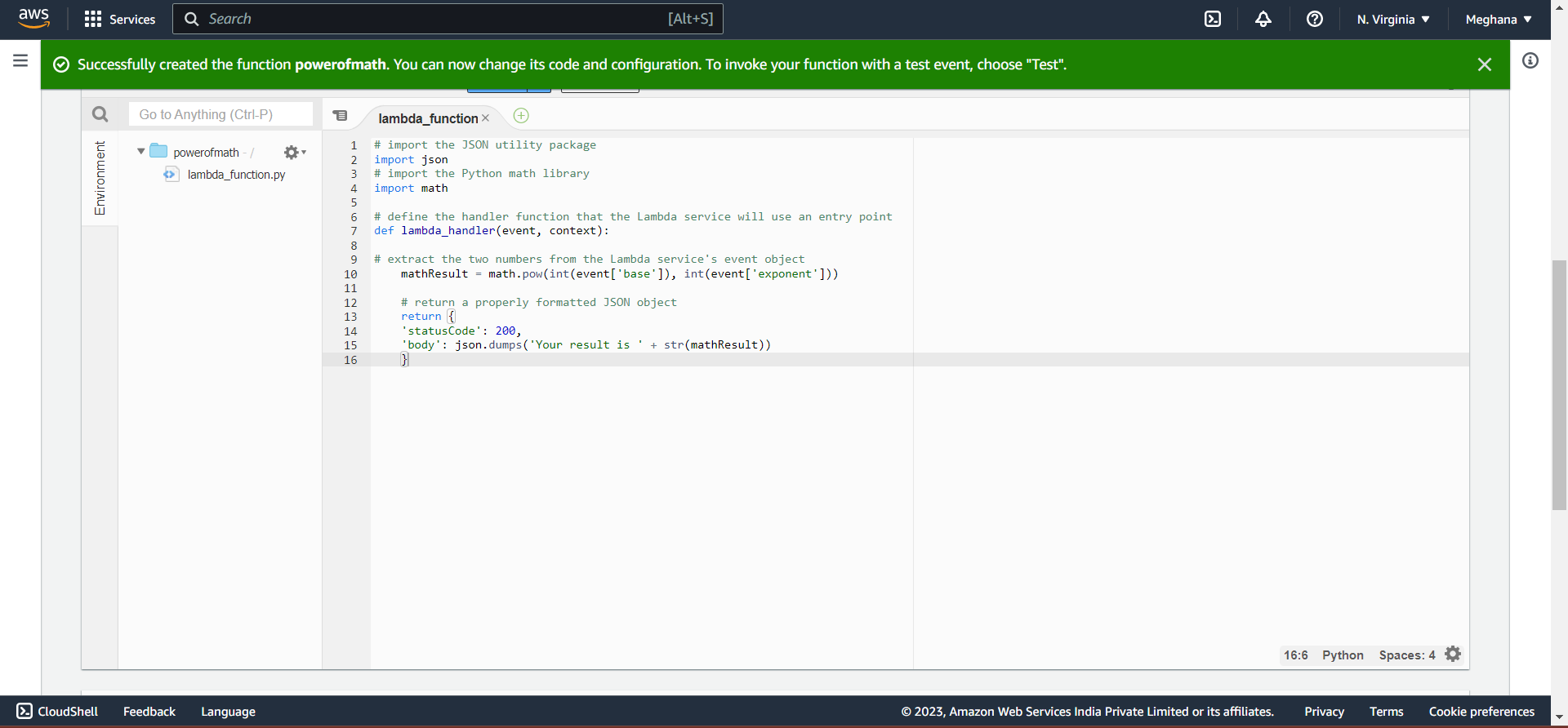


Create a function:





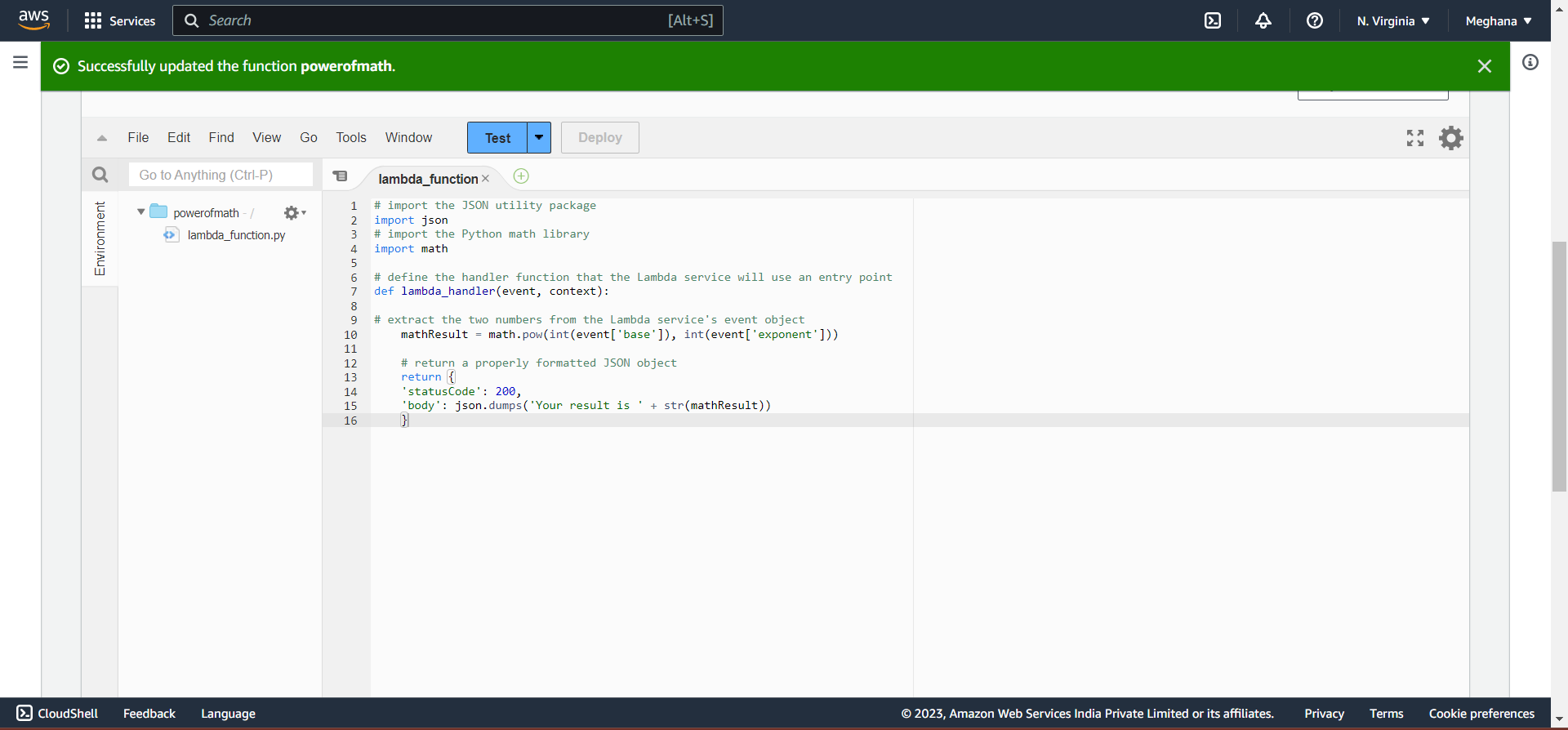
After Creating the function, we are applying a Python code without any server as lambda runs serverless.



In the code,

We have to declare a math function in order to get the power of a number

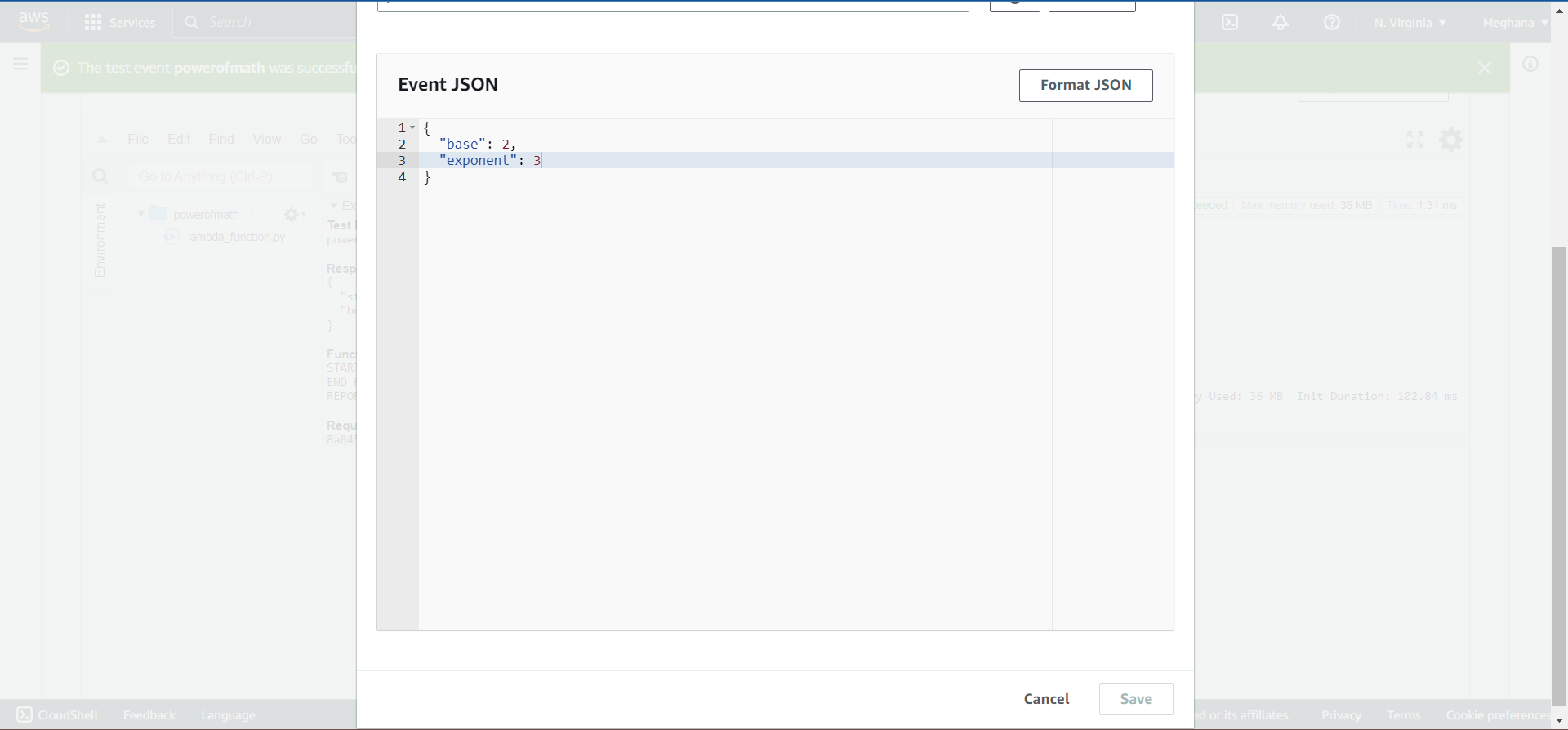
i.e., A Base and Exponent is taken as input and result is returned. Then Deploy



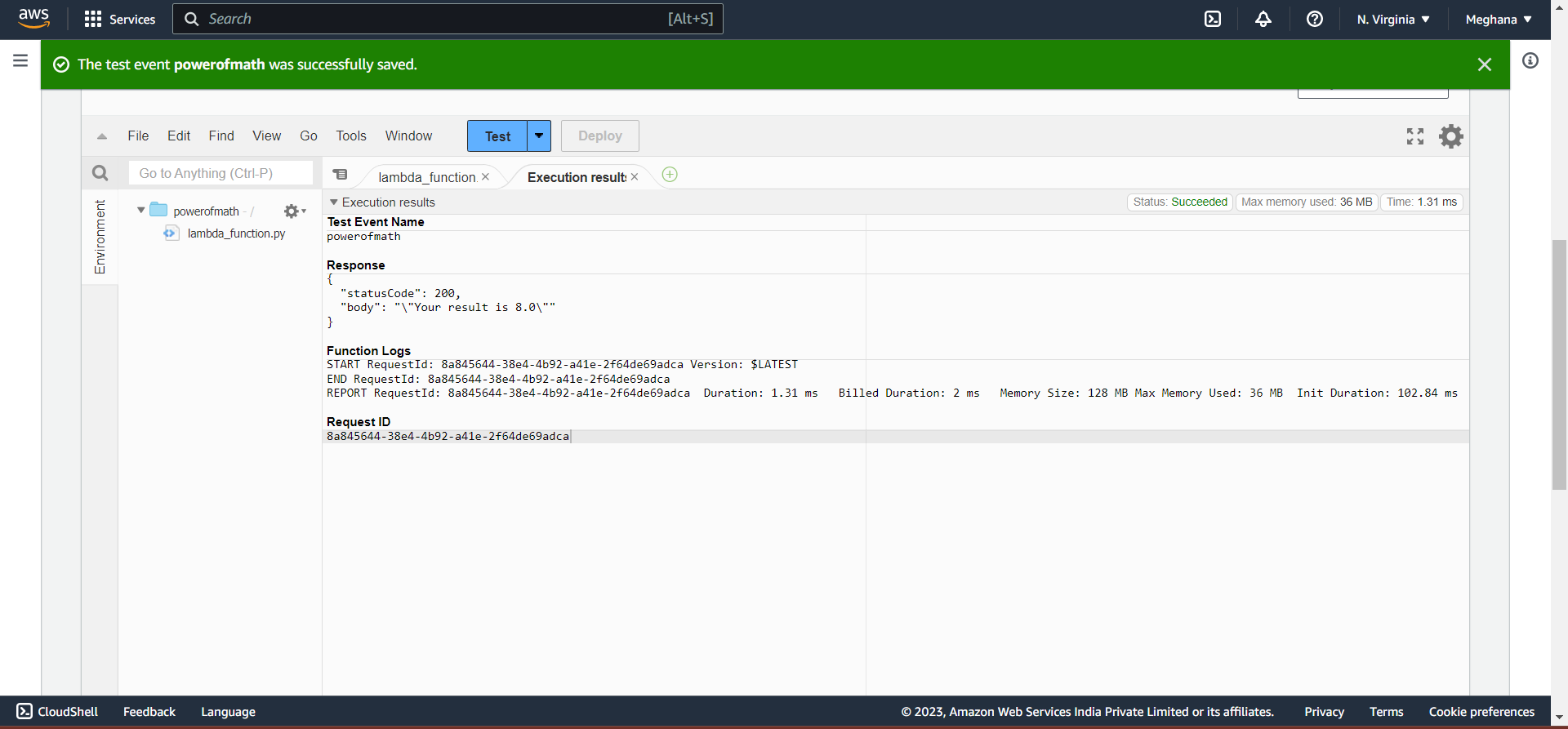
Successfully updated. Now, test the function:

Create a Test event to check the python code,

And test to execute the code



Event 2\*\*3:Successfully Executed



Step 3: Triggering Math Function using API Gateway

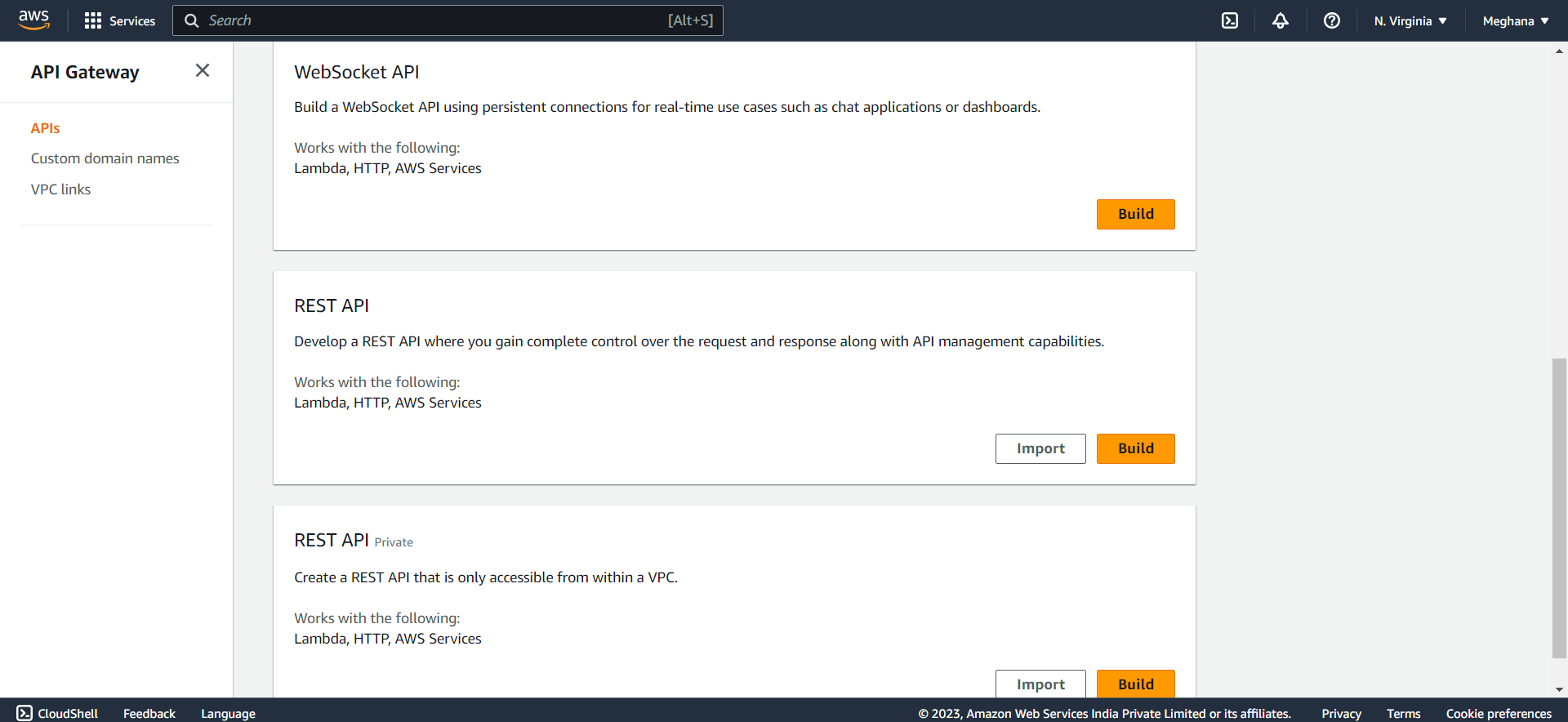
**Amazon API Gateway** is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. Build RESTful APIs optimized for serverless workloads and HTTP backends using HTTP APIs.

**An API gateway** is an API management tool that sits between a client and a collection of backend services.

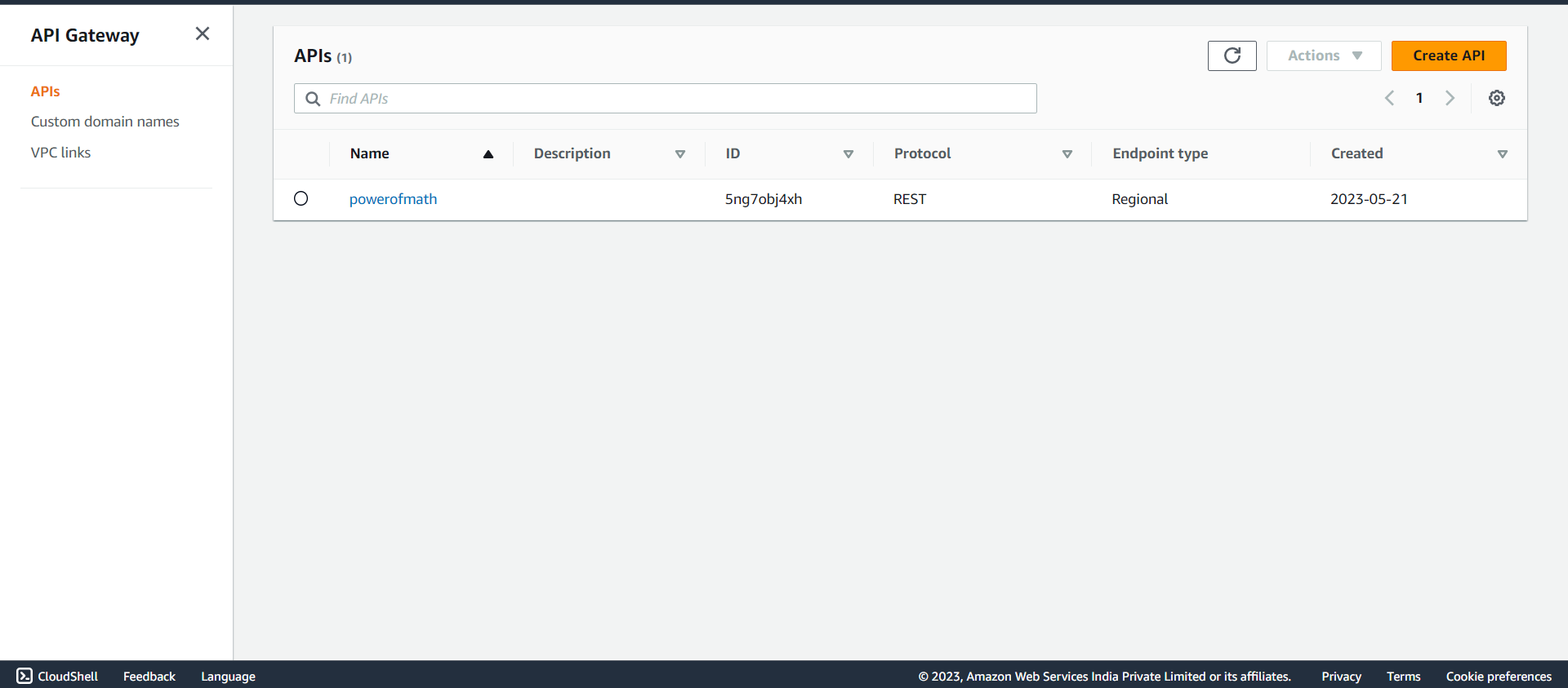
**API-** An application programming interface is a way for two or more computer programs to communicate with each other.

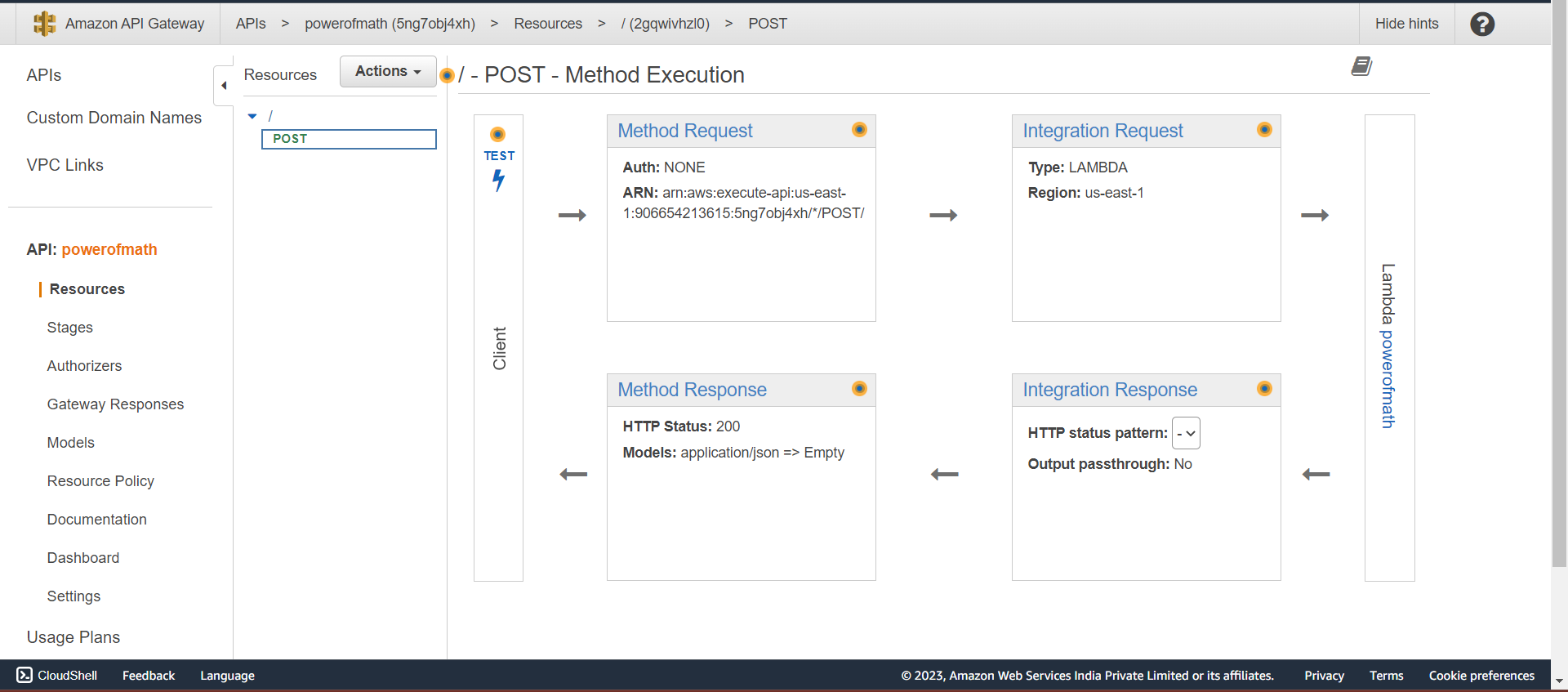
So here we are going to create a connection between the lambda and the Amplify to create website and execute the code in lambda.

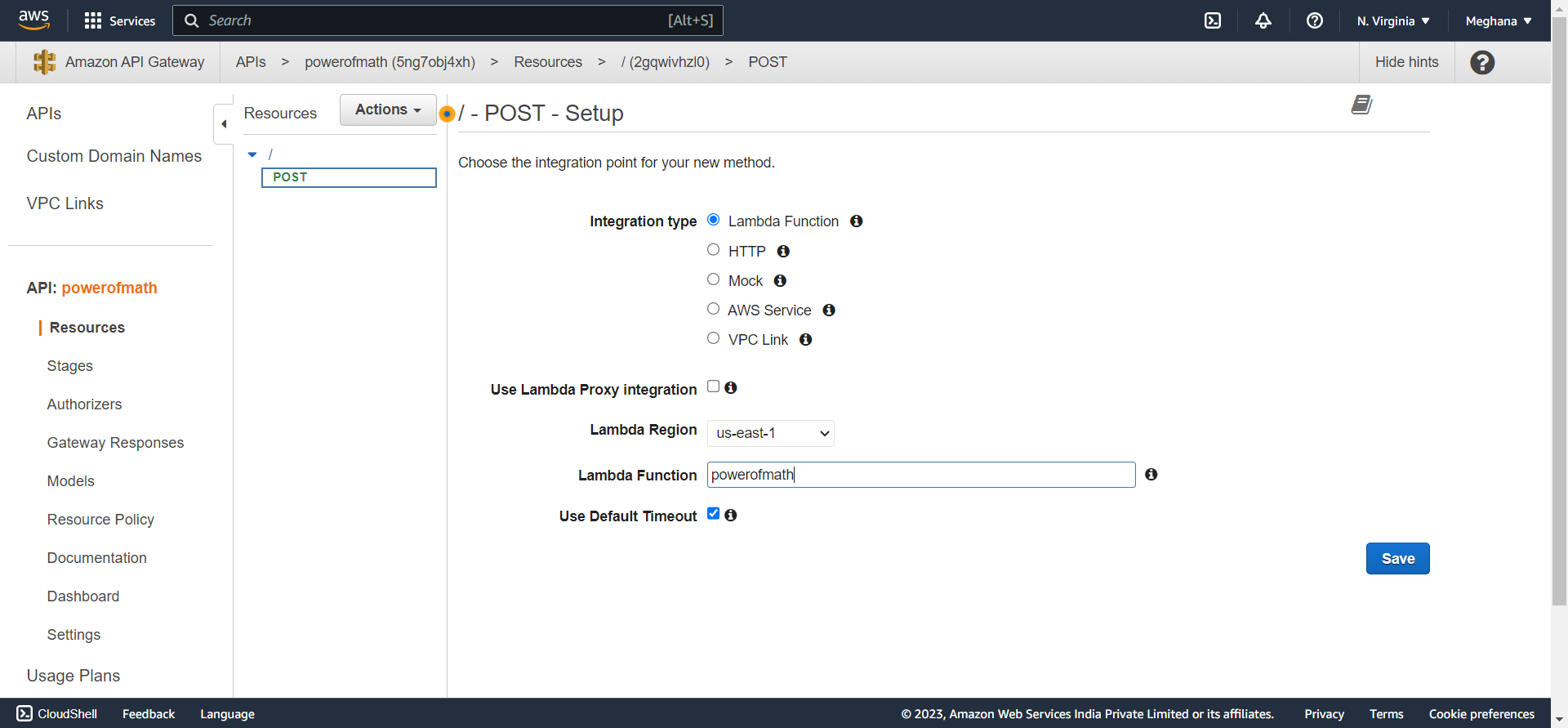
Building a RESTAPI to get control over lambda function we created before:



Select new API give a name and create a API:

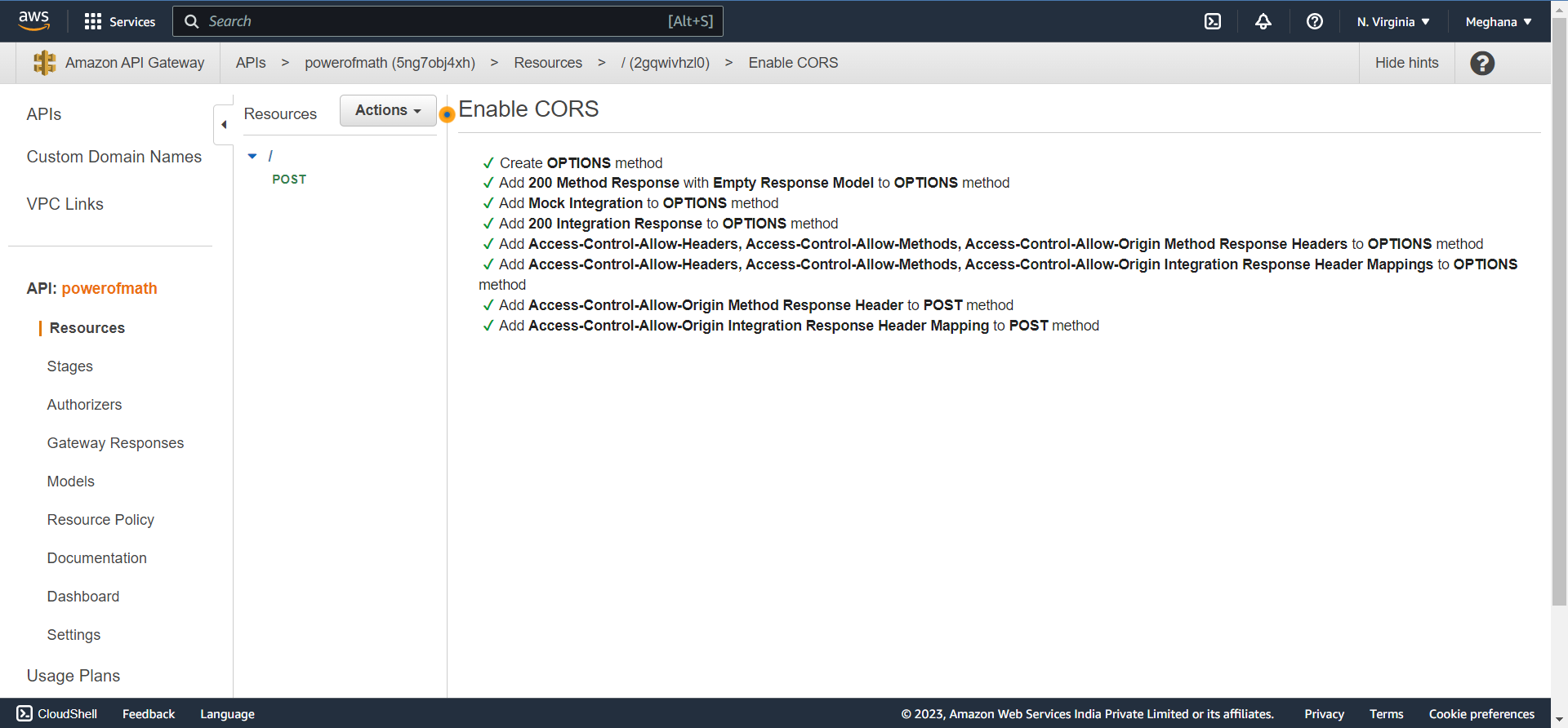


Now create a method in order to trigger lambda function with a post and save: 

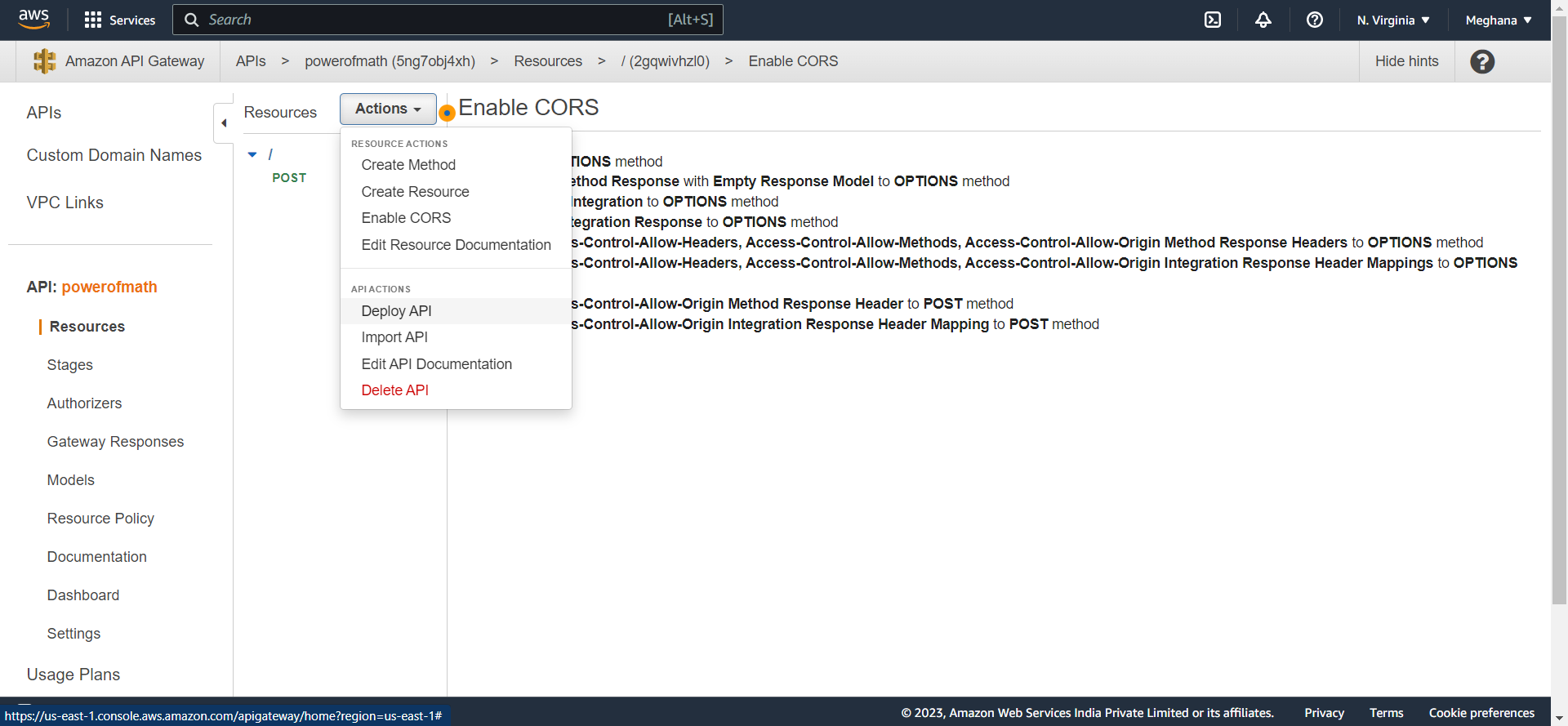


Enable CORS (Cross-Origin Resource Sharing) in order to transmit HTTP headers to determine browsers from accessing.

It helps the website to run in different domains like for a mobile one domain and a laptop another domain.



Setup a new deployment stage:



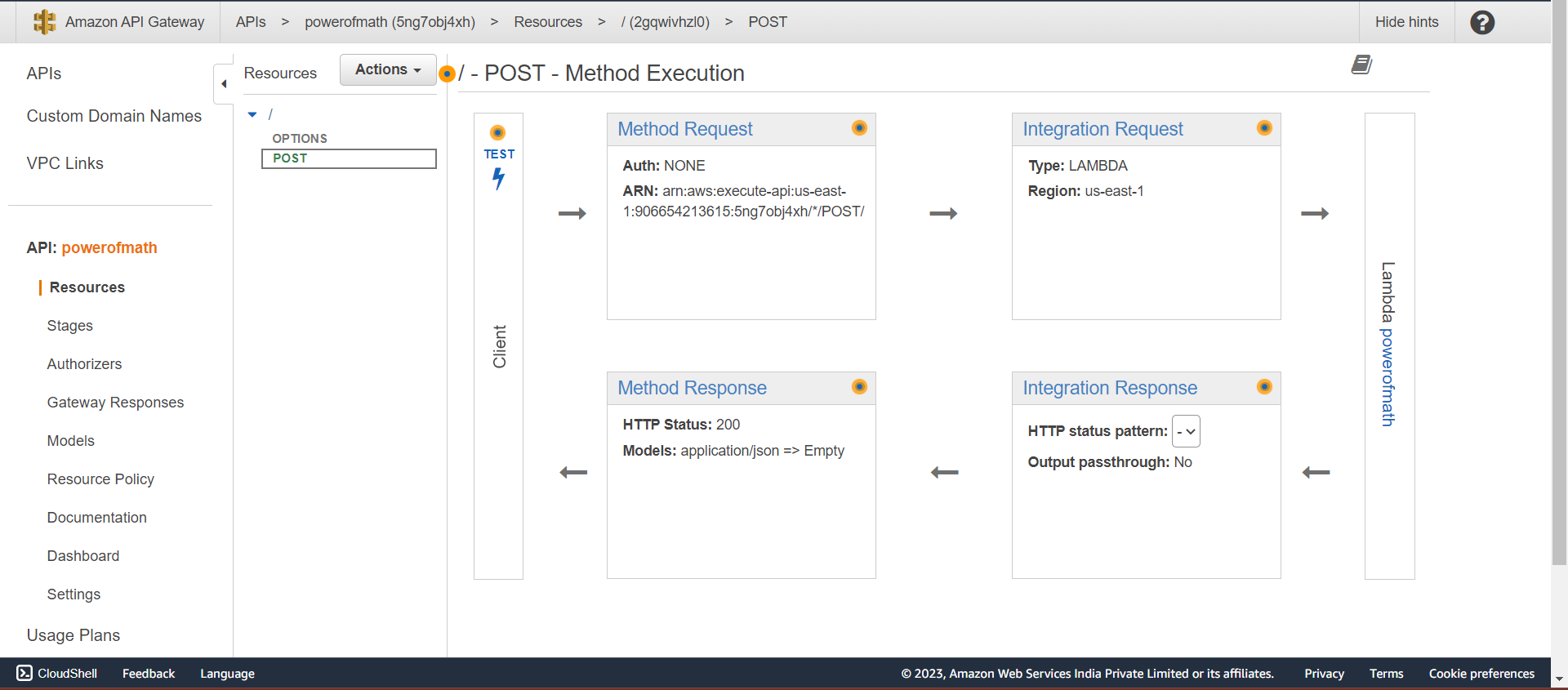


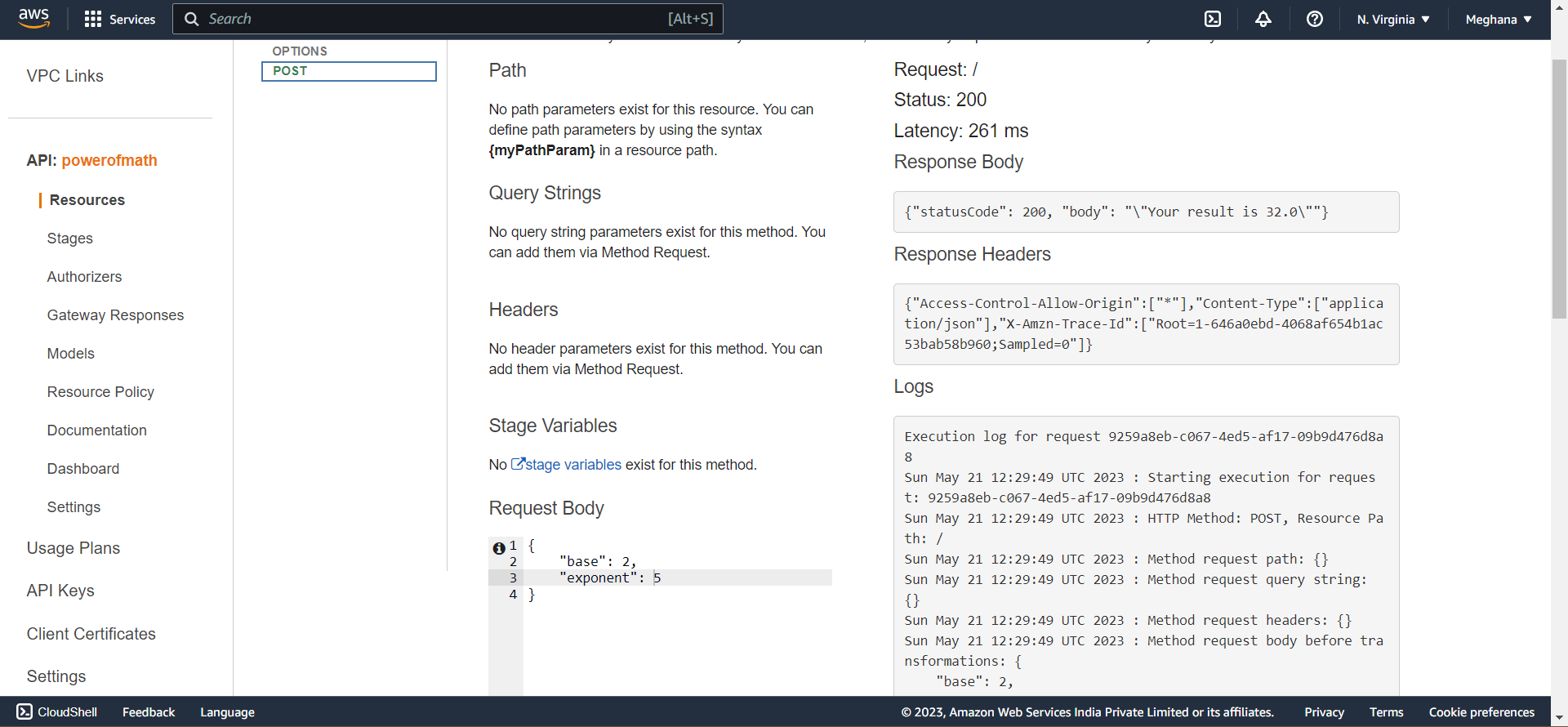
Note the API Gateway URL in a notepad for further purposes:

Invoke URL to Notepad



Let us now test the path if its connected or not by moving to resources and test:





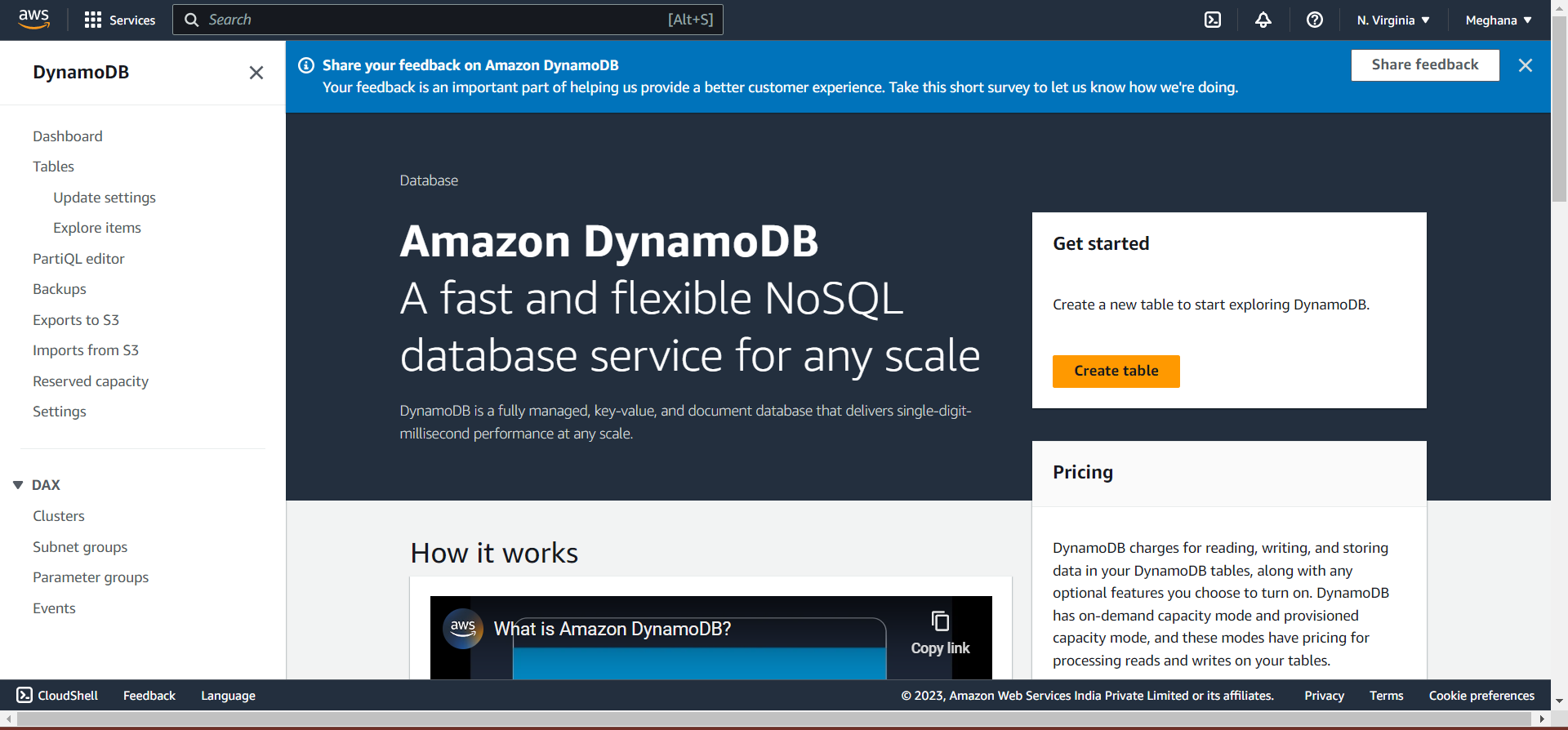
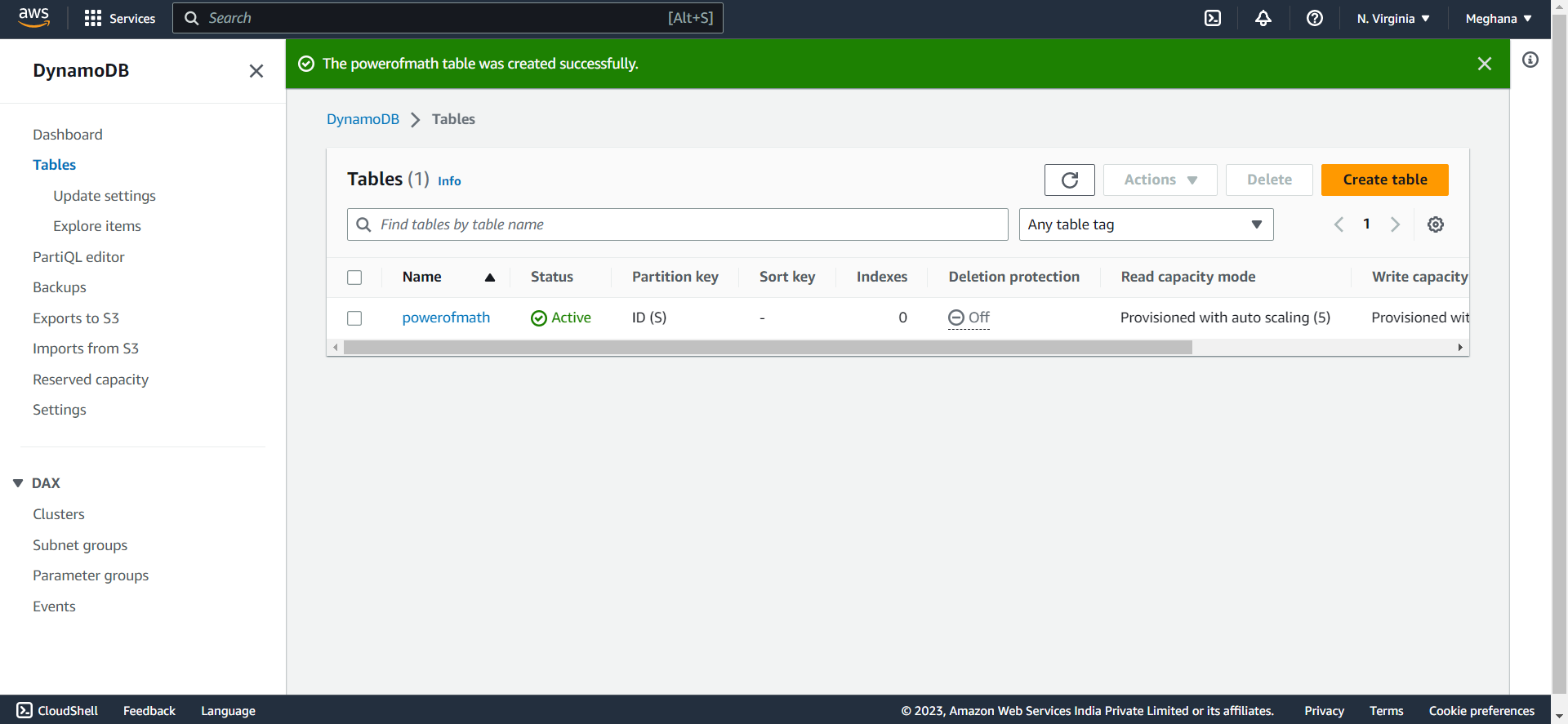
Step 4: Creating a Database to sore the result and time in order to list out the operations performed by users.

To deal with database we are going to use DynamoDB which is a NO-SQL database service and we also need to set permissions using IAM in order to specify lambda working (like permission to write).

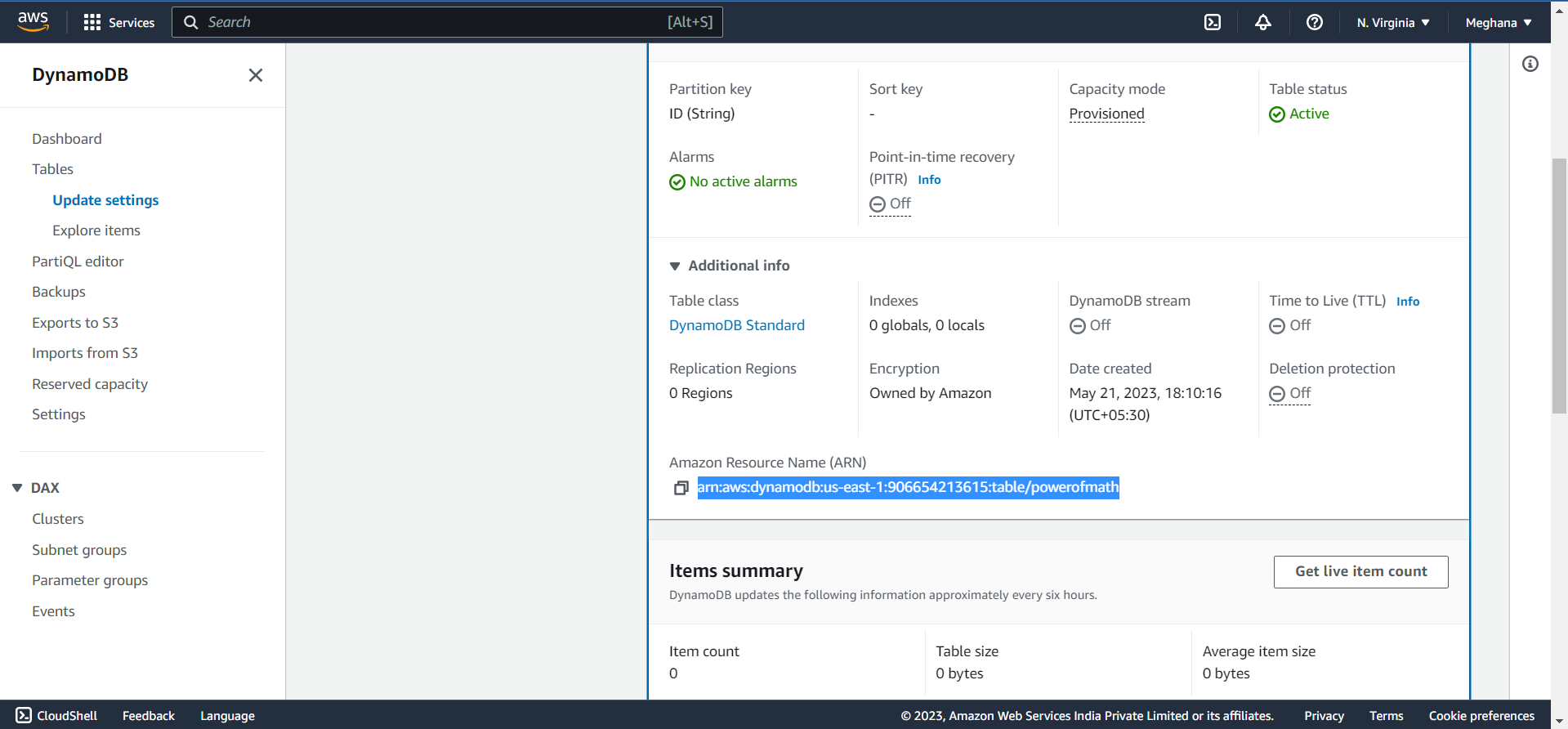
**Amazon DynamoDB** is a fully managed proprietary NoSQL database service that supports key–value and document data structures.

**NoSQL** is an approach to database management that can accommodate a wide variety of data models, including key-value, document, columnar and graph formats.

Creating a new table in DynamoDB:

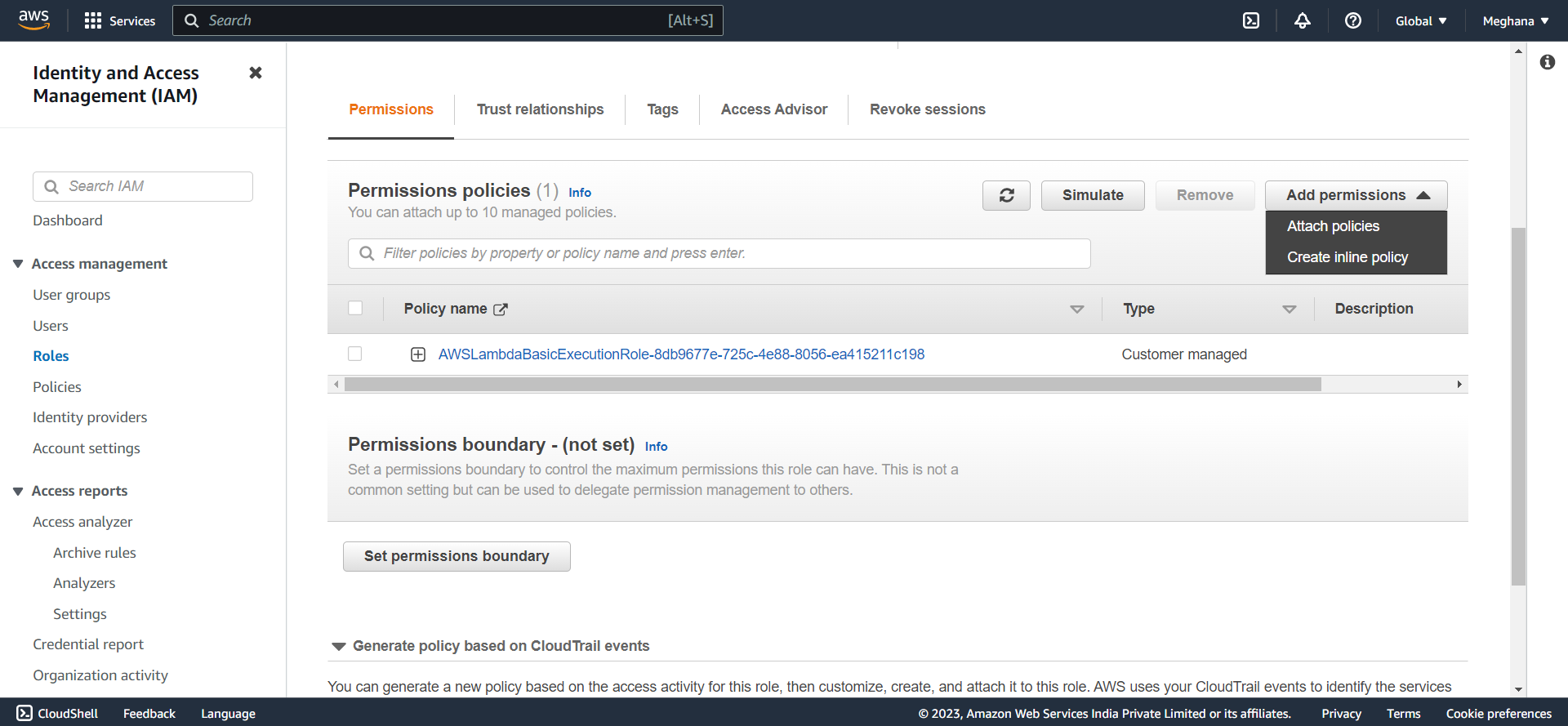
 

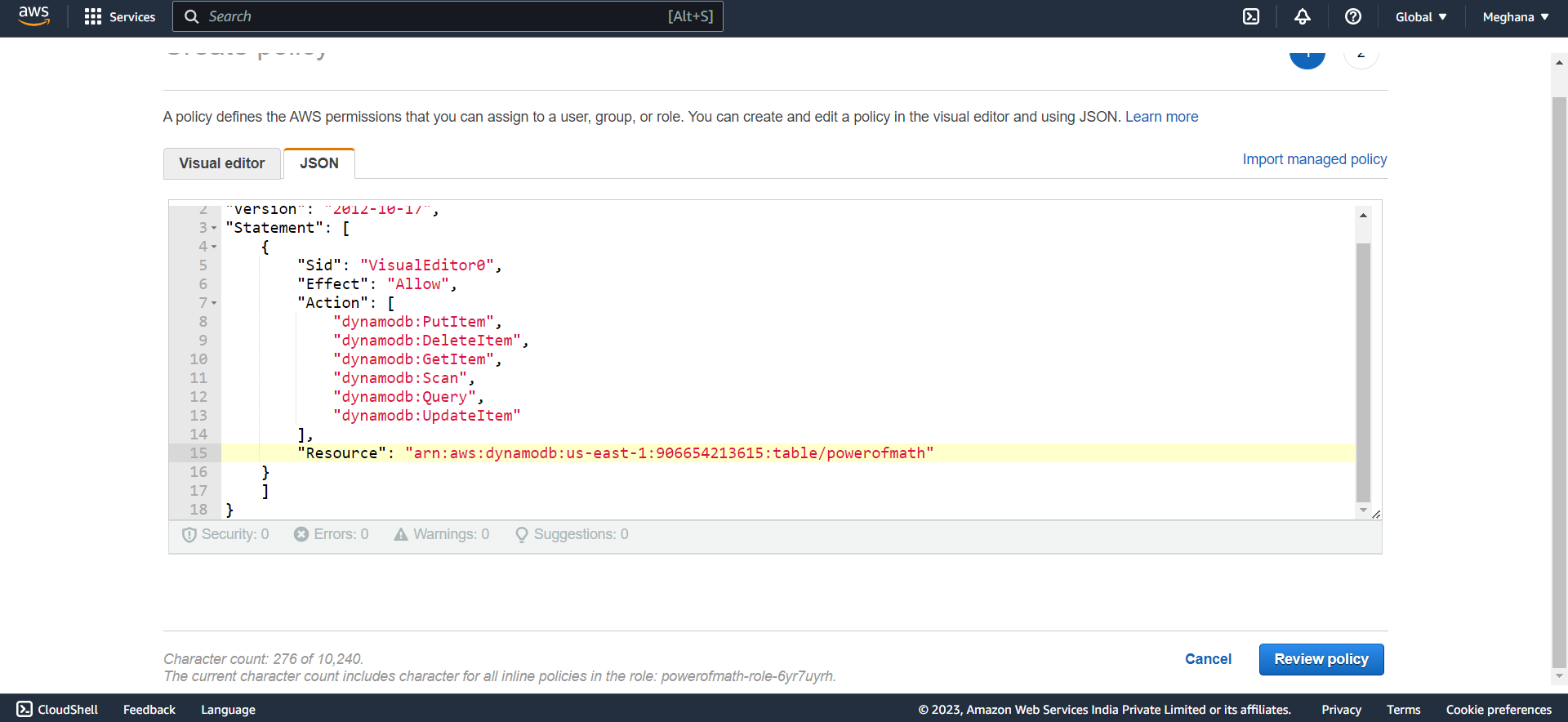
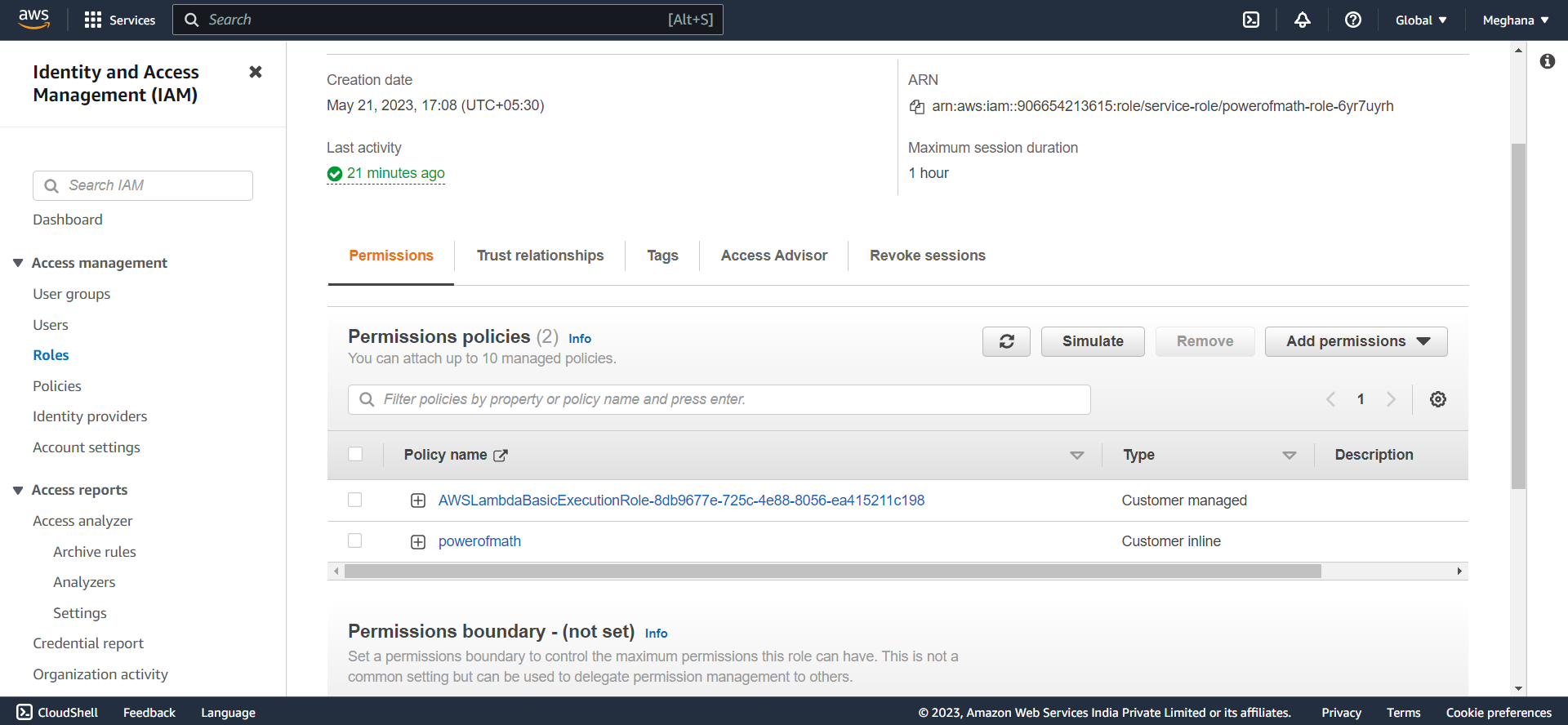
Now store arn of the database table in notepad for further usage:



Now set permissions in Lambda:

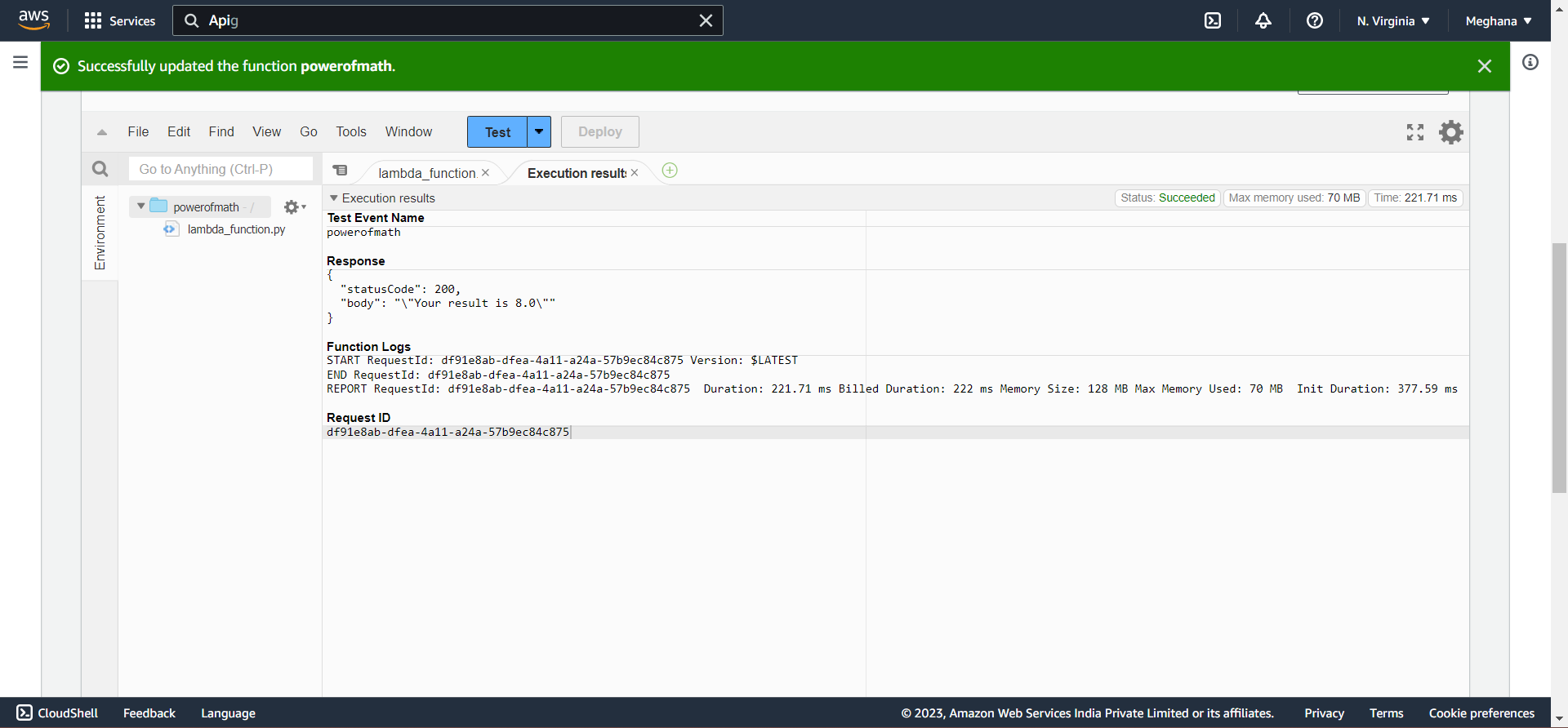
Configuration->Permissions->link->directed to IAM console->ADD Permissions->inline policies->JSON tab-> write the permissions to be given and give the ARN in code to give allowance



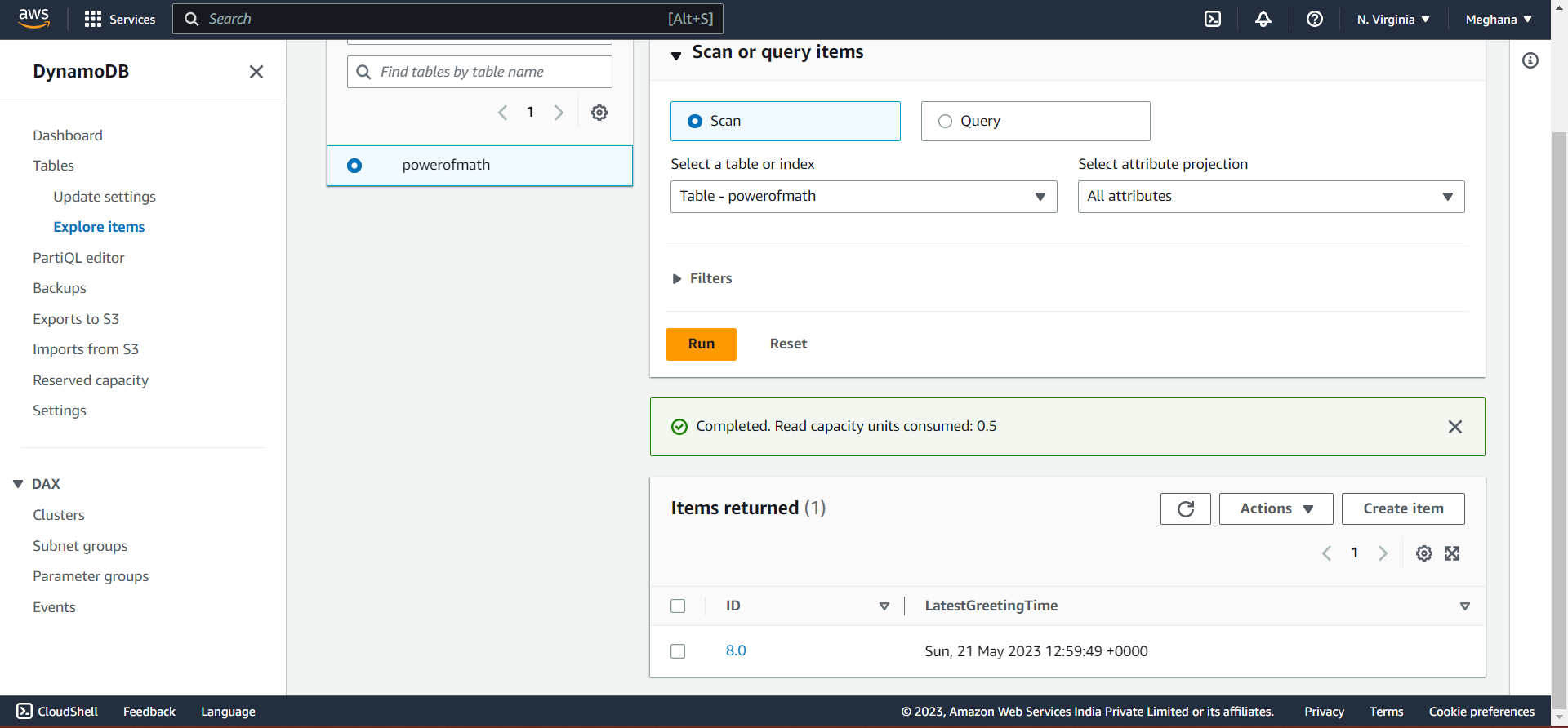
 

Now write the python code in lambda function with database description and funtion to connect those and deploy it:





Results in DynamoDB:



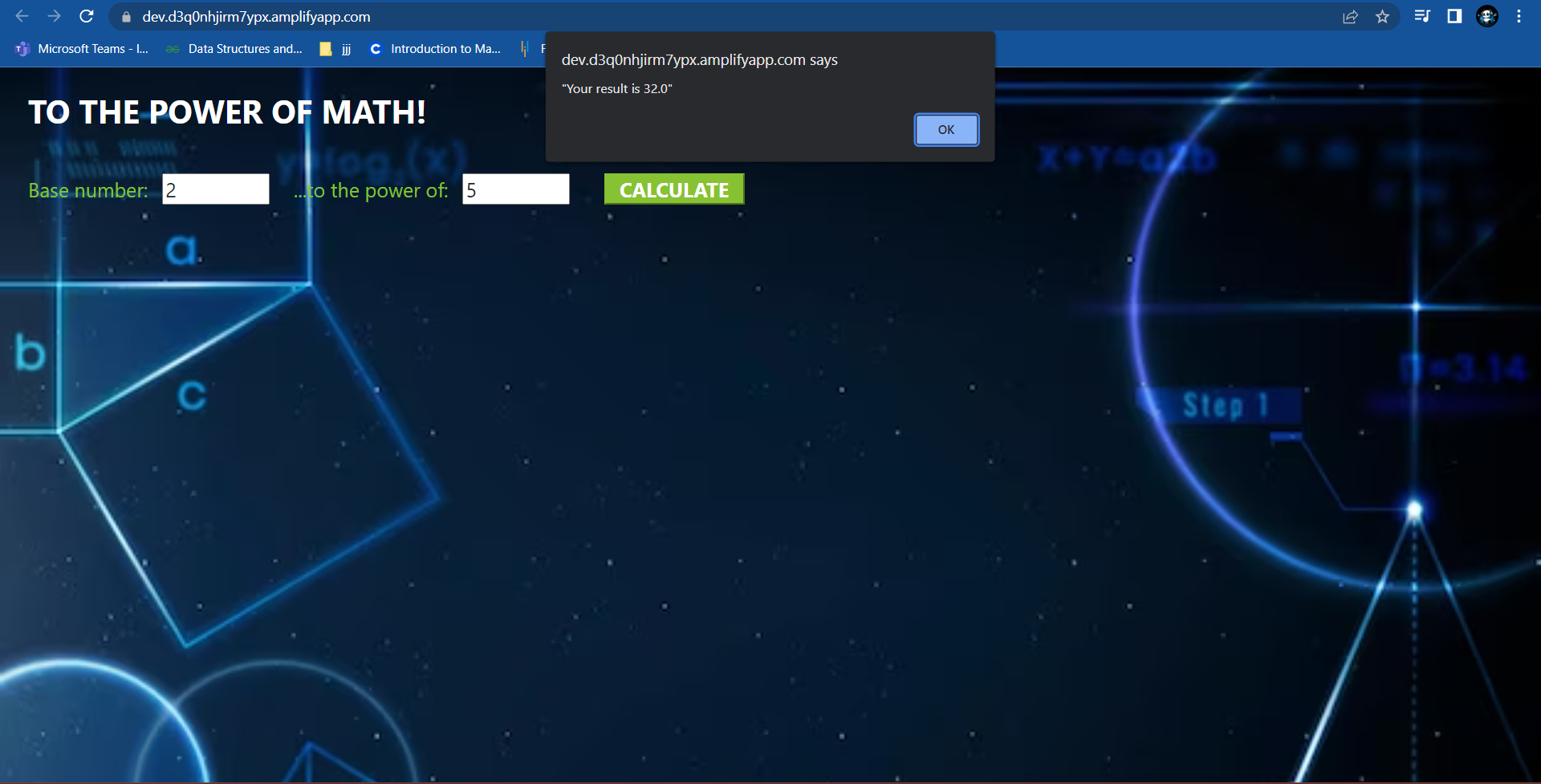
The Connection between the database and lambda are proved.

Step 5: Now we have to create a connection between the whole system and amplify to get the results from the website:

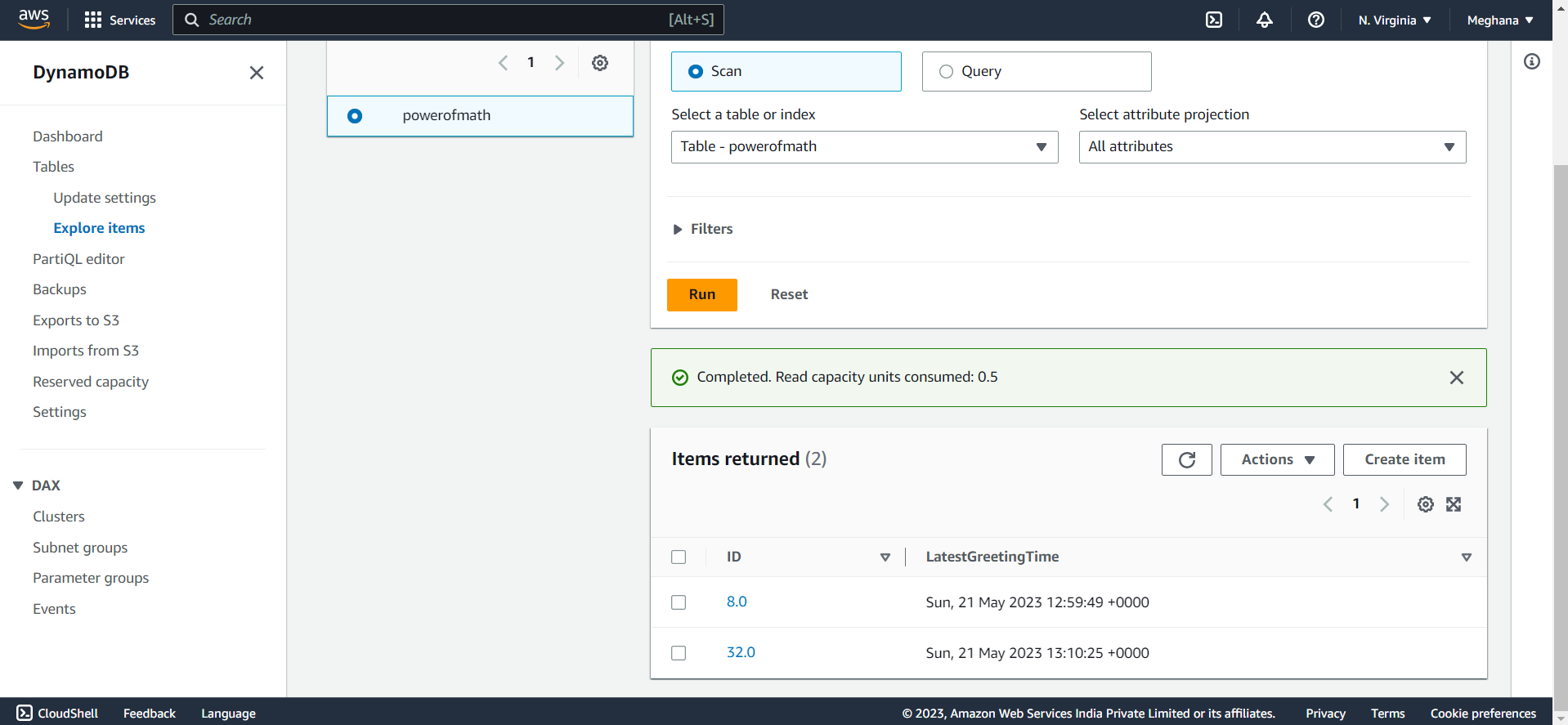
Now in the index.html file we need to add the required API gateway link to link the whole process:

and redeploying it in the AWS Amplify:

This is the page and the post gave the answer (2\*\*5)=32



And hence in DynamoDB we stored the answer and the time:



From this project,

As I am a programmer who usually chooses python to be my first language and have little knowledge on HTML, CSS to create a website, I combined all these to work on AWS in order to create a end-to-end application.