

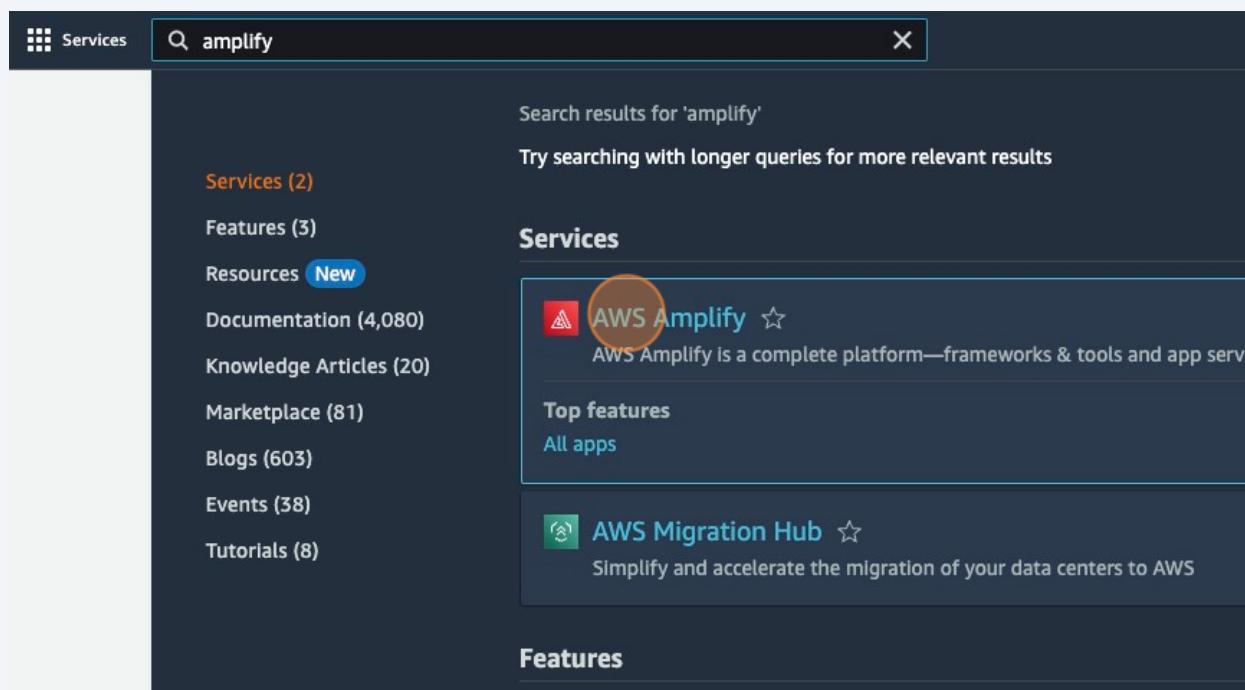
Creating a Web Application with AWS Amplify, API Gateway, Lambda, and DynamoDB

Scribe 

This guide provides step-by-step instructions on how to create a web application using AWS Amplify, API Gateway, Lambda, and DynamoDB. It covers setting up the environment, creating functions, configuring APIs, and managing databases. By following this guide, users can learn how to build and deploy a web application using these AWS services.

1 Note before navigating to the console an index.html file will need to be created.

2 In the search field of your AWS console type amplify and click on AWS Amplify.



3

The screenshot shows the AWS Amplify landing page. At the top is a circular logo with a stylized 'A' icon. Below it is the heading 'AWS Amplify' in large, bold, dark blue letters. Underneath the heading is the tagline 'Fastest, easiest way to develop mobile and web apps that scale.' A prominent orange button labeled 'GET STARTED' is centered below the tagline. Below the button are icons for React (blue atom), Vue (green 'V'), JavaScript (orange 'JS'), iOS (apple), Android (green phone), and AWS Lambda (blue 'L'). A descriptive paragraph follows: 'AWS Amplify is a set of products and tools that enable mobile and front-end web developers to build and deploy secure, scalable full-stack applications, powered by AWS.' At the bottom of the page is a dark footer bar with the AWS logo and the text 'AWS Amplify: Fastest, Easiest W...'.

4

Click "GET STARTED" and click Host your web app.

The screenshot shows the 'Host your web app' section of the AWS Amplify landing page. It features a graphic of a laptop displaying a globe. Below the graphic is the heading 'Host your web app'. A subtext explains: 'Connect your Git repository to continuously deploy your frontend and backend. Host it on a globally available CDN.' Below this are icons for React, Vue, and JavaScript. A large orange button labeled 'Get started' is prominently displayed. At the bottom of the page, there is a question: 'Want to connect to S3, CloudFront, or other AWS resources? Connect to them from your app with the Amplify Libraries.' followed by a link 'Go to docs'.

5 Click "Deploy without Git provider."

The screenshot shows a deployment configuration screen. At the top, there are three options: "Bitbucket" with a bucket icon, "GitLab" with a fox logo icon, and "Deploy without Git provider" with a rocket launching icon. Below these are two input fields: one containing "it" and another containing "/>". To the right of the input fields is a "Continue" button.

6 Click "Continue"

The screenshot shows a deployment configuration screen. It includes input fields for "Git provider" and "Deployment name", both of which are currently empty. A large blue box highlights the "Continue" button at the bottom right, which is also highlighted with an orange circle.

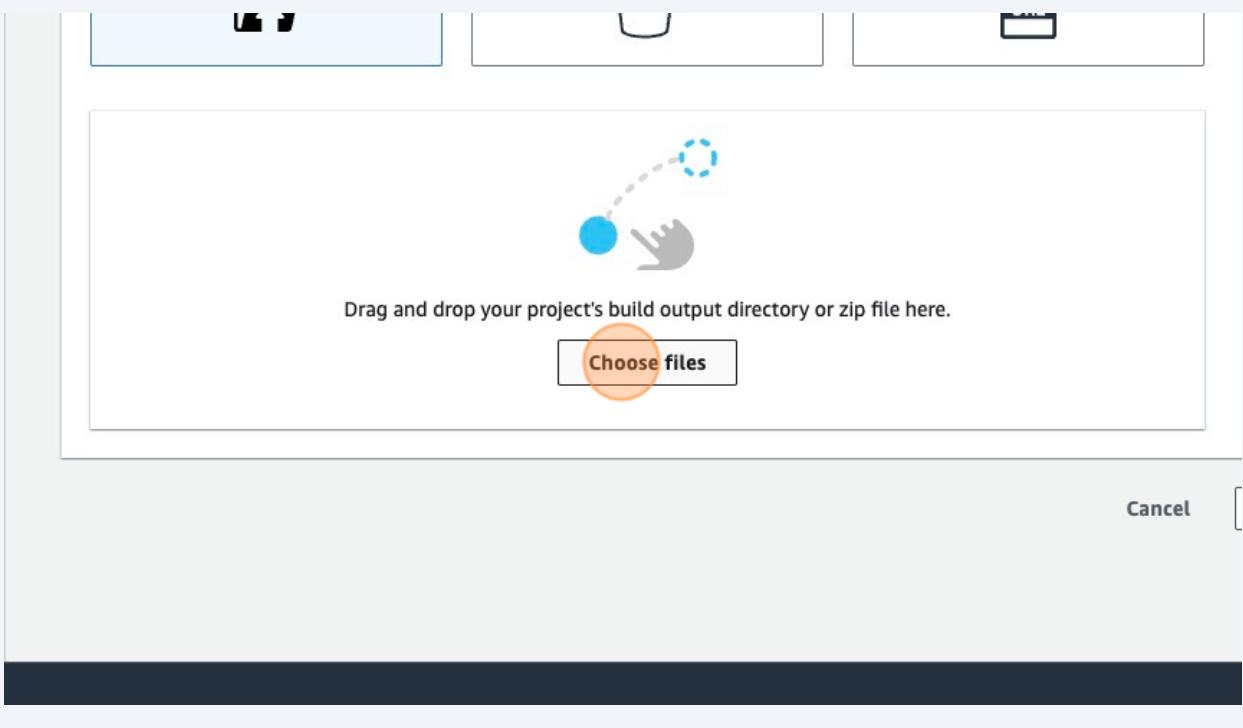
- 7 Click the "App name" field and name the application.

The screenshot shows the AWS Amplify web interface. On the left, there's a sidebar with links for 'All apps', 'Documentation', and 'Support'. The main area is titled 'Manual deploy' and contains fields for 'App name' (with placeholder text 'Give this app a name or we will generate a default for you'), 'Environment name' (with placeholder text 'Give this resource a meaningful environment name, like dev, test, or prod, or we will generate a'), and 'Method' (with options for 'Drag and drop' and 'Amazon S3').

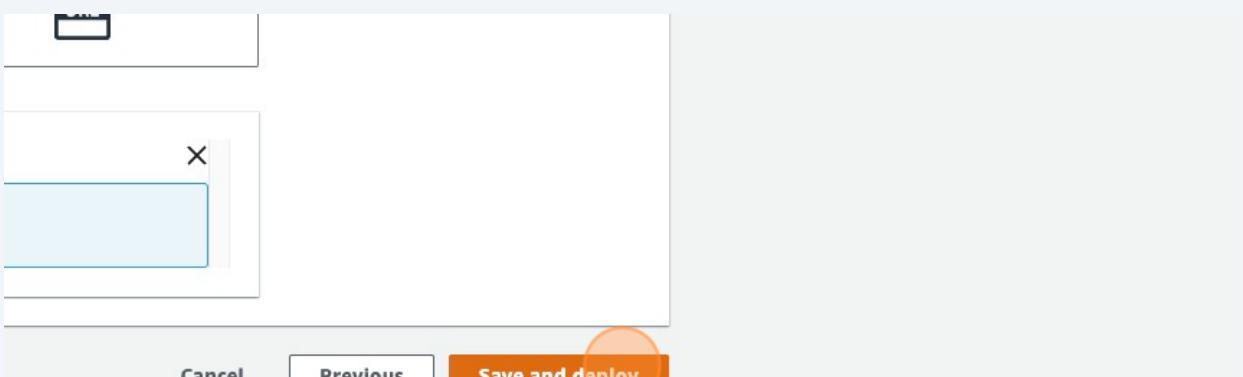
- 8 Click the "Environment name" field and give it a name as well.

This screenshot shows the same 'Manual deploy' interface as the previous one, but with the 'Environment name' field filled in with the value 'PowerofMath'. The rest of the interface remains the same, with the 'App name' field also containing a value.

9 Click "Choose files" and upload your index.html file created.

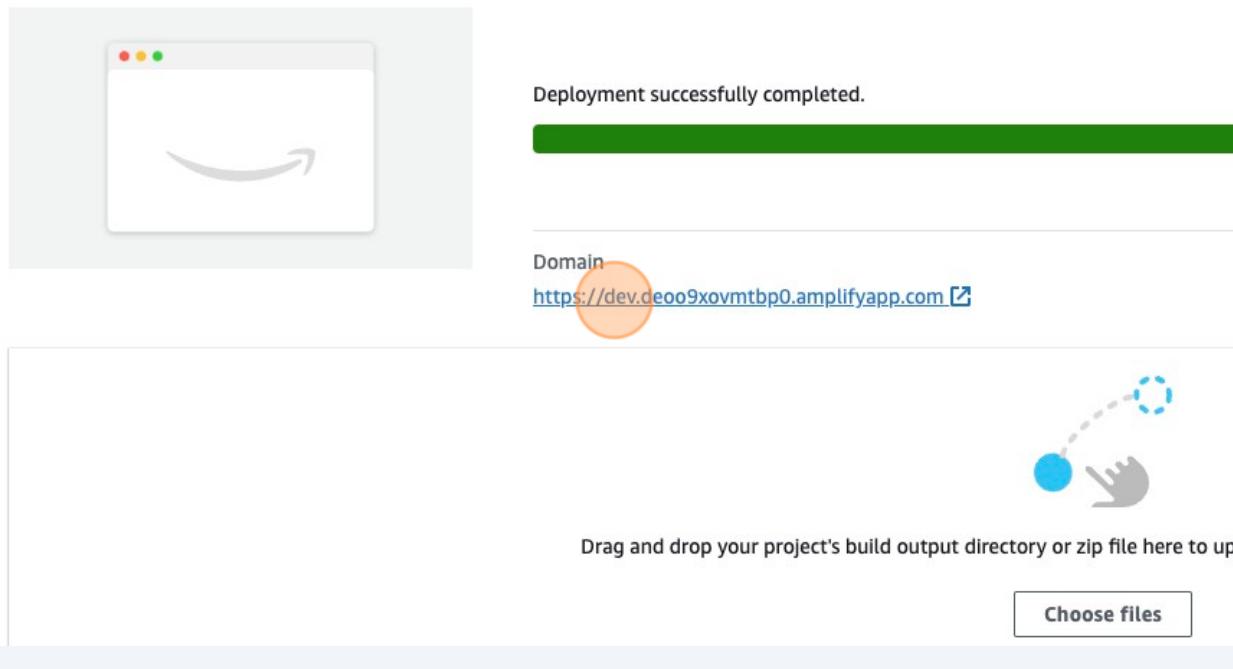


10 Click "Save and deploy"



- 11** Click the link created after files are uploaded under "Domain."

dev



- 12** "To the Power of Math!" is what the website should display in a new tab.

To the Power of Math!



13 In the search field type lambda, Right-click "Lambda" and open in a new tab.

The screenshot shows the AWS Amplify search interface. The search bar at the top contains the query 'lambda'. Below the search bar, there is a sidebar with various navigation links such as 'Amplify', 'Documentation', 'Marketplace', 'Blogs', 'Events', and 'Tutorials'. The main content area displays search results for 'lambda'. At the top of the results, it says 'Search results for 'lambda'' and 'Try searching with longer queries for more relevant results'. The first result is 'Services (7)' which is highlighted with a blue box. Below this, there are four cards: 'Lambda' (Run code without thinking about servers), 'CodeBuild' (Build and Test Code), 'AWS Signer' (Ensuring trust and integrity of your code), and 'Amazon Inspector'.

14 Click "Create function"

The screenshot shows the AWS Lambda console. The top navigation bar includes icons for back, forward, and help, along with region selection ('N. Virginia') and user information ('bbadams'). The main area displays a table of Lambda functions. The columns are 'Last fetched now', 'Actions', and 'Create function'. The 'Create function' button is highlighted with an orange circle. The table also includes filters for 'Package type' (Zip), 'Runtime' (Python 3.10), and 'Last modified' (last month).

- 15 Click the "Function name" field and give the Lambda function a name.

Basic information

Function name
Enter a name that describes the purpose of your function.
 

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
 Node.js 18.x 

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
[Change default execution role](#)

- 16 Click on the latest version of Python for the language.

Ruby 3.2

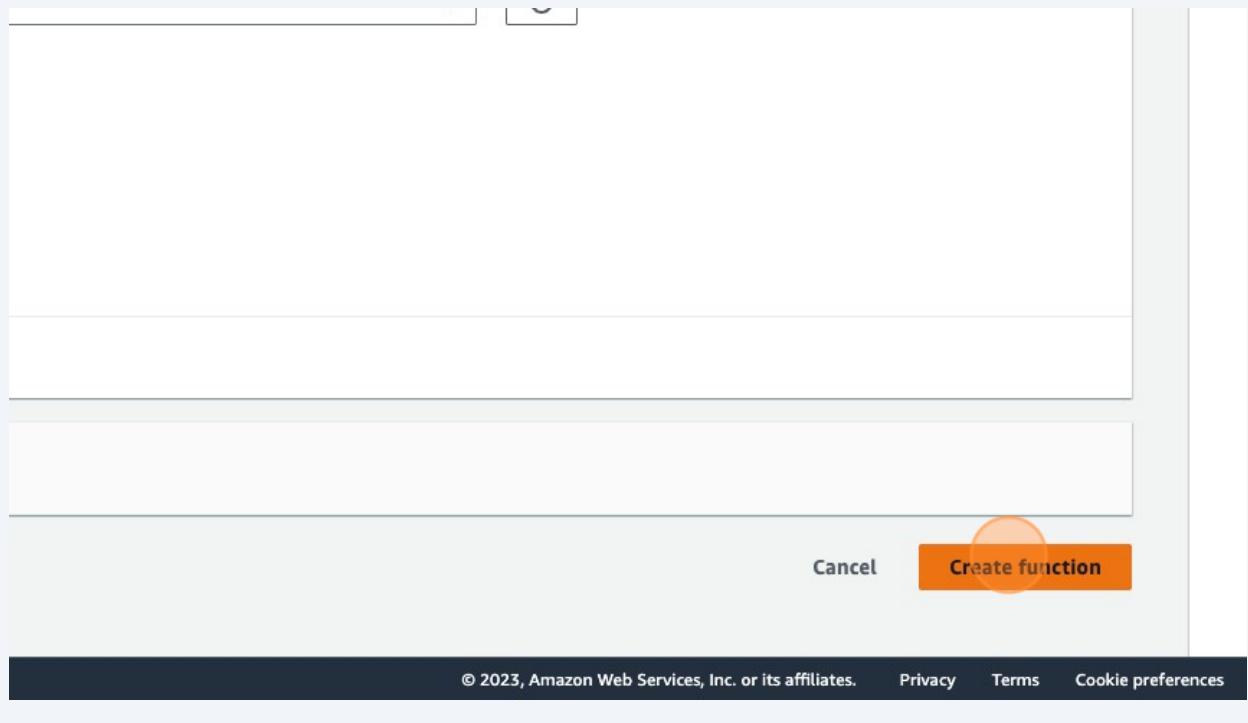
Other supported

Java 11
Java 8 on Amazon Linux 1
Java 8 on Amazon Linux 2
Node.js 14.x
Node.js 16.x
 Python 3.7 
Node.js 18.x

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

17 Click "Create function"



18 See the default code in the lambda_function tab.

A screenshot of the AWS Lambda code editor. The top navigation bar includes "Edit", "Find", "View", "Go", "Tools", "Window", a "Test" dropdown menu, and a "Deploy" button. The main area shows a file named "lambda_function.py" with the following code:

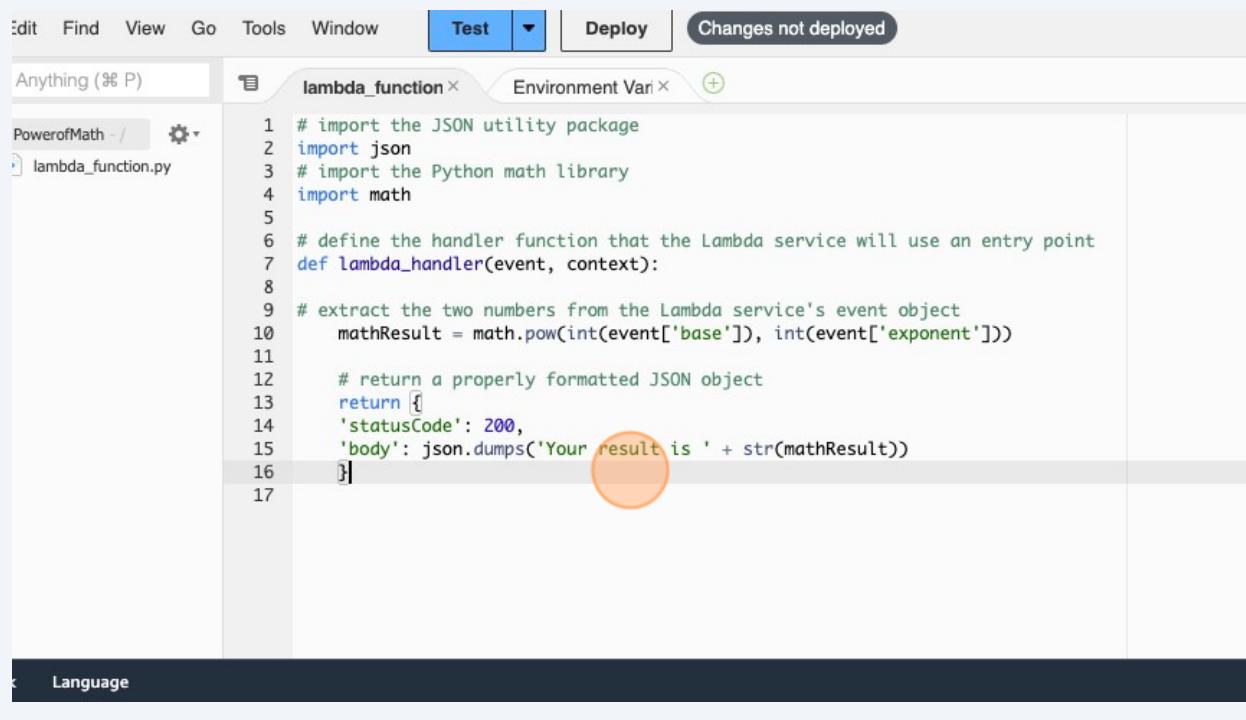
```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

The line "return {" is highlighted with a circular orange callout, indicating it is the current line of interest.

19

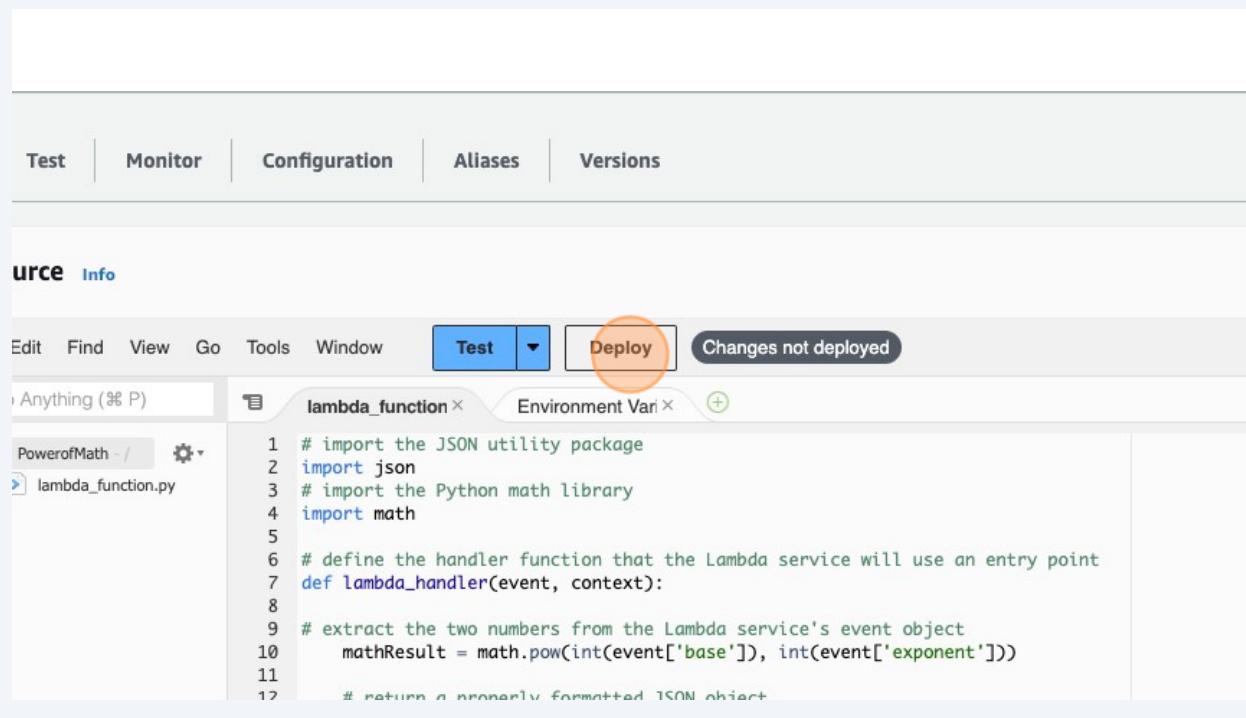
Replace the code with the lambda ORIGINAL code (it will be in the files with this project) and save it.



```
1 # import the JSON utility package
2 import json
3 # import the Python math library
4 import math
5
6 # define the handler function that the Lambda service will use an entry point
7 def lambda_handler(event, context):
8
9     # extract the two numbers from the Lambda service's event object
10    mathResult = math.pow(int(event['base']), int(event['exponent']))
11
12    # return a properly formatted JSON object
13    return {
14        'statusCode': 200,
15        'body': json.dumps('Your result is ' + str(mathResult))
16    }
17
```

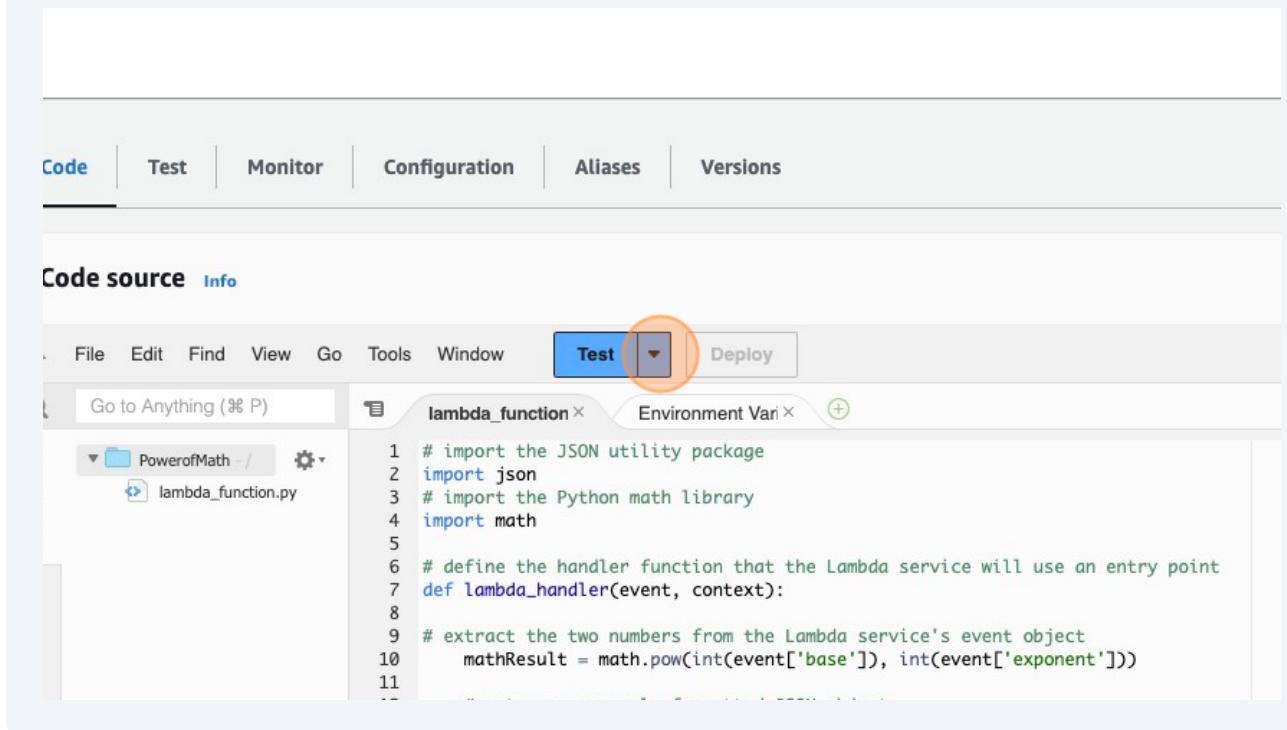
20

Click "Deploy"

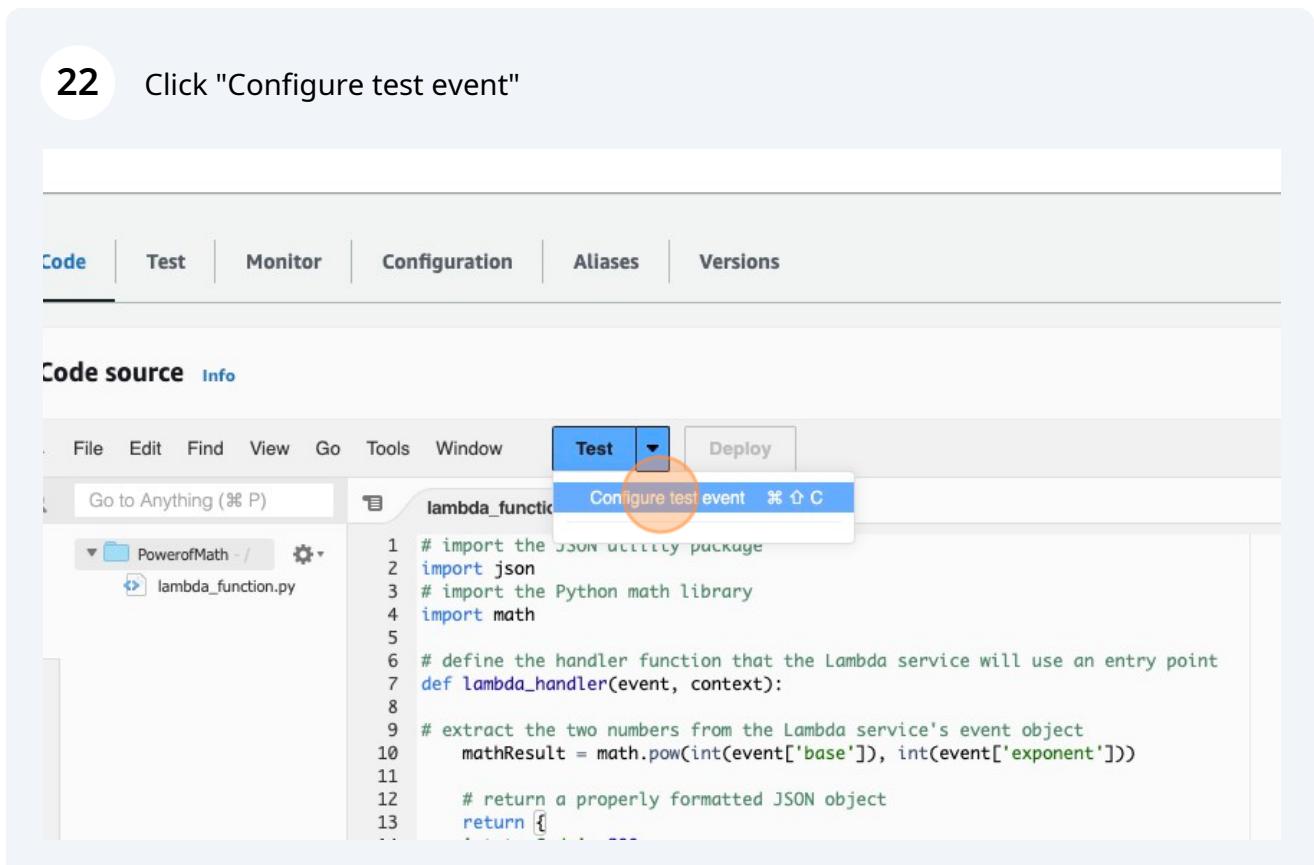


The screenshot shows the AWS Lambda function configuration interface. At the top, there are tabs for Test, Monitor, Configuration, Aliases, and Versions. Below these, there are tabs for Source and Info. The main area displays the Lambda function code in a code editor, which is identical to the one shown in step 19. The 'Deploy' button in the top navigation bar is highlighted with an orange circle.

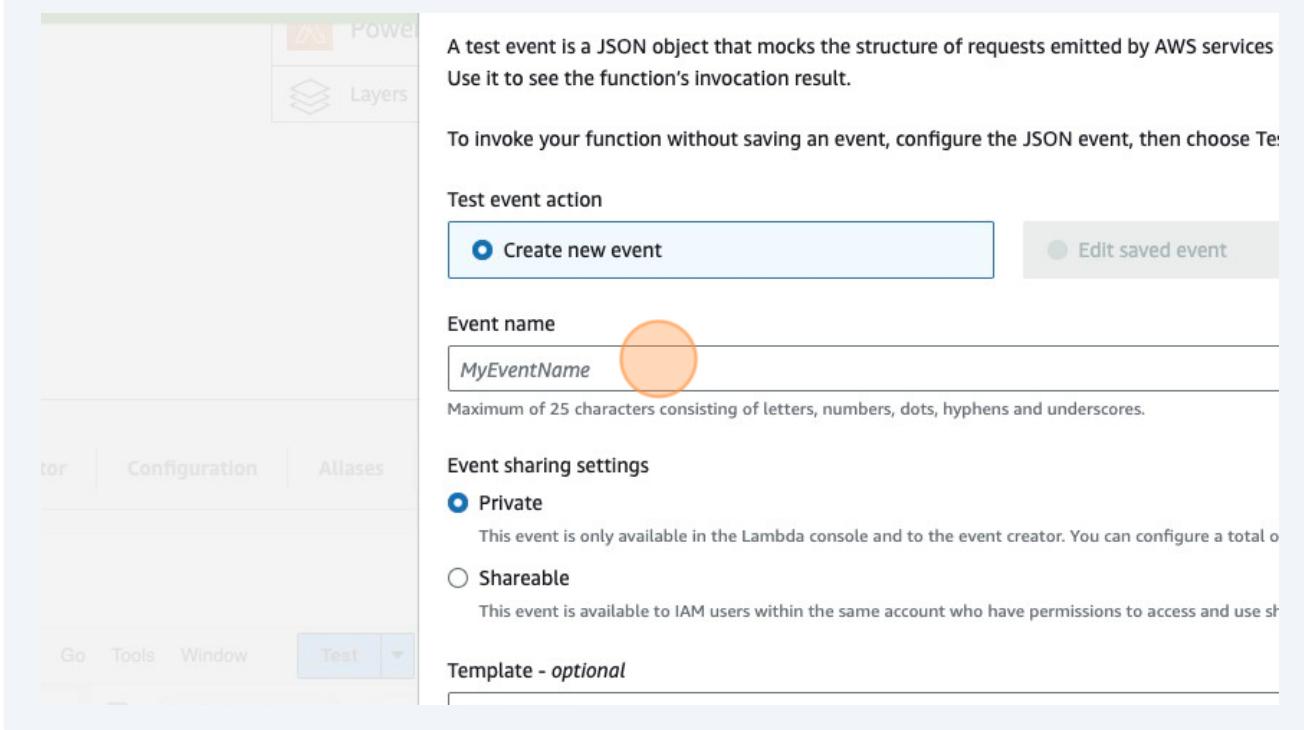
21 Click here.



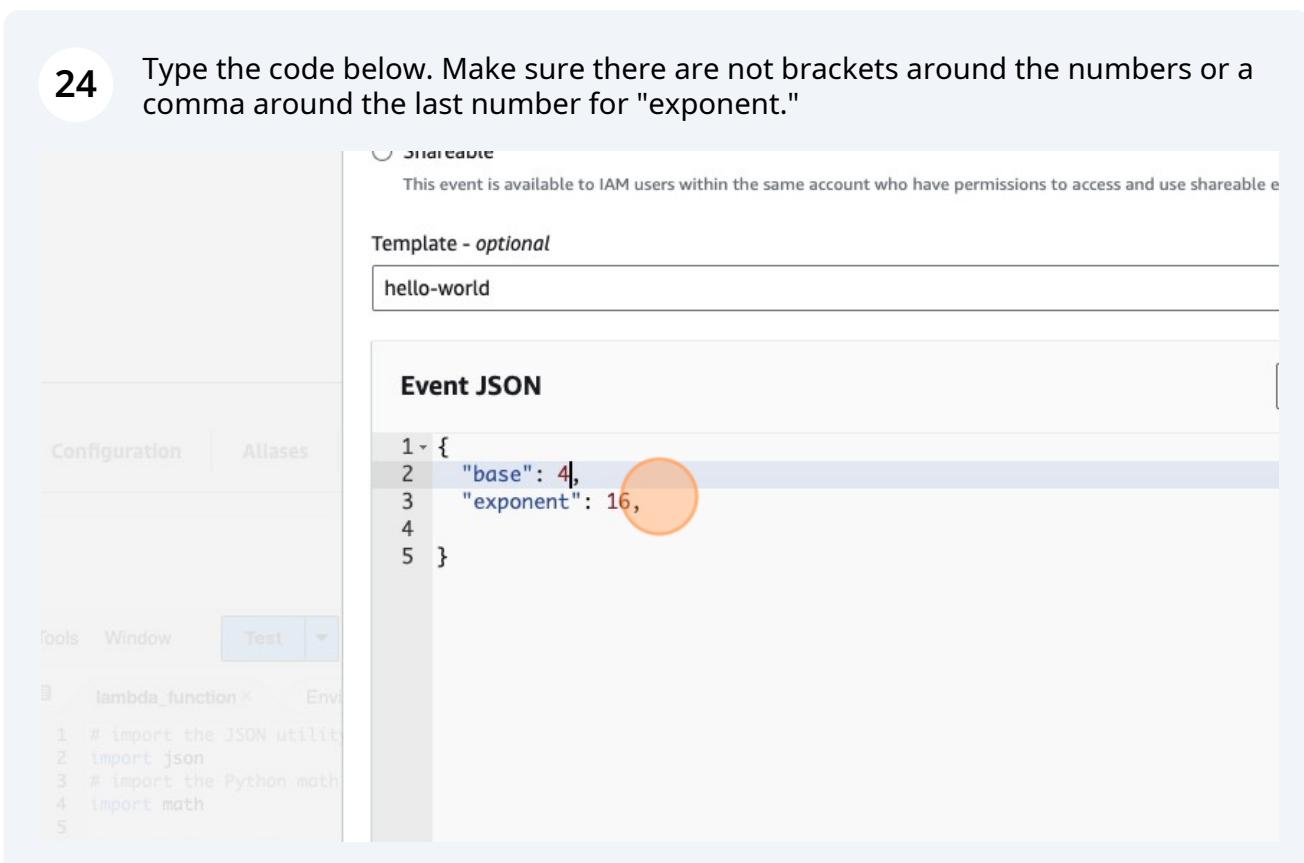
22 Click "Configure test event"



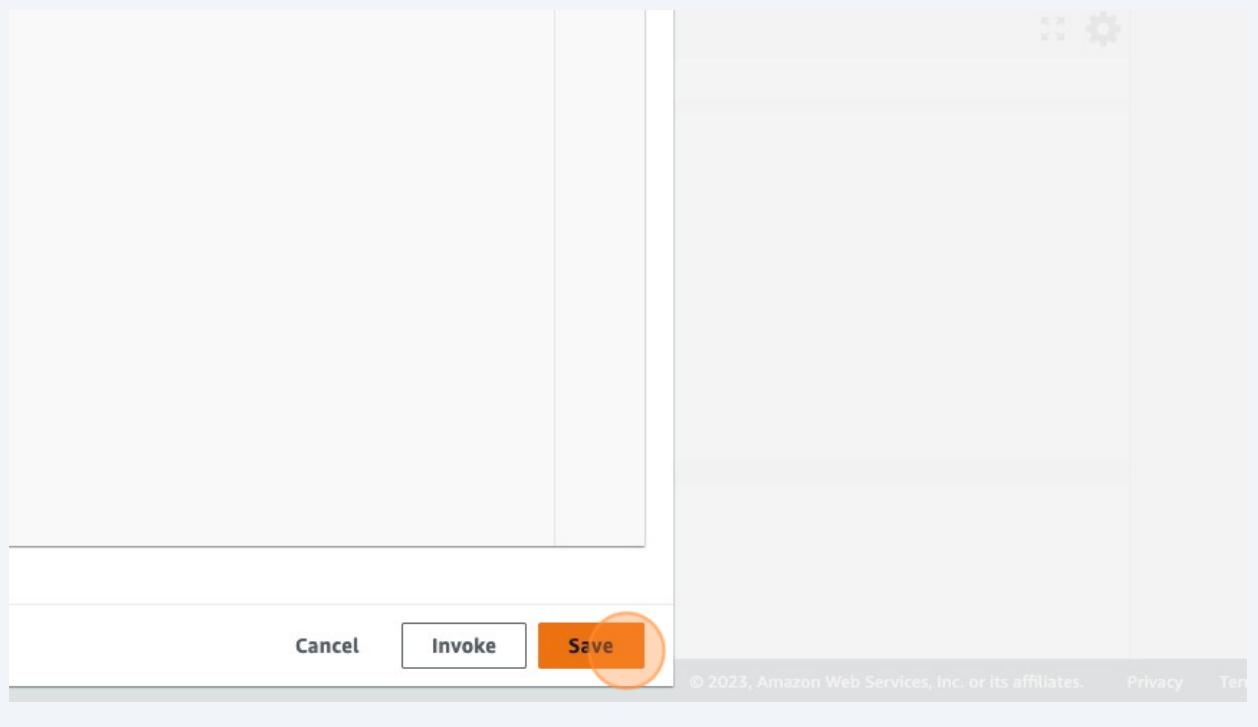
23 Click the "Event name" field and name the test.



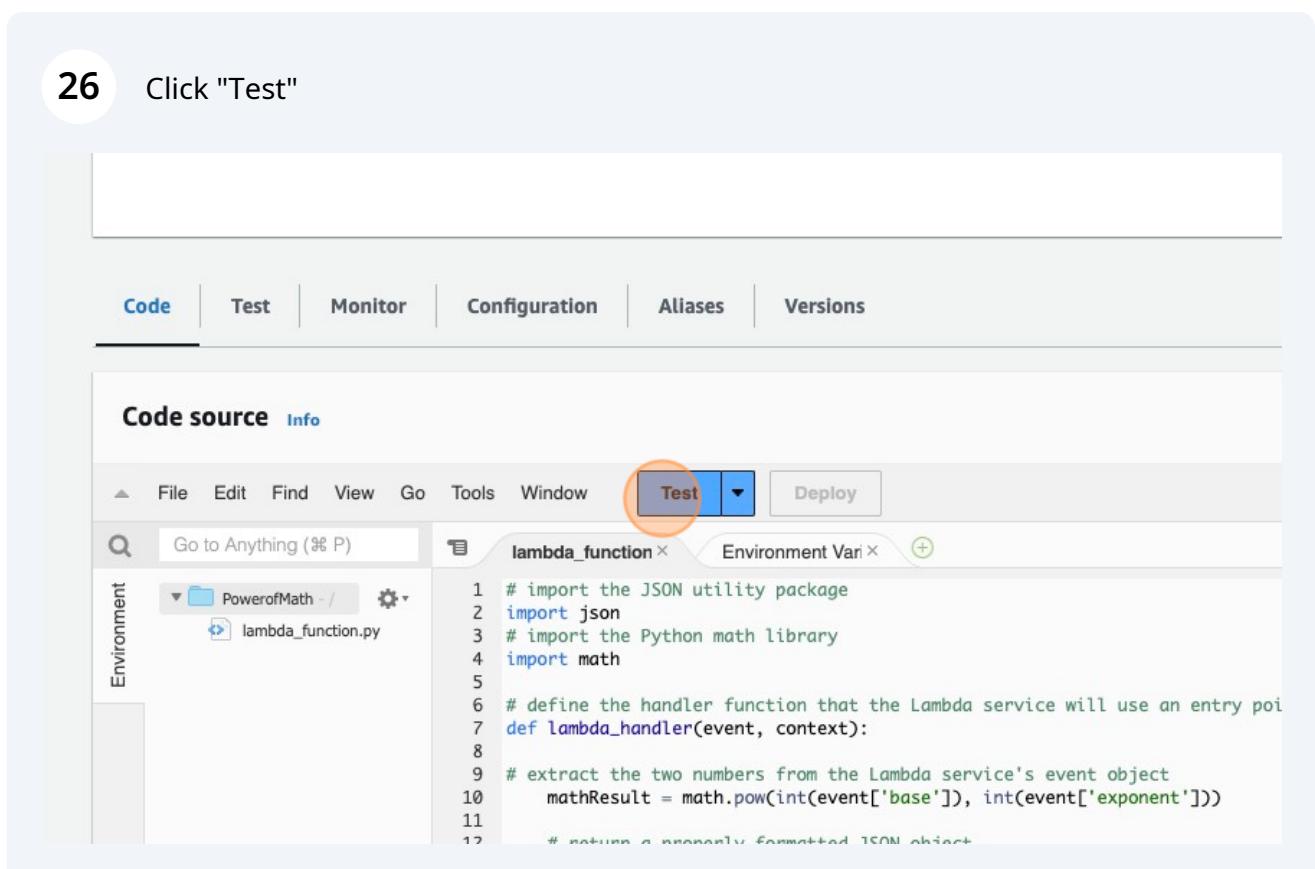
24 Type the code below. Make sure there are not brackets around the numbers or a comma around the last number for "exponent."



25 Click "Save"



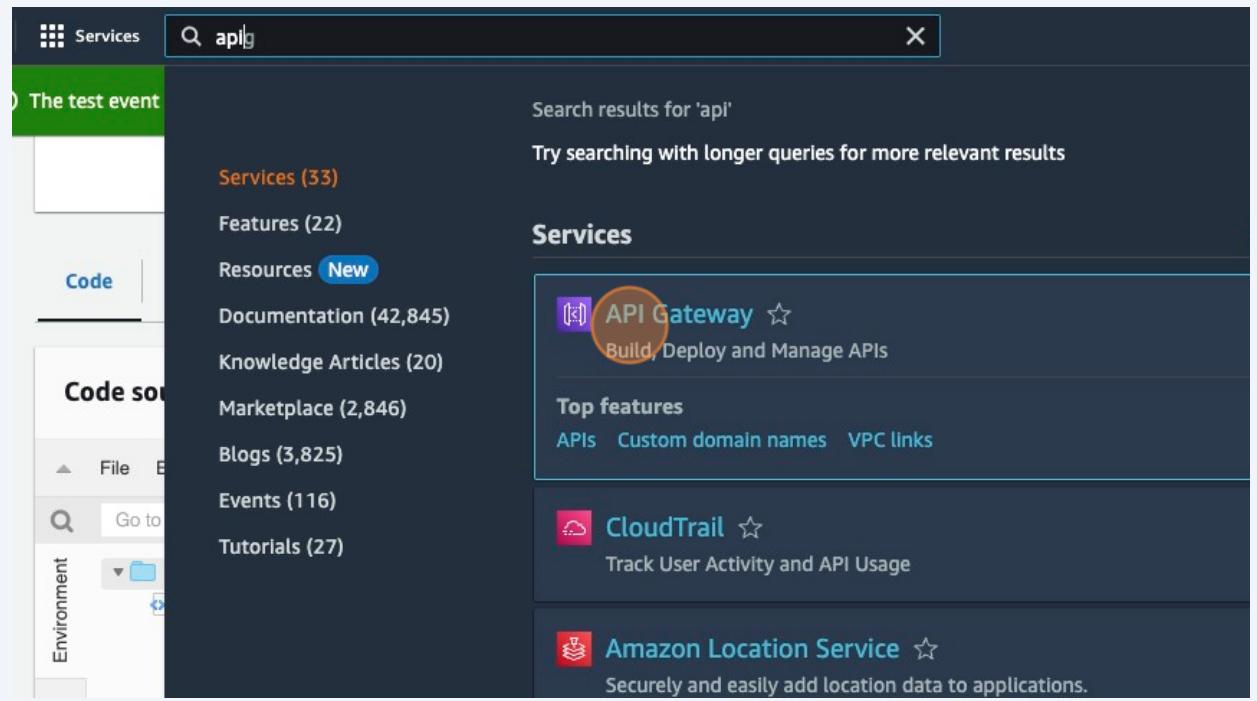
26 Click "Test"



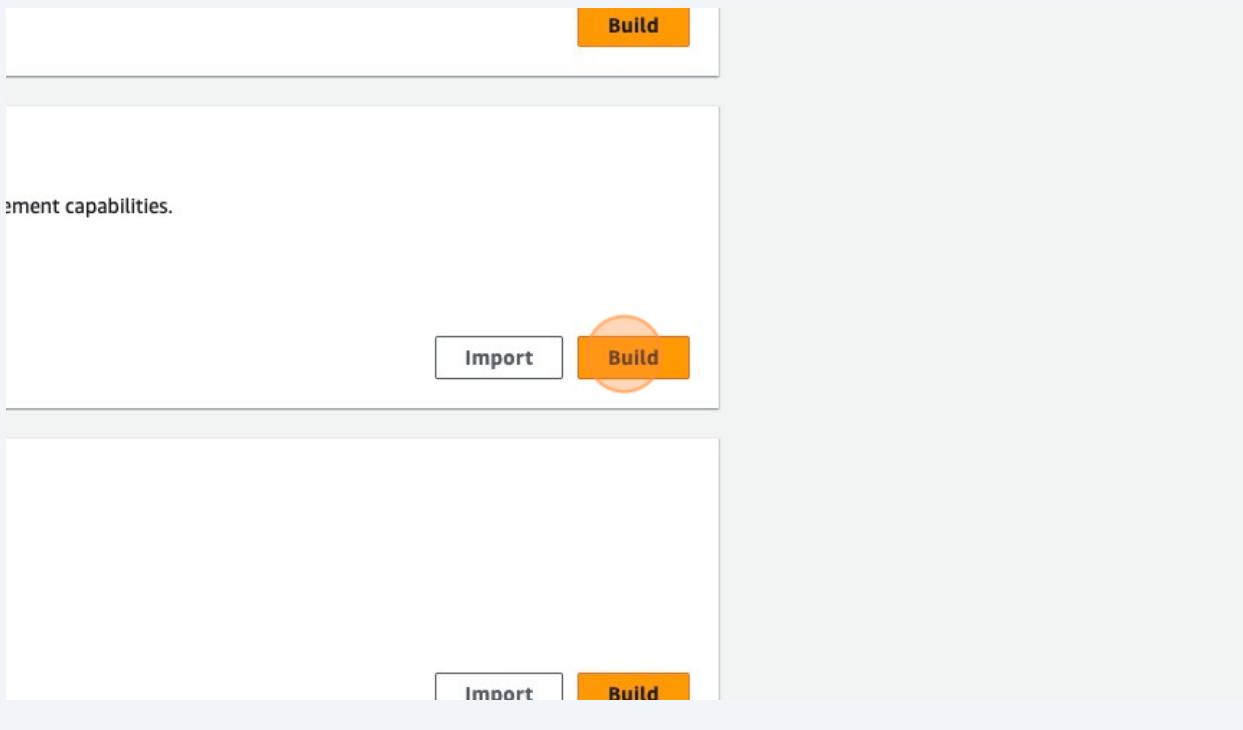
27 See your results here.

The screenshot shows the AWS Lambda Test interface. At the top, there are tabs for 'Test' and 'Deploy'. Below the tabs, there's a file tree on the left with 'function.py' selected. The main area displays the 'Execution result' tab, which includes the 'Test Event Name' ('PowerOfMathTestEvent'), the 'Response' (a JSON object with 'statusCode': 200 and 'body': '\"Your result is 4294967296.0\"'), 'Function Logs' (request and response details), and the 'Request ID' ('9dd86c87-1d56-4bce-82f5-0c8915d349fd'). An orange circle highlights the 'body' value in the response.

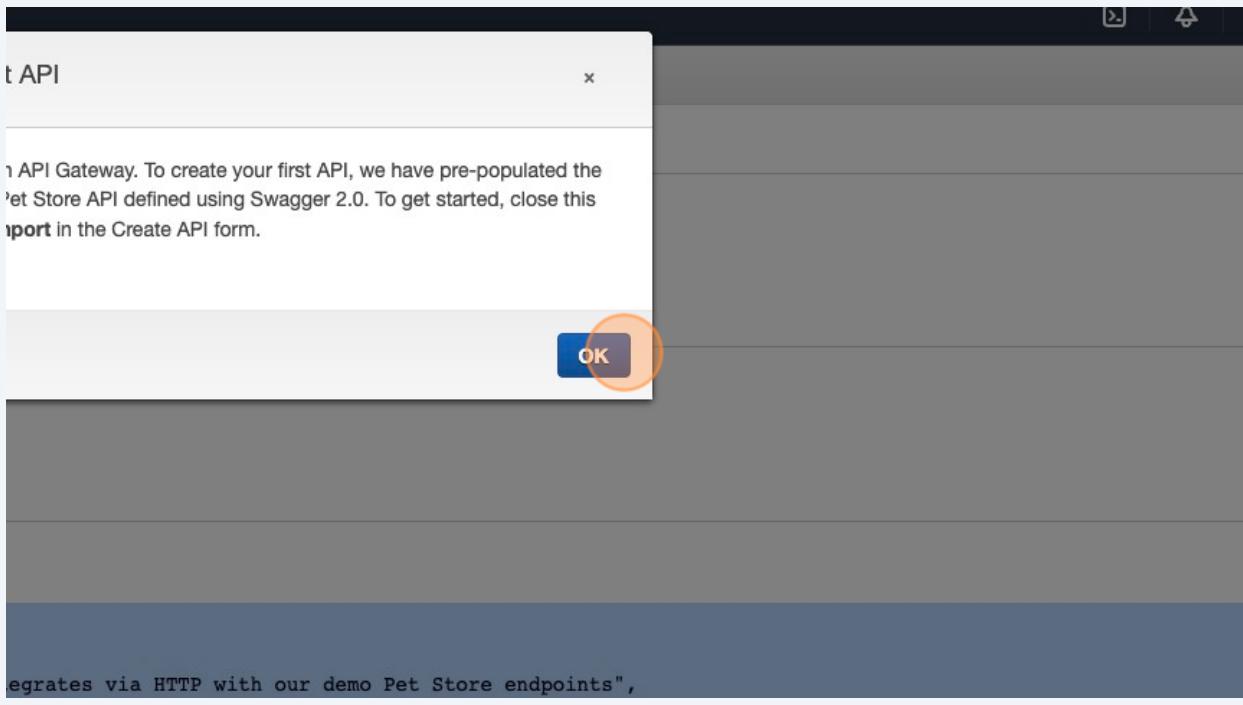
28 In the search field type api and Right-click "API Gateway" and open in a new tab.



29 Click on the second-to-last "Build."



30 Click "OK"



- 31 Click the "New API" field.

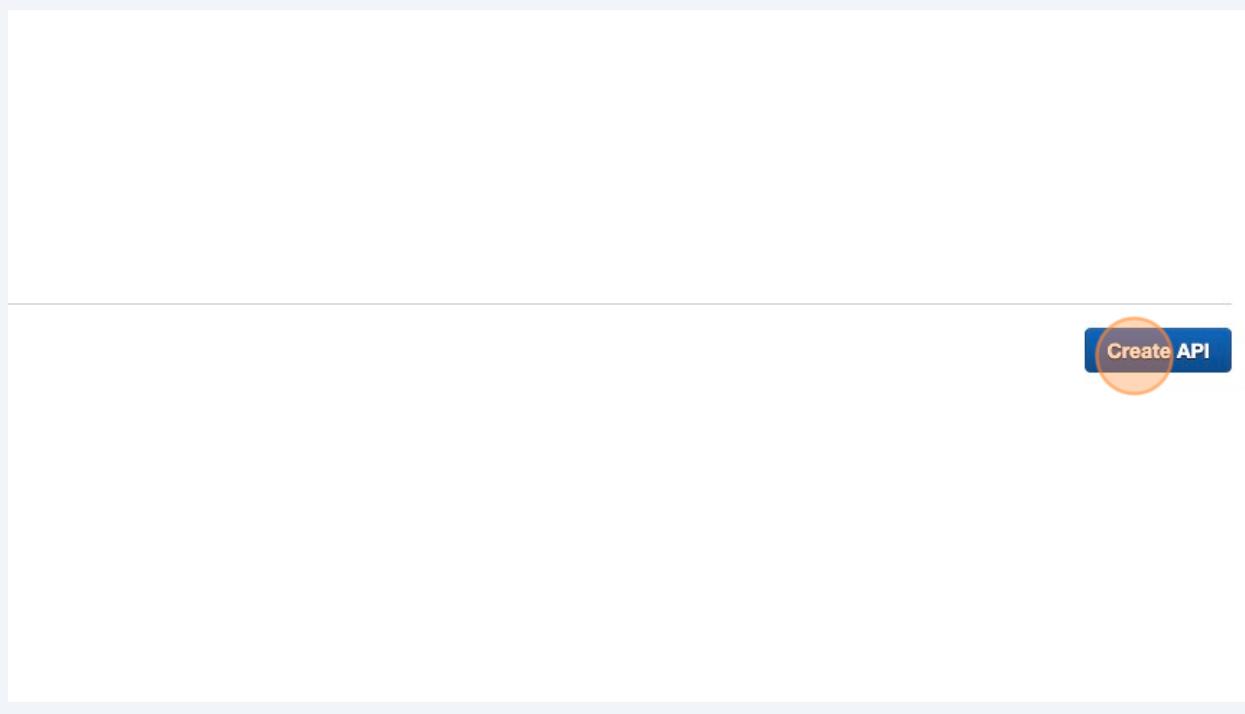
The screenshot shows the 'Create new API' section of the Amazon API Gateway console. At the top, it says 'Choose the protocol' with options for REST (selected) and WebSocket. Below that is the 'Create new API' heading. A note states that a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints. There are three radio button options: 'New API' (selected and highlighted with an orange circle), 'Import from Swagger or Open API 3', and 'Example API'. Under 'Example API', there is a code snippet in JSON format:

```
1 {
2   "swagger": "2.0",
3   "info": {
4     "description": "Your first API with Amazon API Gateway. This is a sample API that integrates with the AWS Lambda service and the Amazon S3 service to provide a simple Pet Store demo.",
5     "title": "PetStore"
6   },
7   "schemes": [
```

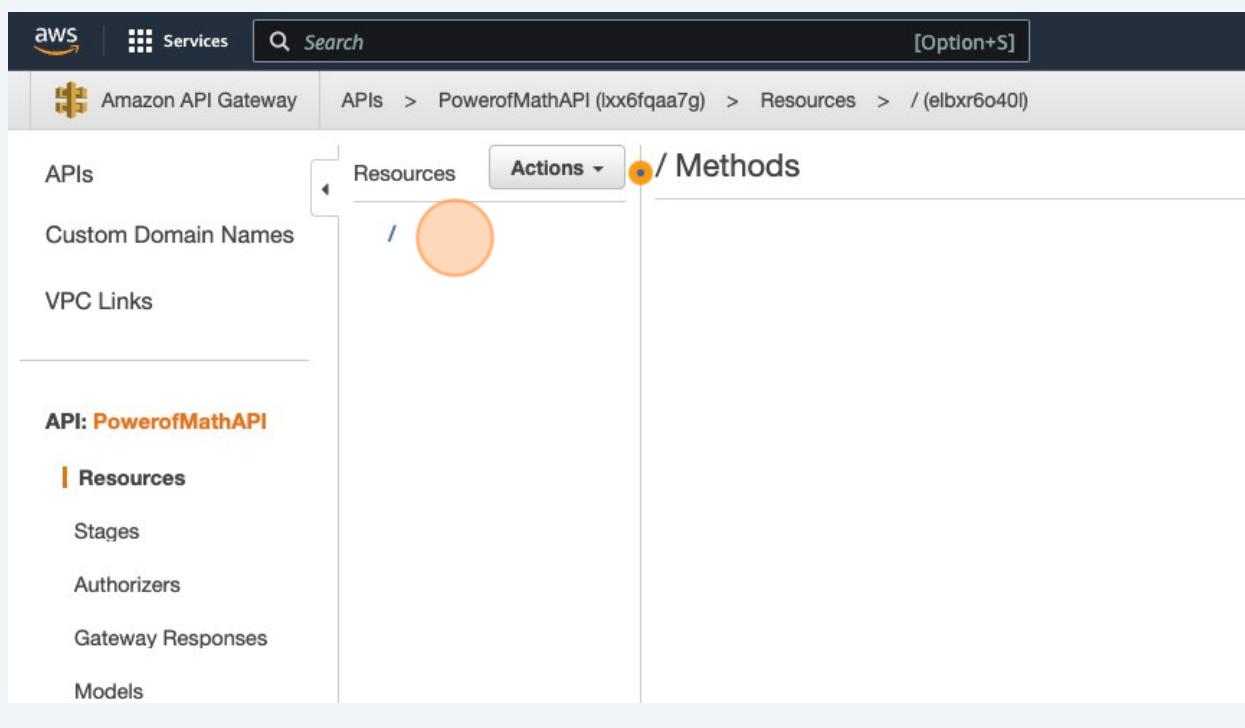
- 32 Click the "My API" field and name the API.

The screenshot shows the 'Create new API' settings page. It includes a note about what a REST API is, radio button options for 'New API' (selected), 'Import from Swagger or Open API 3', and 'Example API', and a note about endpoint types with a dropdown set to 'Regional'. The 'Settings' section allows users to enter a friendly name and description for their API. The 'API name*' field is highlighted with an orange circle and contains the value 'My API'. The 'Description' field is empty. The 'Endpoint Type' dropdown is set to 'Regional'.

33 Click "Create API"



34 Click on the "/"



35

The screenshot shows the AWS API Gateway console. The top navigation bar includes the AWS logo, Services dropdown, and a search bar. Below the navigation is a breadcrumb trail: Amazon API Gateway > APIs > PowerofMathAPI (ixx6fqa7g) > Resources > / (elbxr6o40l). On the left, a sidebar lists 'APIs', 'Custom Domain Names', and 'VPC Links'. Under 'API: PowerofMathAPI', the 'Resources' tab is selected, while 'Stages', 'Authorizers', 'Gateway Responses', and 'Models' are listed below it. The main content area shows the path '/ Methods'. A dropdown menu labeled 'Actions' is open, with the 'Create Method' option highlighted by an orange circle.

36 Click the Actions button and select "Create Method"

This screenshot shows the same AWS API Gateway interface as the previous one, but the 'Actions' dropdown menu is now fully visible. The 'Create Method' option is clearly highlighted with an orange circle. Other options in the RESOURCE ACTIONS section include 'Create Resource', 'Enable CORS', and 'Edit Resource Documentation'. The API ACTIONS section contains 'Deploy API', 'Import API', 'Edit API Documentation', and 'Delete API'.

37 Click this dropdown.

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with the AWS logo, 'Services' button, and a search bar. Below the navigation bar, the path 'APIs > PowerofMathAPI (lxx6fqa7g) > Resources > / (elbxr6o40l)' is displayed. On the left, a sidebar lists 'APIs', 'Custom Domain Names', and 'VPC Links'. Under 'API: PowerofMathAPI', the 'Resources' tab is selected, showing options like 'Stages', 'Authorizers', 'Gateway Responses', and 'Models'. In the main content area, the 'Actions' dropdown menu is open, showing options: 'PUT', 'DELETE', 'OPTIONS', 'HEAD', 'POST', and 'PATCH'. The 'POST' option is highlighted with a yellow circle and a checkmark, indicating it has been selected.

38 Click on Post and then click "Confirm creating Method." The checkmark.

This screenshot is identical to the previous one, showing the AWS Lambda console interface. The 'Actions' dropdown menu is open, and the 'POST' option is highlighted with a yellow circle and a checkmark. This indicates that the user has selected the 'POST' method for the resource.

39

Click the "Lambda Function" field and type the name of the lambda function you created exactly and the name should appear.

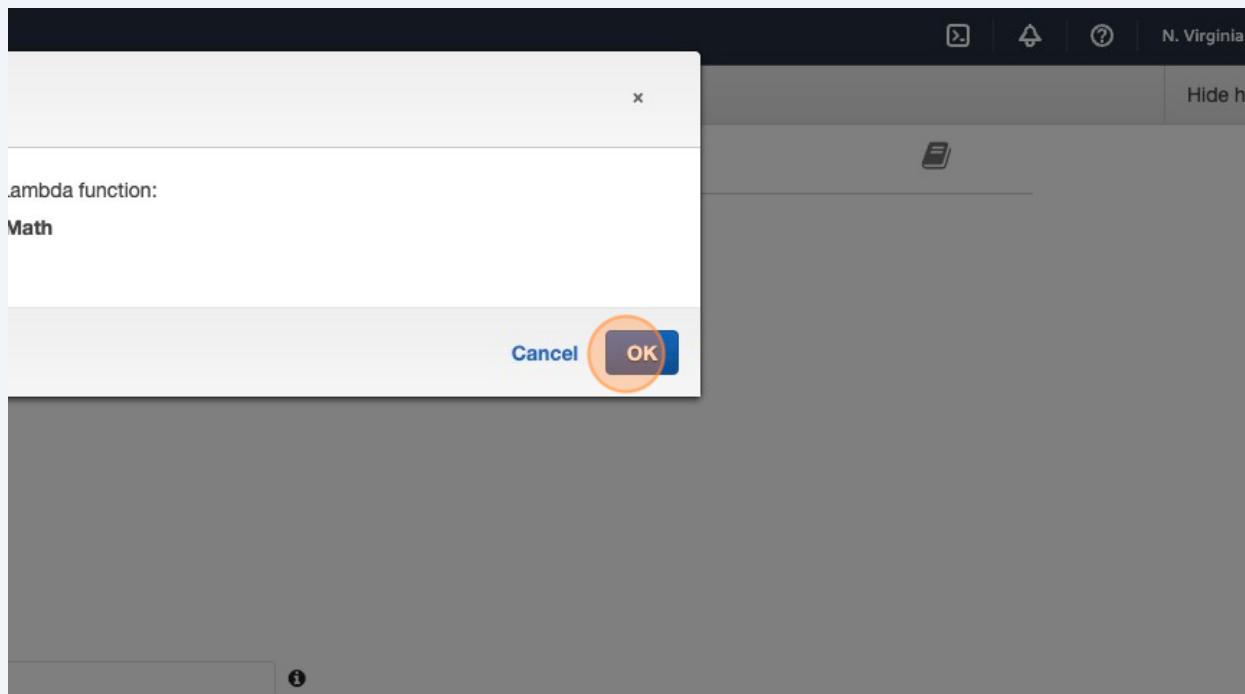
The screenshot shows the AWS Lambda configuration interface. At the top, there are three radio button options: "Mock" (with an info icon), "AWS Service" (with an info icon), and "VPC Link" (with an info icon). Below these is a checkbox labeled "Use Lambda Proxy integration" with an info icon. A dropdown menu labeled "Lambda Region" is set to "us-east-1". The "Lambda Function" field contains the text "PowerofMath" and has an info icon to its right. A button below it also displays "PowerofMath". The entire "Lambda Function" input field is circled in orange.

40

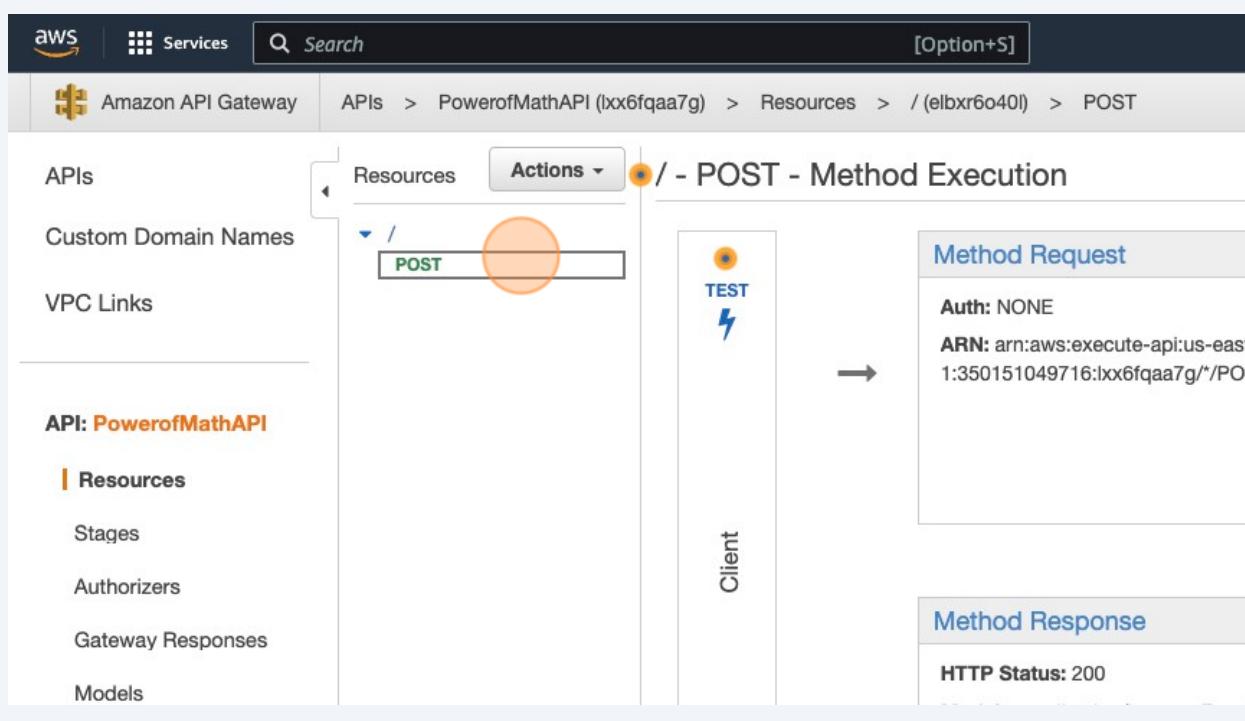
Click "Save"

The screenshot shows the AWS Lambda configuration interface. At the top left is a small rectangular input field with an info icon. In the center is a large blue "Save" button with white text, which is circled in orange. The rest of the interface is mostly blank white space.

41 Click "OK"



42 Click "POST"



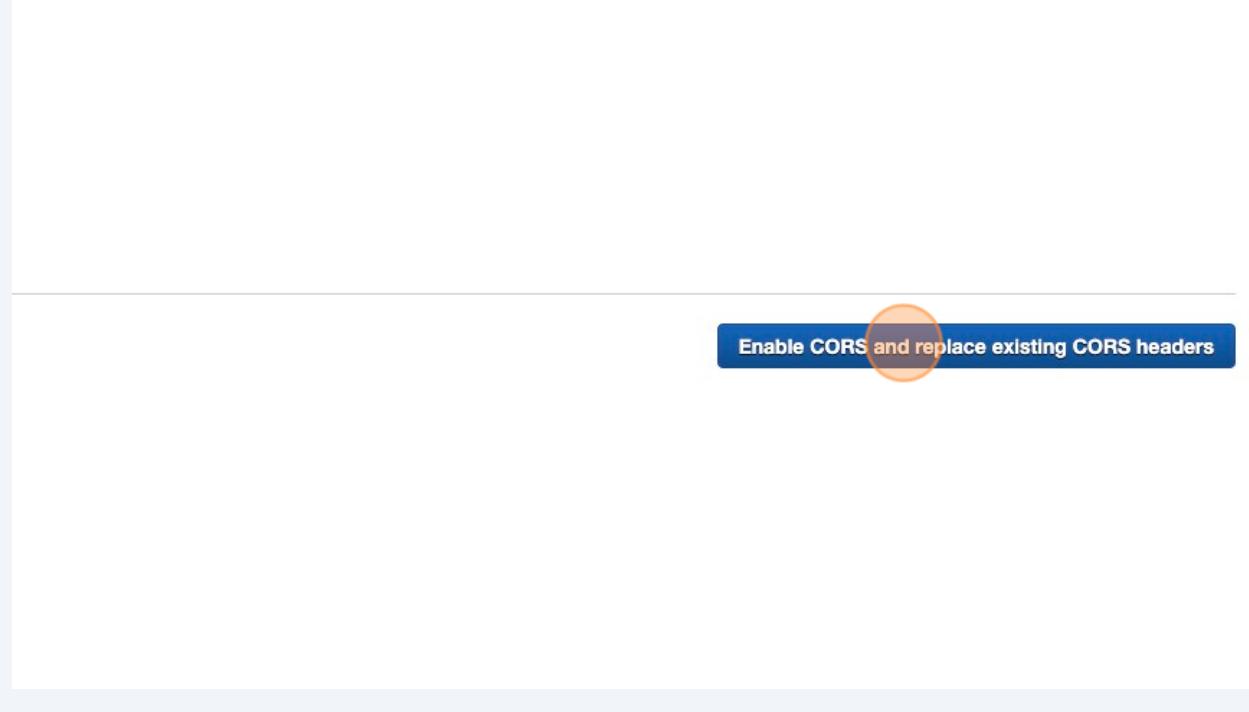
43 Click "Actions"

The screenshot shows the AWS API Gateway console. The top navigation bar includes the AWS logo, Services, a search bar, and a [Option+S] button. Below the navigation, the path is: APIs > PowerofMathAPI (lxx6fqa7g) > Resources > / (elbxr6o40l) > POST. On the left sidebar, under the API: PowerofMathAPI section, the Resources tab is selected. The main area displays the path / and the method POST. A dropdown menu labeled 'Actions' is open, with the 'TEST' button highlighted. To the right, there are sections for 'Method Request' (Auth: NONE, ARN: arn:aws:execute-api:us-east-1:350151049716:lxx6fqa7g/*/POST) and 'Method Response' (HTTP Status: 200).

44 Click "Enable CORS"

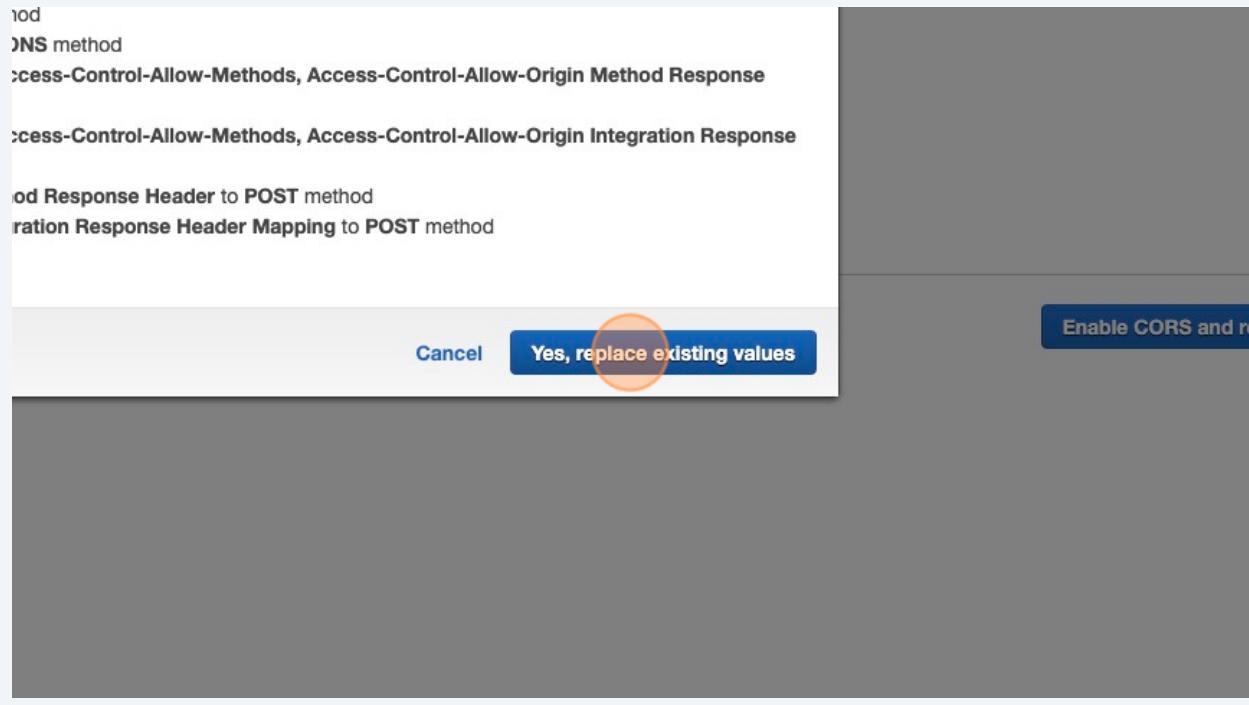
This screenshot is similar to the previous one, showing the AWS API Gateway console. The path is the same: APIs > PowerofMathAPI (lxx6fqa7g) > Resources > / (elbxr6o40l) > POST. The 'Actions' dropdown menu is open, and the 'Enable CORS' option is highlighted with a red circle. The 'Method Request' and 'Method Response' sections are also visible on the right.

45 Click "Enable CORS and replace existing CORS headers"



Enable CORS and replace existing CORS headers

46 Click "Yes, replace existing values"



Cancel Yes, replace existing values

Enable CORS and r

47 Click "Actions"

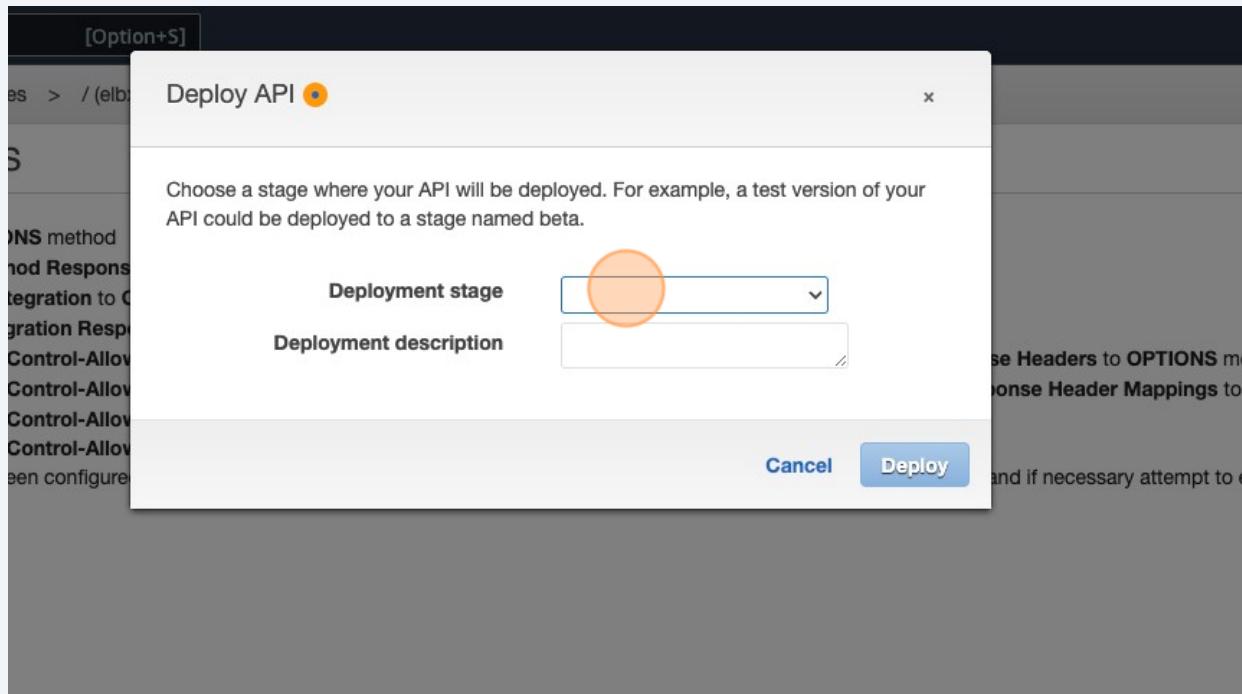
The screenshot shows the AWS API Gateway interface. The top navigation bar includes the AWS logo, Services dropdown, a search bar, and a [Option+S] button. Below the navigation, the path is: APIs > PowerofMathAPI (lxx6fqaa7g) > Resources > / (elbxr6o40l) > Enable CORS. On the left sidebar, under the API: PowerofMathAPI section, the Resources option is selected. The main content area shows a POST method under the Resources section. A dropdown menu labeled "Actions" is open, with the "Enable CORS" option highlighted and circled in orange. To the right of the method, a list of completed actions is displayed, each preceded by a green checkmark:

- Create OPTIONS method
- Add 200 Method Response with Empty Response Model
- 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 Headers
- Add Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Response Headers
- Add Access-Control-Allow-Origin Method Response Header
- Add Access-Control-Allow-Origin Integration Response Header

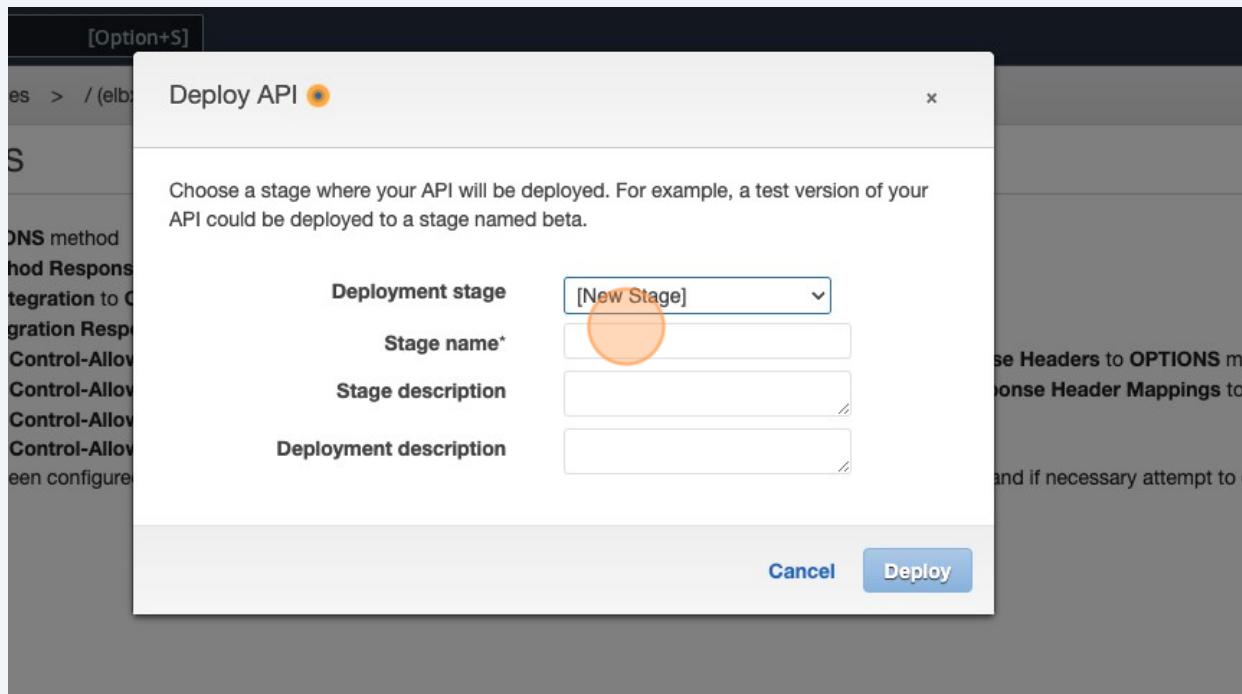
48 Click "Deploy API"

This screenshot is similar to the previous one, showing the AWS API Gateway interface for the POST method. The "Actions" dropdown menu is open, and the "Deploy API" option is highlighted and circled in orange. The rest of the interface and the list of completed actions are identical to the previous screenshot.

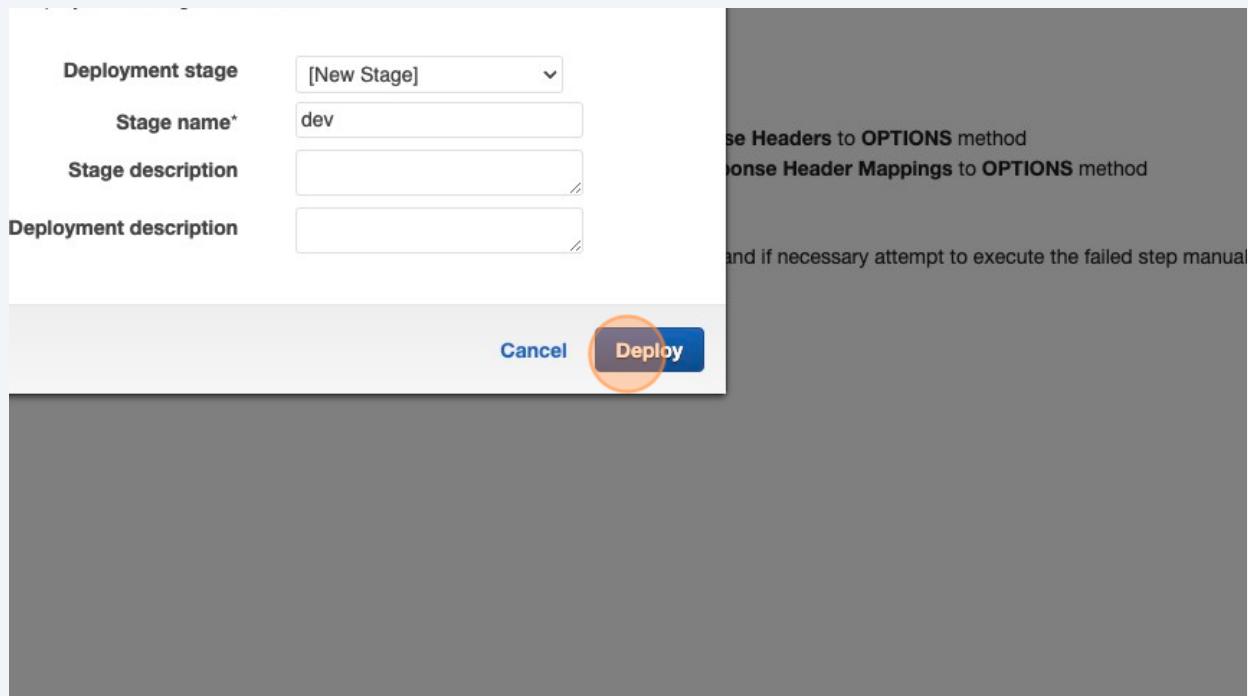
49 Click this dropdown.



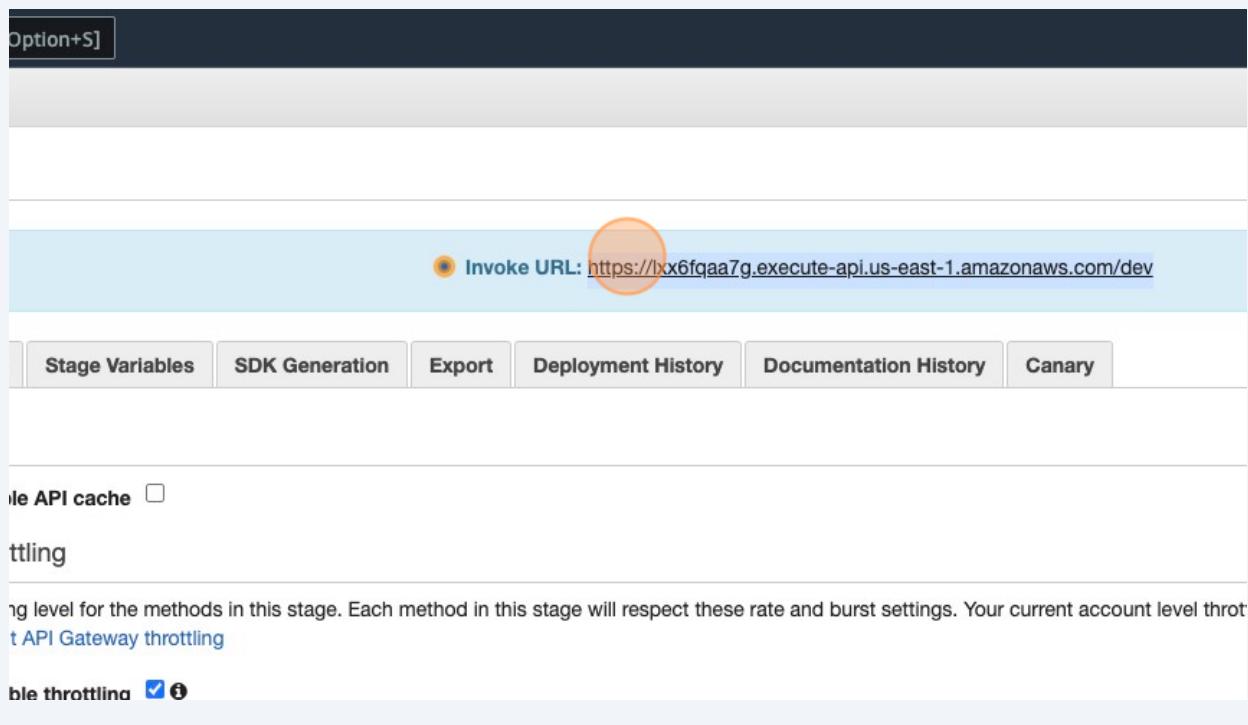
50 Click the "Stage name" field and give the stage a name.



51 Click "Deploy"



52 Copy and paste the Invoke URL and save it on a textpad. You will need it later.



53 Click "Resources"

The screenshot shows the 'dev Stage Editor' for the 'PowerofMathAPI'. On the left, a sidebar lists various API components: APIs, Custom Domain Names, VPC Links, API: PowerofMathAPI (with 'Resources' highlighted and circled in orange), Stages (with 'dev' selected), Authorizers, Gateway Responses, Models, Resource Policy, and Documentation. The main panel displays 'Cache Settings' with an 'Enable API cache' checkbox (unchecked). Below it is 'Default Method Throttling' with settings for 'Rate' (10000 requests) and 'Burst' (5000 requests). The top right features tabs for 'Settings' (selected), 'Logs/Tracing', 'Stage Variables', and 'SDK Gen'.

54 Click "POST"

The screenshot shows the 'Resources' page for the 'PowerofMathAPI'. The left sidebar includes 'APIs', 'Custom Domain Names', 'VPC Links', and 'API: PowerofMathAPI' (with 'Resources' selected). In the main area, under the path 'OPTIONS /', the 'POST' method is highlighted and circled in orange. To the right, a detailed view of the 'OPTIONS' method shows 'Mock Endpoint' settings: 'Authorization' set to 'None' and 'API Key' set to 'Not required'. A green 'Save' button is visible on the far right.

55 Click here on the blue lightning bolt.

The screenshot shows the AWS API Gateway interface. In the center, there's a 'Method Request' section with a 'Client' icon containing a blue lightning bolt icon. This lightning bolt icon is highlighted with a red circle. To the right, there's a 'Method Response' section showing 'HTTP Status: 200'. At the top, the URL path is visible: 'Amazon API Gateway > APIs > PowerofMathAPI (lxx6fqaa7g) > Resources > / (elbxr6o40) > POST'.

56 Click here and enter the same test configuration as you did in step 24. You can type in different numbers.

The screenshot shows the 'Method Request' configuration page. On the left, there's a sidebar with options like 'Responses', 'Policy', 'Integration', 'Test', 'Logs', and 'Metrics'. The main area has sections for 'Headers' (empty), 'Stage Variables' (empty), and 'Request Body'. The 'Request Body' section contains a text input field with a placeholder 'Type your request body here' and a small info icon, which is highlighted with a red circle.

57 Click "Test"



58 See your results on the page.

Authorization and directly invokes your method

Test: /

Is: 200

Time: 233 ms

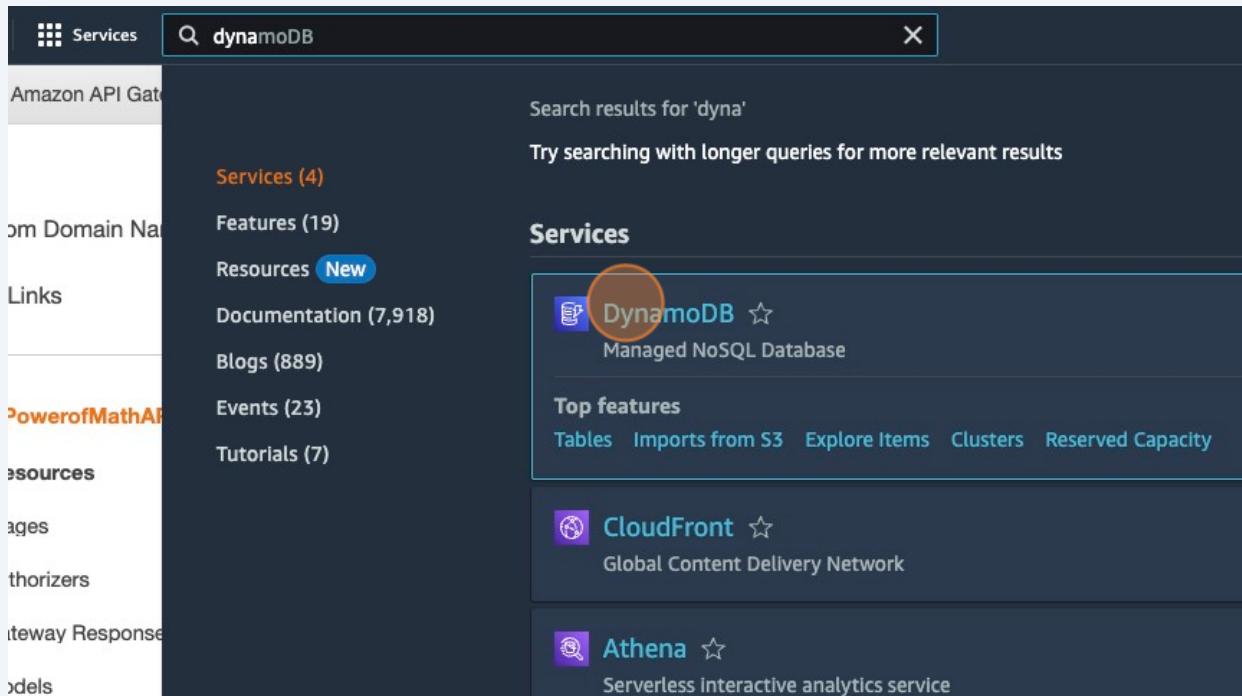
Response Body

```
statusCode": 200, "body": "\"Your result is 256.0\""}  
Response Headers
```

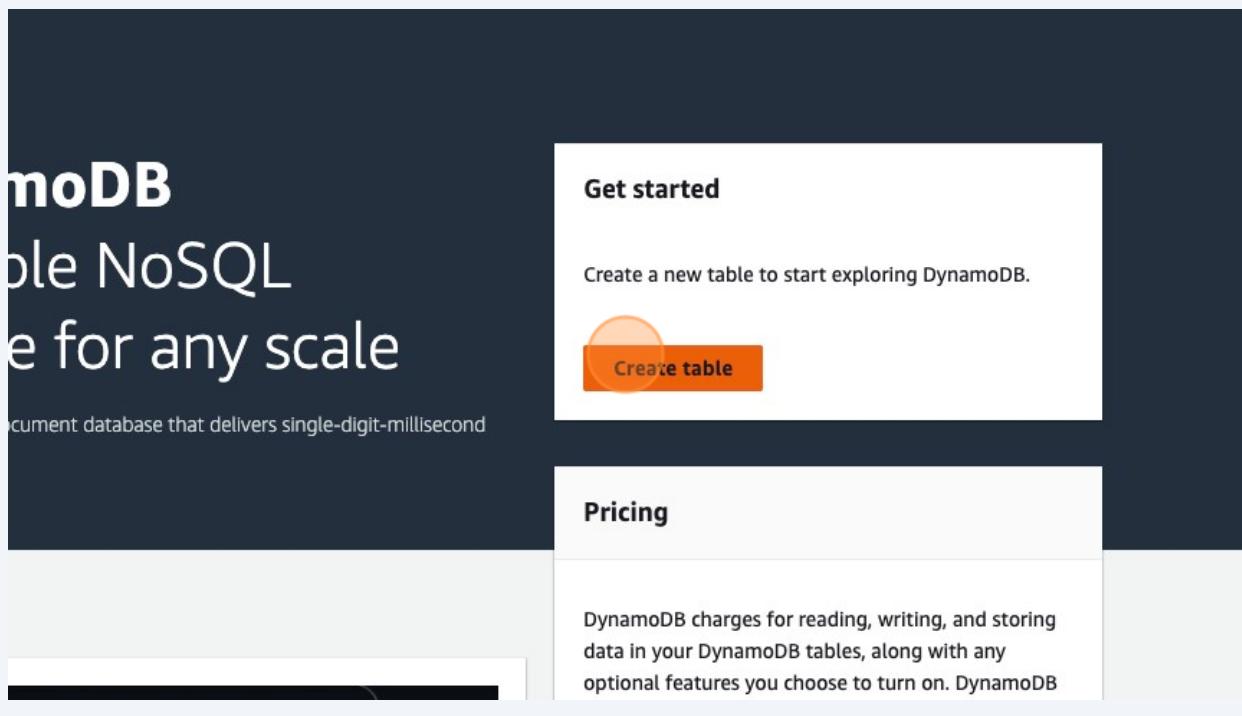
Access-Control-Allow-Origin": ["*"], "Content-Type": ["application/json"], "X-Amzn-Trace-Id": ["Root=1-64c1dd0d-1322e21599b467daa4fd46dd; SampledIngest=d390e599:0"]}

Action log for request 748cf274-138a-483d-a9af-c5745c5451a6

59 In search field type in dynamo and Right-click "DynamoDB" and open in a new tab.



60 Click "Create table"



61 Give it a name.

Table name

This will be used to identify your table.

PowerofMathDatabase

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

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.

Enter the partition key name

String

1 to 255 characters and case sensitive.

Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name

String

1 to 255 characters and case sensitive.

Table settings

62 Click the "Enter the partition key name" field and type ID.

Table name

This will be used to identify your table.

PowerofMathDatabase

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

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.

Enter the partition key name

String

1 to 255 characters and case sensitive.

Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

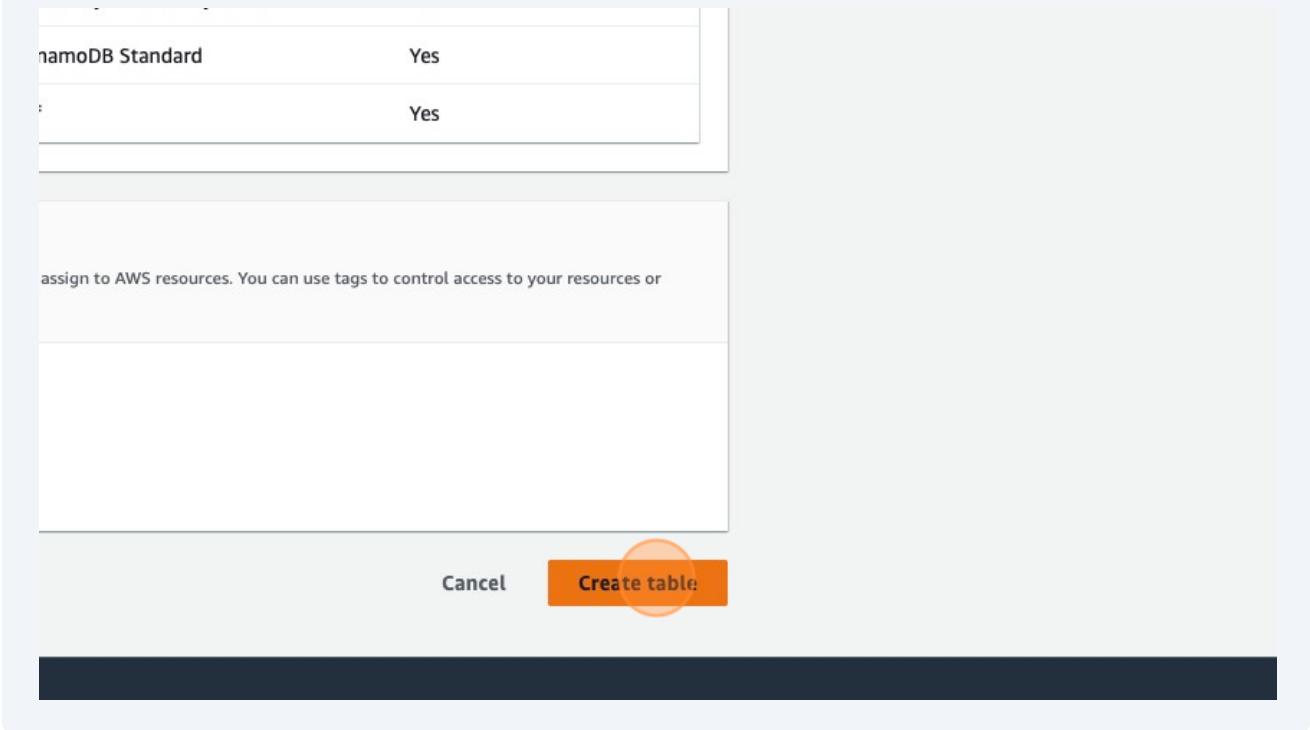
Enter the sort key name

String

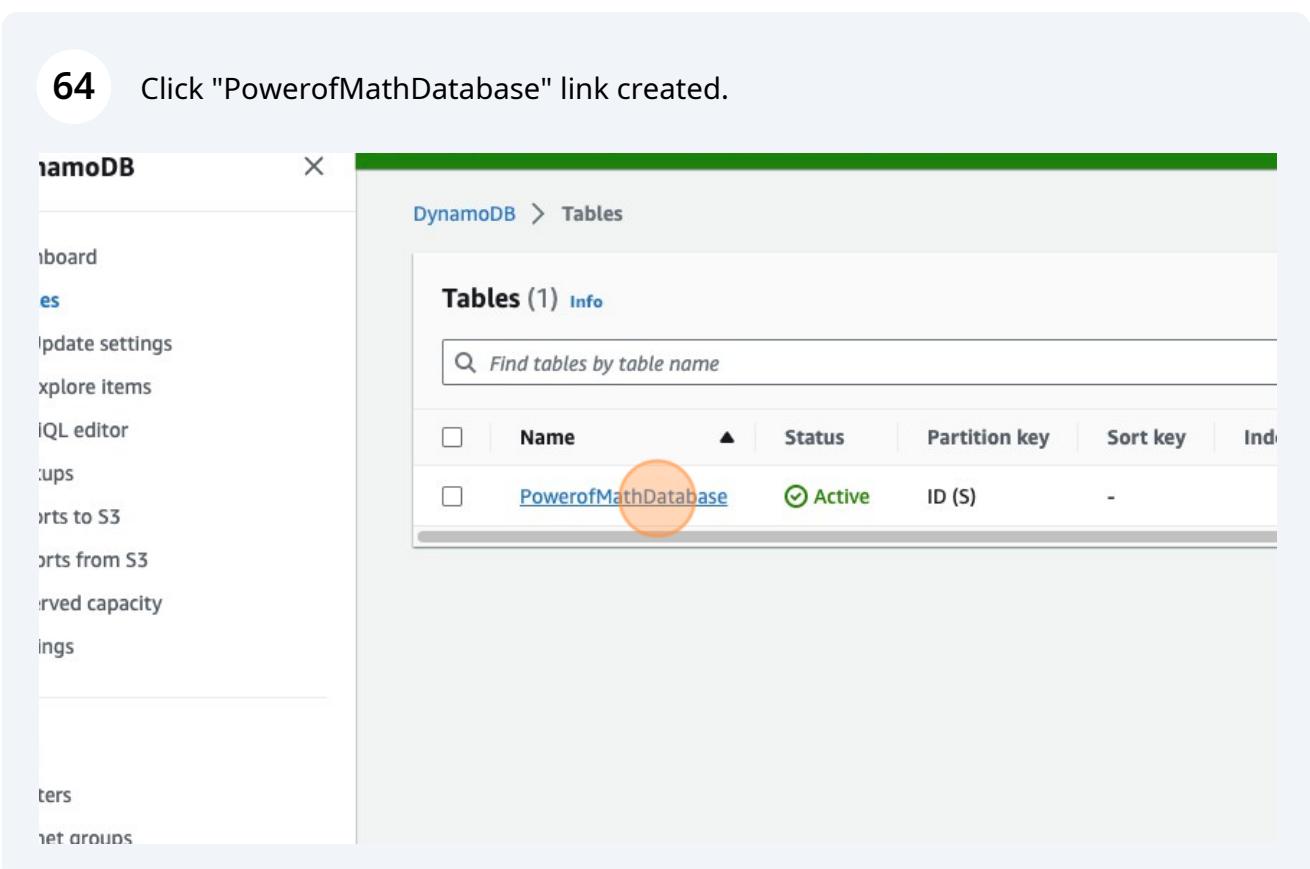
1 to 255 characters and case sensitive.

Table settings

63 Click "Create table"



64 Click "PowerofMathDatabase" link created.



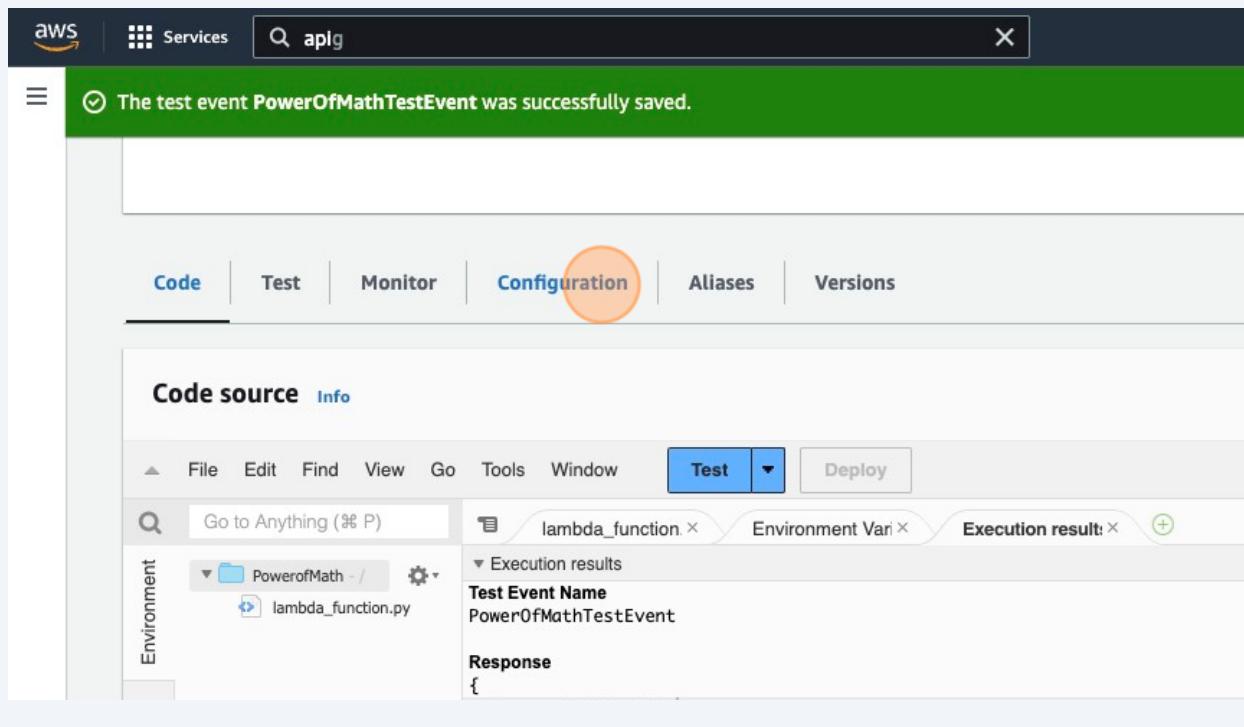
- 65 Click on "Additional info."

The screenshot shows the 'General information' section of a DynamoDB table configuration. It includes fields for Partition key (ID, String), Sort key (-), Alarms (No active alarms), Point-in-time recovery (PITR) (Off), and a large orange circle highlighting the 'Additional info' button.

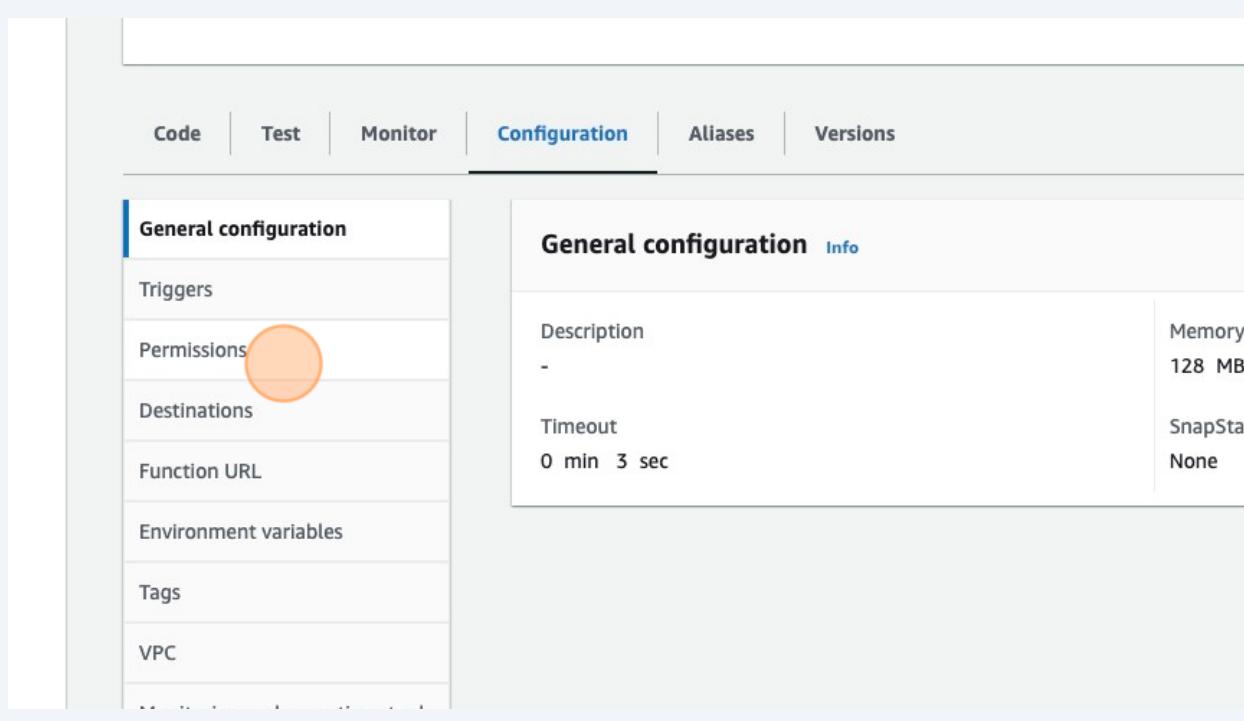
- 66 Copy the ARN and save it on a textpad. You will need it later.

The screenshot shows the 'Additional info' section of a DynamoDB table configuration. It includes fields for No active alarms, Off, Table class (DynamoDB Standard), Indexes (0 globals, 0 locals), Replication Regions (0 Regions), Encryption (Owned by Amazon), and an orange circle highlighting the 'Amazon Resource Name (ARN)' field containing the value arn:aws:dynamodb:us-east-1:350151049716:table/PowerofMath.

67 Click "Configuration"



68 Click "Permissions"



69 Click the Role name link.

The test event PowerOfMathTestEvent was successfully saved.

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

Monitoring and operations tools

Execution role

Role name
PowerofMath-role-49aq7y91

Resource summary

Amazon CloudWatch Logs
3 actions, 2 resources

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

By action | **By resource**

70 Click "Add permissions"

arn:aws:iam::350151049716:role/service-role/PowerofMath-role-49aq7y91

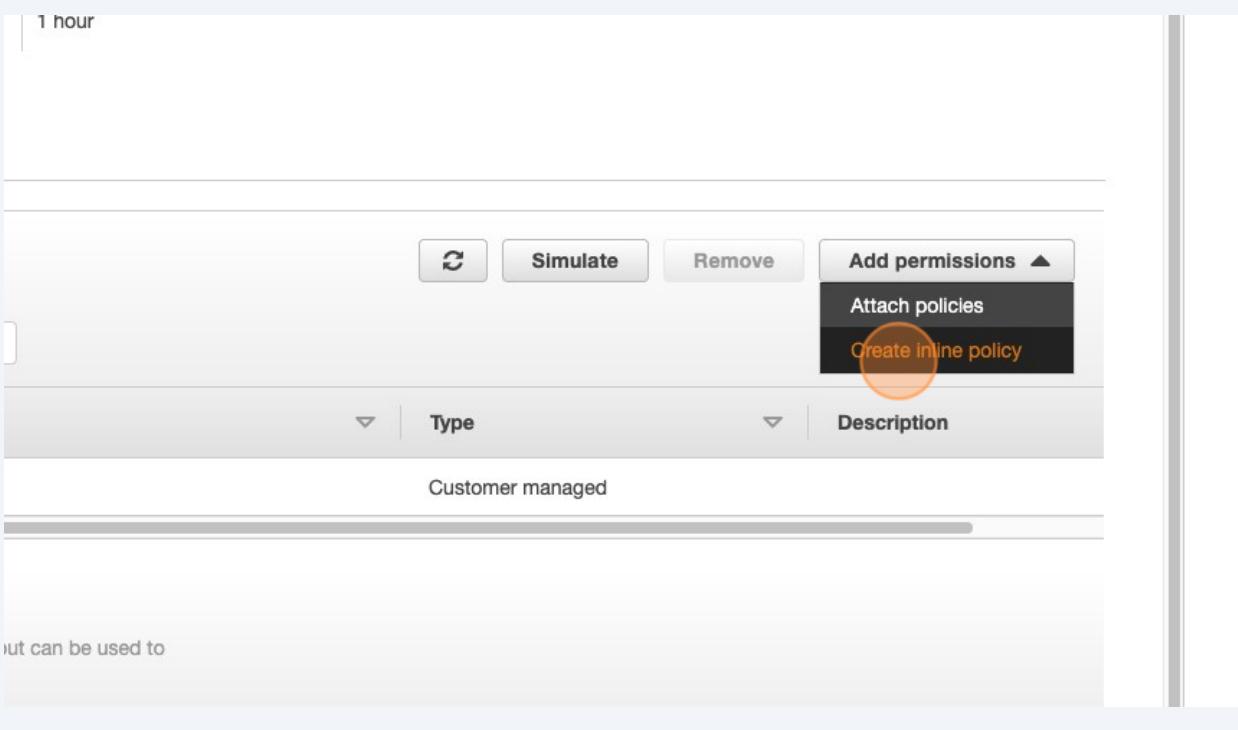
Maximum session duration
1 hour

Add permissions ▾

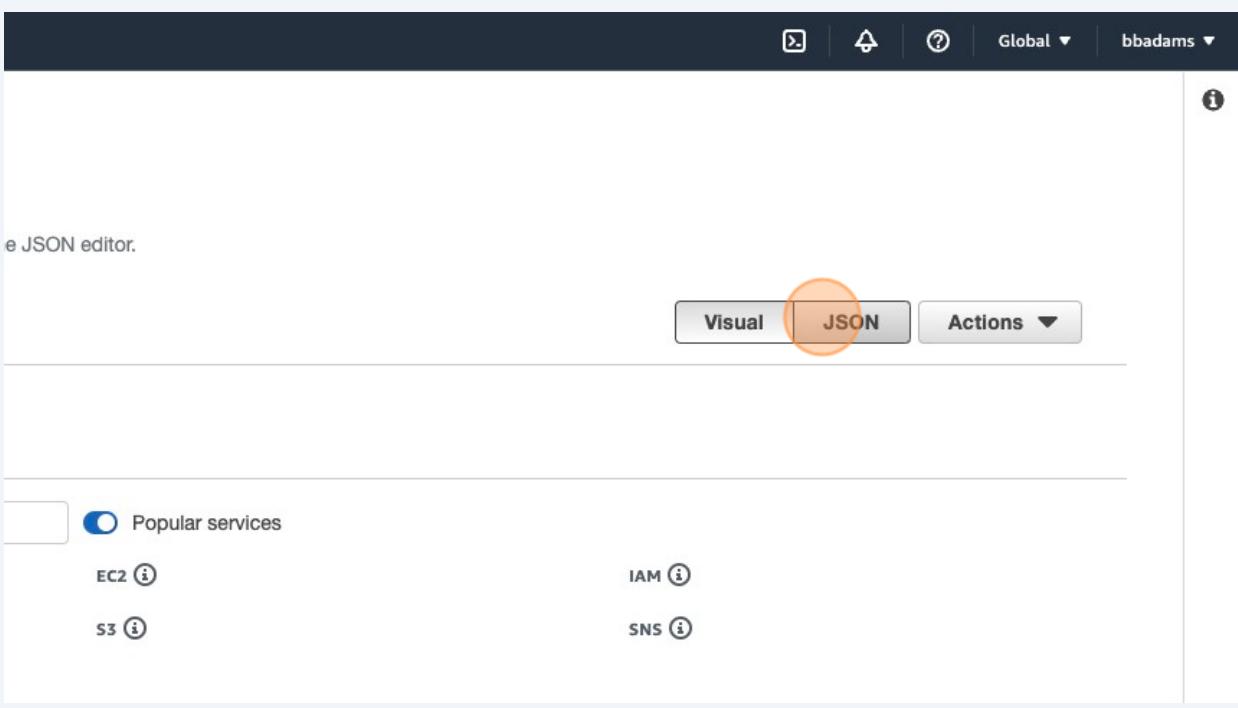
< 1 > |

| Type | Description |
|------------------|-------------|
| Customer managed | |

71 Click "Create inline policy"



72 Click "JSON"



73

Copy and paste the DynamoDB policy information. It will be in the files folder of this project.

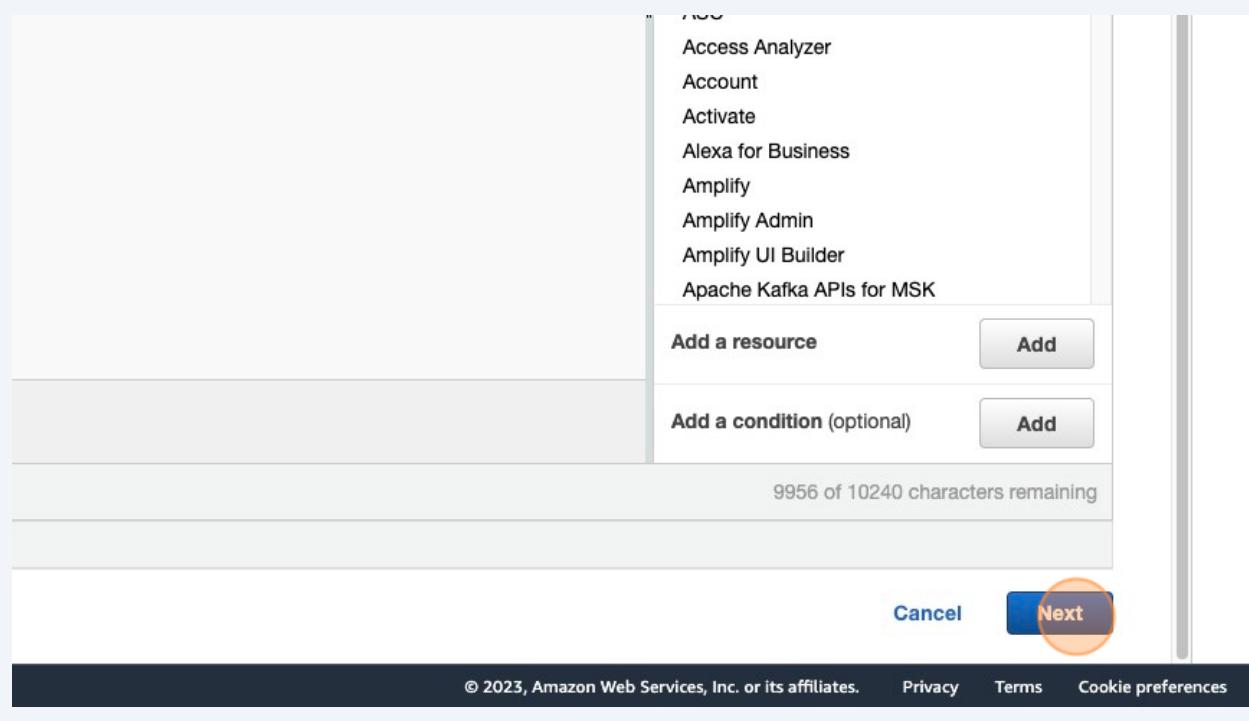
```
3 "Statement": L
4 {
5     "Sid": "VisualEditor0",
6     "Effect": "Allow",
7     "Action": [
8         "dynamodb:PutItem",
9         "dynamodb>DeleteItem",
10        "dynamodb:GetItem",
11        "dynamodb:Scan",
12        "dynamodb:Query",
13        "dynamodb:UpdateItem"
14    ],
15    "Resource":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

74

Paste the ARN link you saved on a notepad earlier here.

```
3 "Statement": L
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": ""
16 }
17 ]
18 }
```

75 Click "Next"



76 Click the "Policy name" field and give the policy a name.

The screenshot shows the 'Review and create' step of the AWS IAM Policy Editor. On the left, there are two tabs: 'Step 1 Specify permissions' and 'Step 2 Review and create'. The 'Review and create' tab is selected. The main area is titled 'Review and create' and contains the instruction 'Review the permissions, specify details, and tags.' Below this is a 'Policy details' section. In the 'Policy name' field, the placeholder 'Enter a meaningful name to identify this policy.' is visible, and the input field is highlighted with a red circle. Below the input field, the text 'Maximum 128 characters. Use alphanumeric and '+,-,@,_' characters.' is displayed. At the bottom of the page, there is a 'Permissions defined in this policy' section with a search bar and a button labeled 'Allow (1 of 384 services)'.

77

permissions

and create

Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

PowerOfMathDynamoPolicy

Maximum 128 characters. Use alphanumeric and '+=-,.@-_` characters.

Permissions defined in this policy Info

Permissions in the policy document specify which actions are allowed or denied.

Search

Allow (1 of 384 services)

78

Click "Create policy"

Request condition

Show remaining 383 services

None

Cancel Previous

79 Click "Code"

+ Add trigger

Code | Test | Monitor | Configuration | Aliases | Versions

General configuration

Triggers

Permissions

Destinations

Function URL

Execution role

Role name
PowerofMath-role-49aq7y91

Resource summary

80 Copy and paste the lambda code FINAL in the files of this project and paste it here.
Save it.

Environment

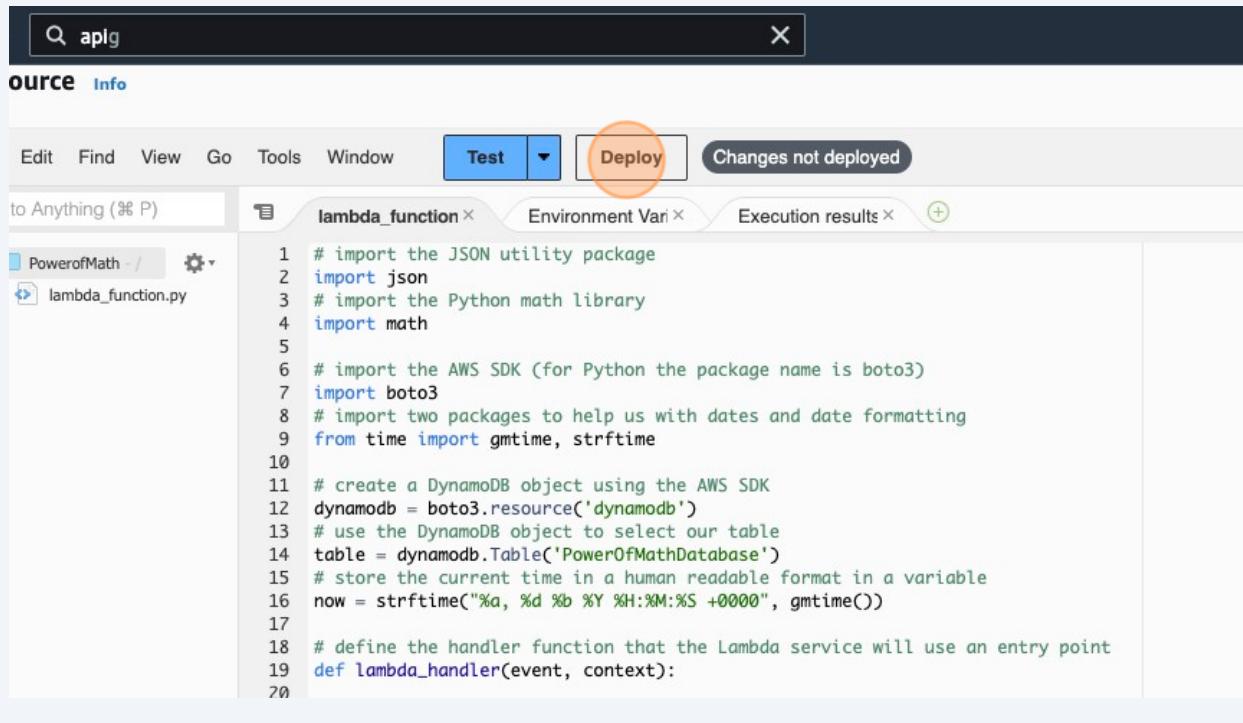
PowerofMath - /

lambda_function.py

```
1 if __name__ == '__main__':
2     import json
3     # import the Python math library
4     import math
5
6     # define the handler function that the Lambda service will use
7     def lambda_handler(event, context):
8
9         # extract the two numbers from the Lambda service's event object
10        mathResult = math.pow(int(event['base']), int(event['exponent']))
11
12        # return a properly formatted JSON object
13        return {
14            'statusCode': 200,
15            'body': json.dumps('Your result is ' + str(mathResult))
16        }
17
```

CloudShell Feedback Language

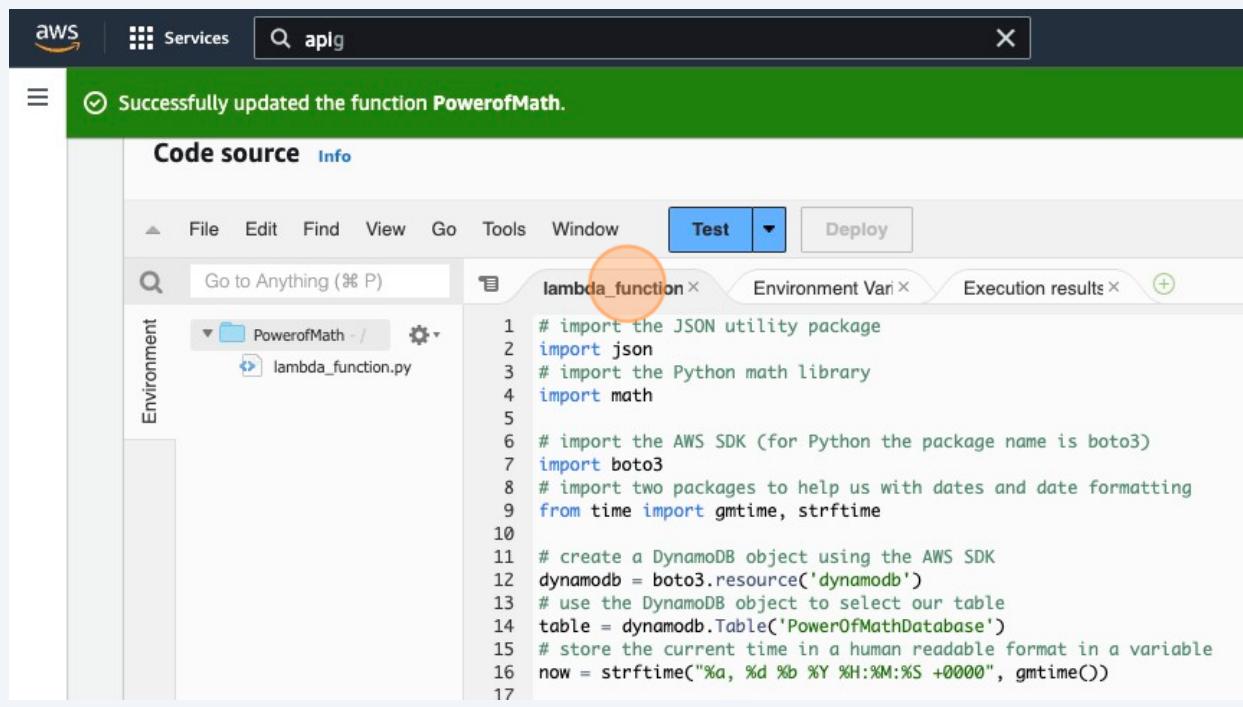
81 Click "Deploy"



The screenshot shows the AWS Lambda function editor interface. At the top, there is a search bar with the text "aplg". Below it, a navigation bar includes "source" and "Info" tabs, followed by "Test" and "Deploy" buttons. The "Deploy" button is highlighted with a red circle. A status message "Changes not deployed" is displayed next to it. The main area shows a file named "lambda_function.py" with the following code:

```
1 # import the JSON utility package
2 import json
3 # import the Python math library
4 import math
5
6 # import the AWS SDK (for Python the package name is boto3)
7 import boto3
8 # import two packages to help us with dates and date formatting
9 from time import gmtime, strftime
10
11 # create a DynamoDB object using the AWS SDK
12 dynamodb = boto3.resource('dynamodb')
13 # use the DynamoDB object to select our table
14 table = dynamodb.Table('PowerOfMathDatabase')
15 # store the current time in a human readable format in a variable
16 now = strftime("%a, %d %b %Y %H:%M:%S +0000", gmtime())
17
18 # define the handler function that the Lambda service will use as an entry point
19 def lambda_handler(event, context):
```

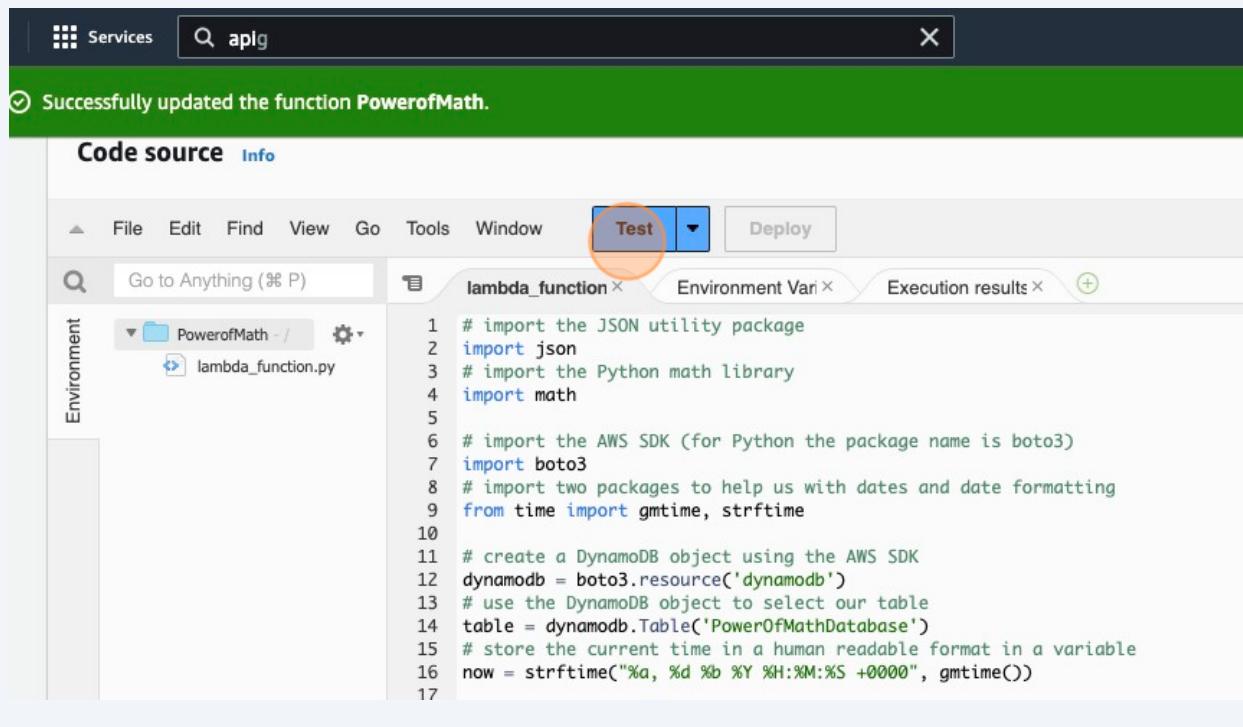
82 Click "lambda_function.py"



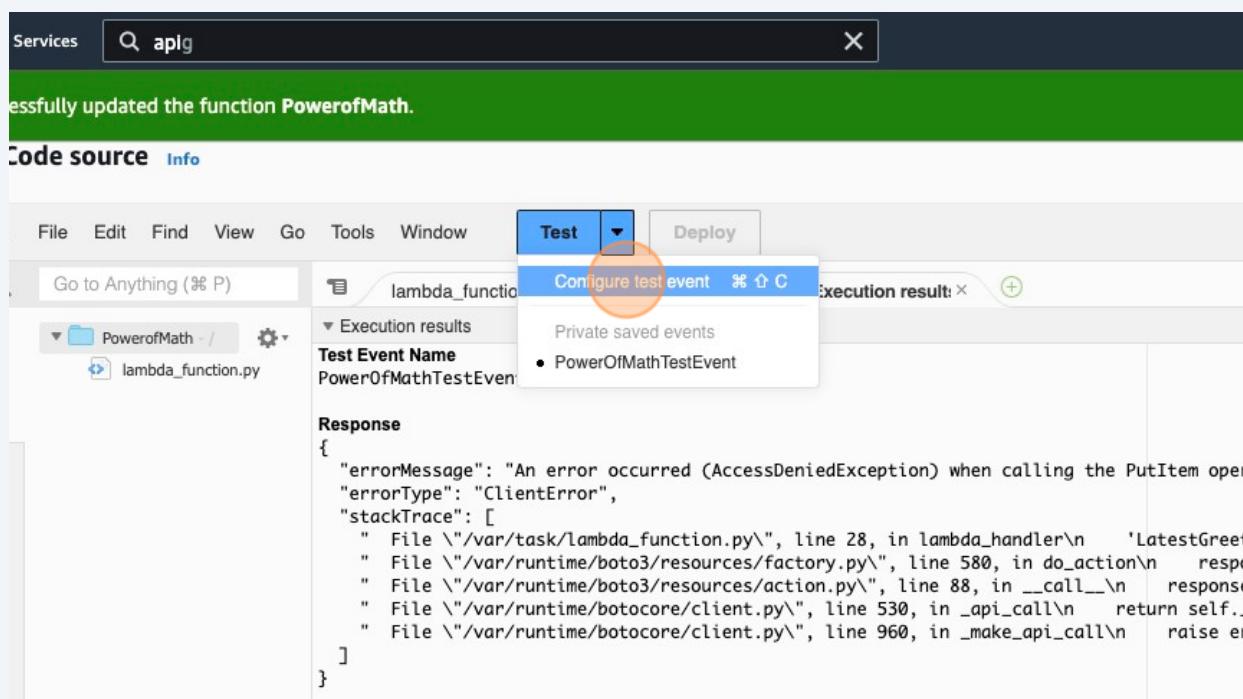
The screenshot shows the AWS Lambda function editor interface. At the top, there is a navigation bar with "aws", "Services", and a search bar with the text "aplg". Below the search bar, a green notification bar says "Successfully updated the function PowerofMath.". The main area shows the "Code source" tab selected. On the left, there is a sidebar with "Environment" and a "Go to Anything" search bar. The main workspace shows a file named "lambda_function.py" with the same code as in the previous screenshot:

```
1 # import the JSON utility package
2 import json
3 # import the Python math library
4 import math
5
6 # import the AWS SDK (for Python the package name is boto3)
7 import boto3
8 # import two packages to help us with dates and date formatting
9 from time import gmtime, strftime
10
11 # create a DynamoDB object using the AWS SDK
12 dynamodb = boto3.resource('dynamodb')
13 # use the DynamoDB object to select our table
14 table = dynamodb.Table('PowerOfMathDatabase')
15 # store the current time in a human readable format in a variable
16 now = strftime("%a, %d %b %Y %H:%M:%S +0000", gmtime())
17
18 # define the handler function that the Lambda service will use as an entry point
19 def lambda_handler(event, context):
```

83 Click "Test"



84 Click the drop down menu of the Test button and select "Configure test event"



85 Click here and type in what numbers you would like an answer to.

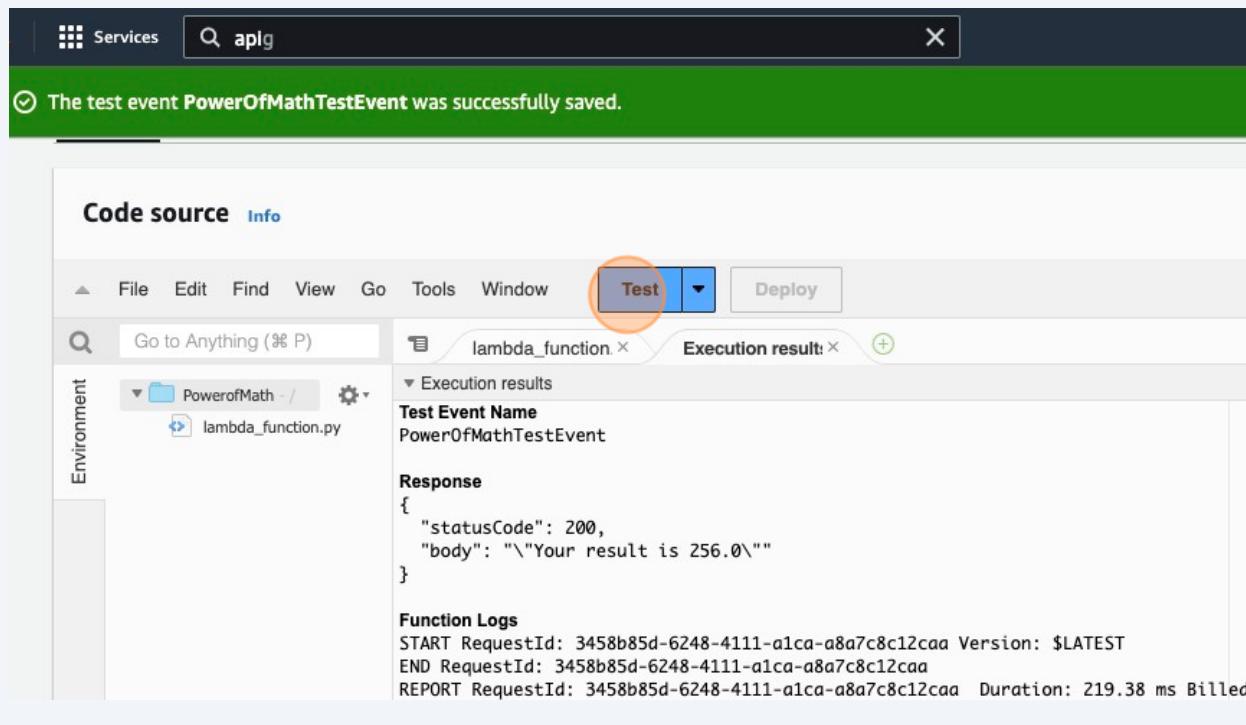
The screenshot shows a Lambda function configuration page. On the left, there is a 'Response' section containing JSON code. On the right, there are two buttons: 'Create new event' (unselected) and 'Edit saved event' (selected). Below these buttons is an 'Event name' field containing 'PowerOfMathTestEvent'. Underneath the event name is a section titled 'Event JSON' containing the following JSON code:

```
1 - {  
2   "base": 4,  
3   "exponent": 16  
4 }  
5 }
```

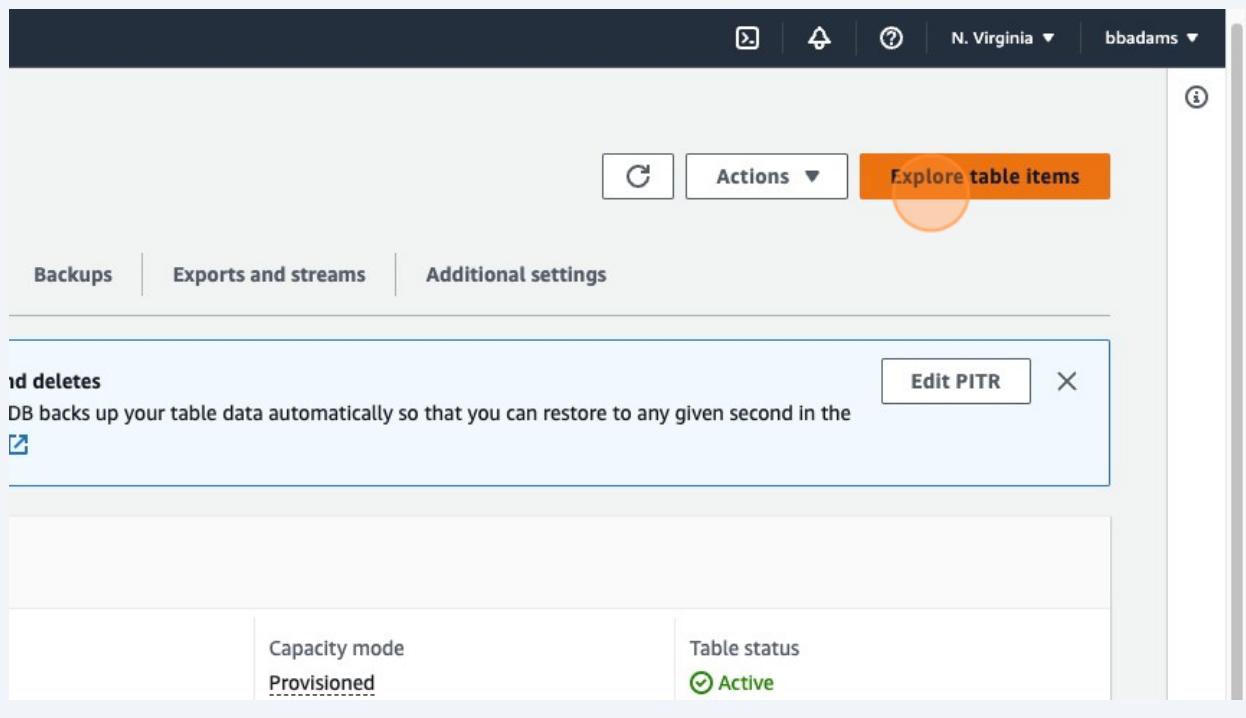
The value '4' in the 'base' field is highlighted with a red circle.

86 Click save.

- 87 Click "Test" and see your results below.



- 88 Click "Explore table items"



89

See answer there as well with a timestamp. All of this is from the new code you pasted and saved.

Completed. Read capacity units consumed: 0.5

Items returned (1)

| | ID | LatestGreetingTime |
|--------------------------|-------|--|
| <input type="checkbox"/> | 256.0 | <input type="button"/> Thu, 27 Jul 2023 04:14:40 +0000 |

90

Go back to amplify and upload the "index.html" file. (Do not choose the html file with the word "ORIGINAL" in it.) Click on the domain click to open in a new tab. See the html page.

and backend environments.

ironments

branch to view build details.

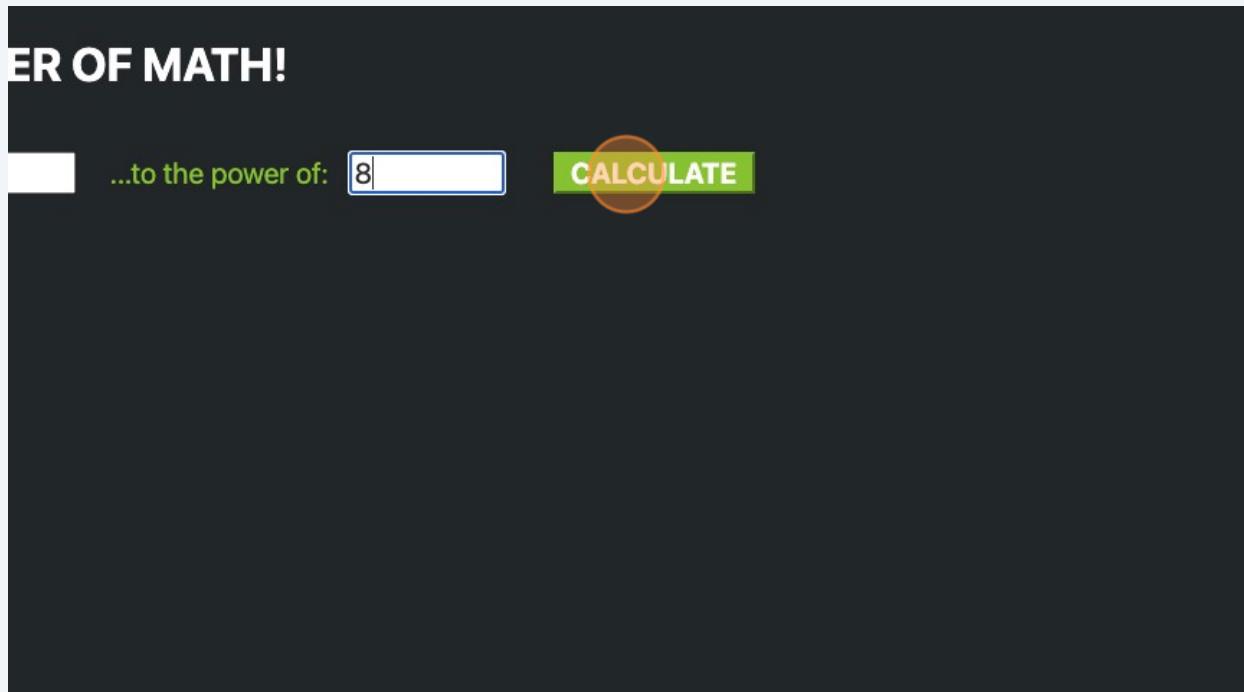
Deployment successfully completed.

Domain
<https://dev.de009xovmtbp0.amplifyapp.com>

Last deployment
7/26/2023, 11:22:20 PM

91

Type in the two numbers you want to do math with and click "CALCULATE" and see your results in a popup window.

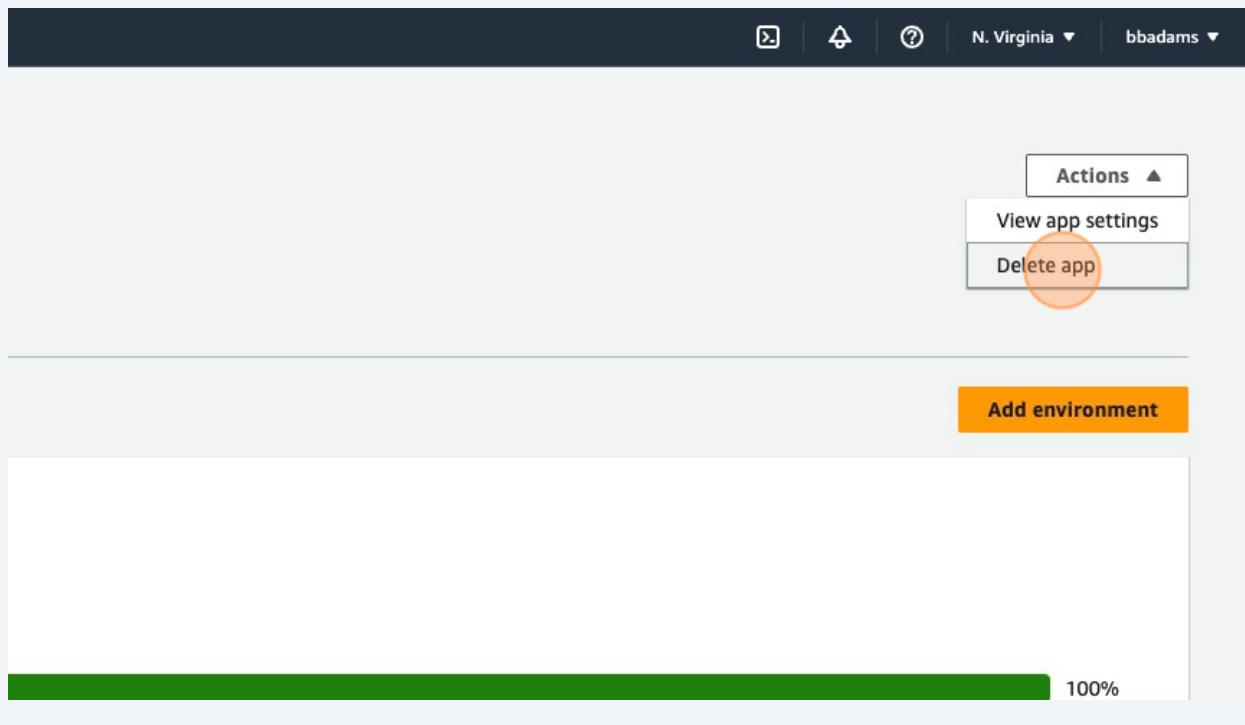


92

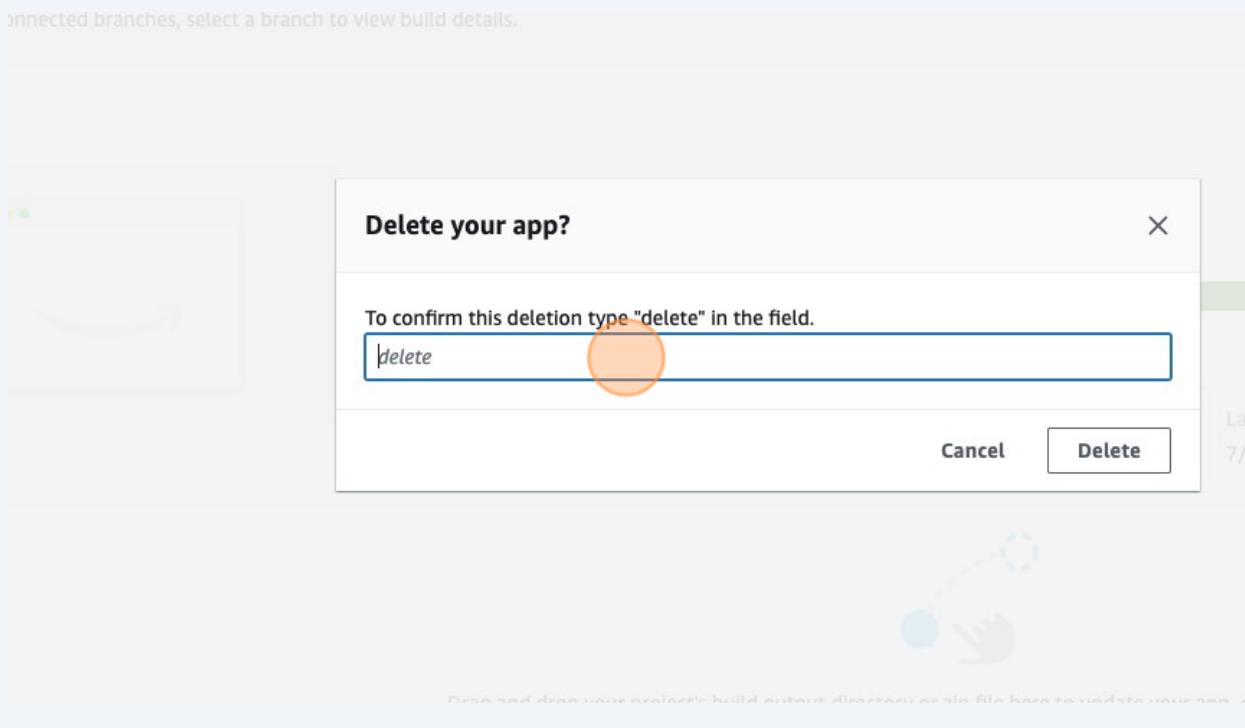
CLEAN UP YOUR RESOURCES. Delete your amplify app.

A screenshot of the Amplify console. At the top, there is a dark header bar with several icons: a square, a triangle, a question mark, and user information for "N. Virginia" and "bbadams". Below the header is a large, empty white area. In the top right corner of this area, there is a small orange button labeled "Actions" with a dropdown arrow. Further down, there is an orange button labeled "Add environment". At the very bottom of the page, there is a green progress bar indicating "100%".

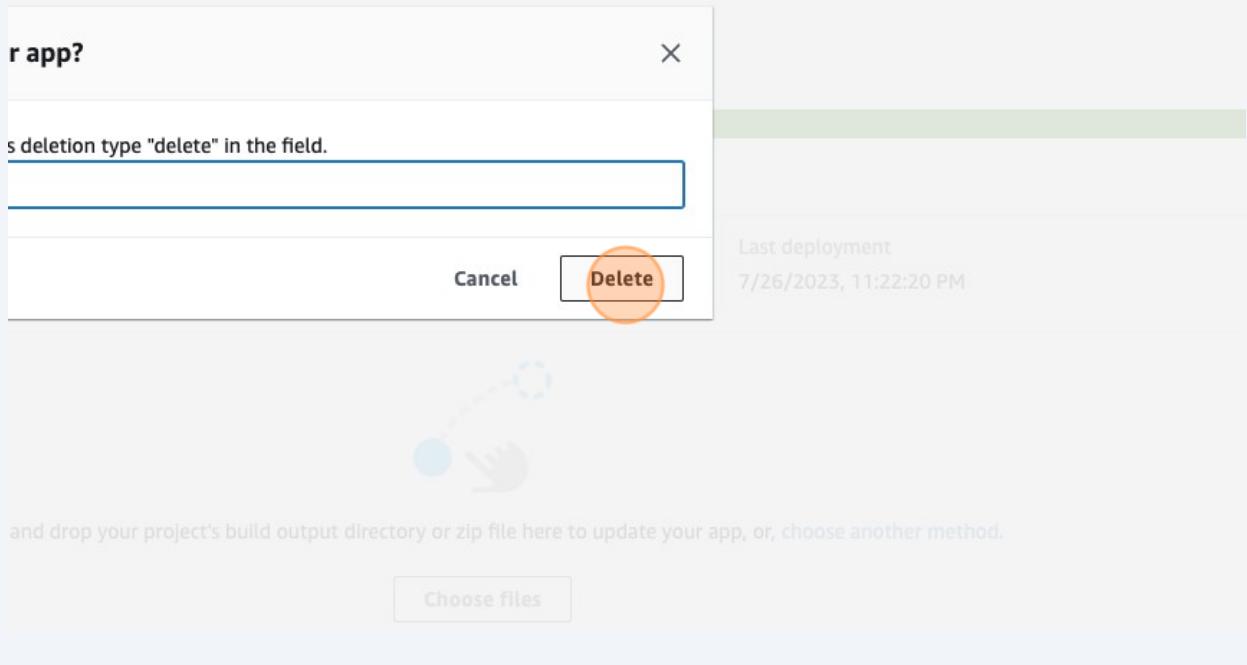
93 Click "Delete app"



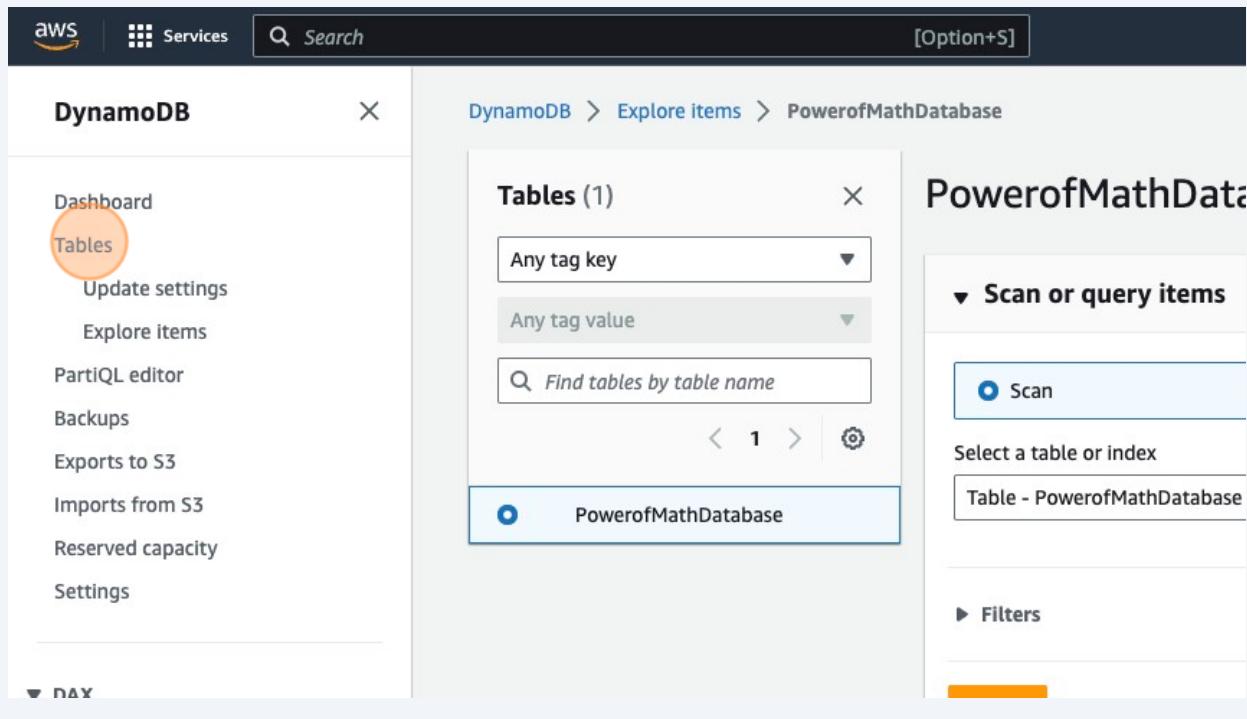
94 Type in "delete" in the field.



95 Click "Delete"



96 Click "Tables" of your DynamoDB.



97 Click this checkbox.

The screenshot shows the AWS DynamoDB console. On the left, there's a sidebar with options like Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, Settings, and DAX. The 'Tables' option is selected. The main area shows a table titled 'Tables (1)'. The table has columns: Name, Status, Partition key, and Sort key. There is one row for 'PowerofMathDatabase', which is marked as 'Active'. The checkbox in the first column of this row is highlighted with a red circle.

98 Click "Delete"

The screenshot shows the AWS DynamoDB table configuration page for a table named 'PowerofMathDatabase'. At the top, there are buttons for Actions, Delete (which is highlighted with a red circle), and Create table. Below this, there are dropdown menus for Key and Any tag value, and navigation controls (back, forward, search). The table details section includes fields for Provisioned with auto scaling (5), Write capacity mode, Total size (0 bytes), and Table class (Standard).

99 Type "confirm"

100 Click "Delete"

An on-demand backup of your table for long-term retention and data archiving.
Use this backup to restore your data to its exact state before table deletion.
These apply for on-demand backup and restore. For more information see [Pricing](#).

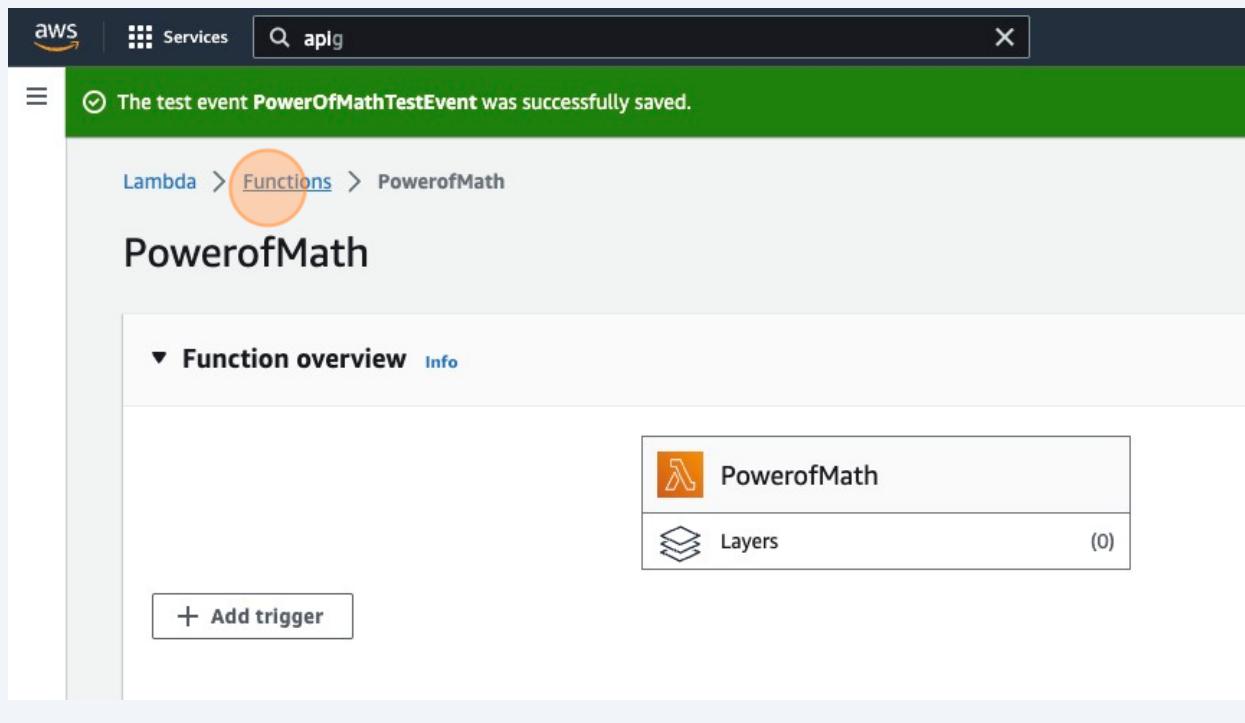
For additional deletions, we ask you to provide additional confirmation.

To complete this deletion, type "confirm".

Cancel Delete

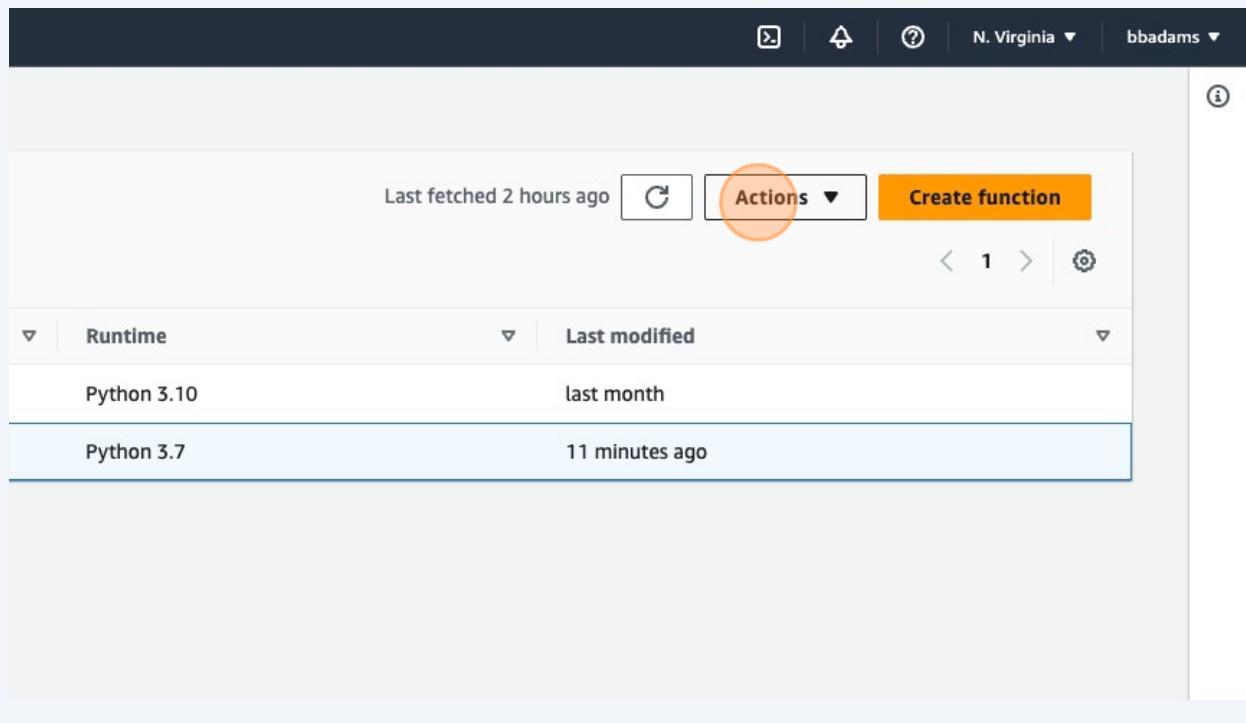
© 2023, Amazon Web Services, Inc. or its affiliates.

101 Click "Functions"

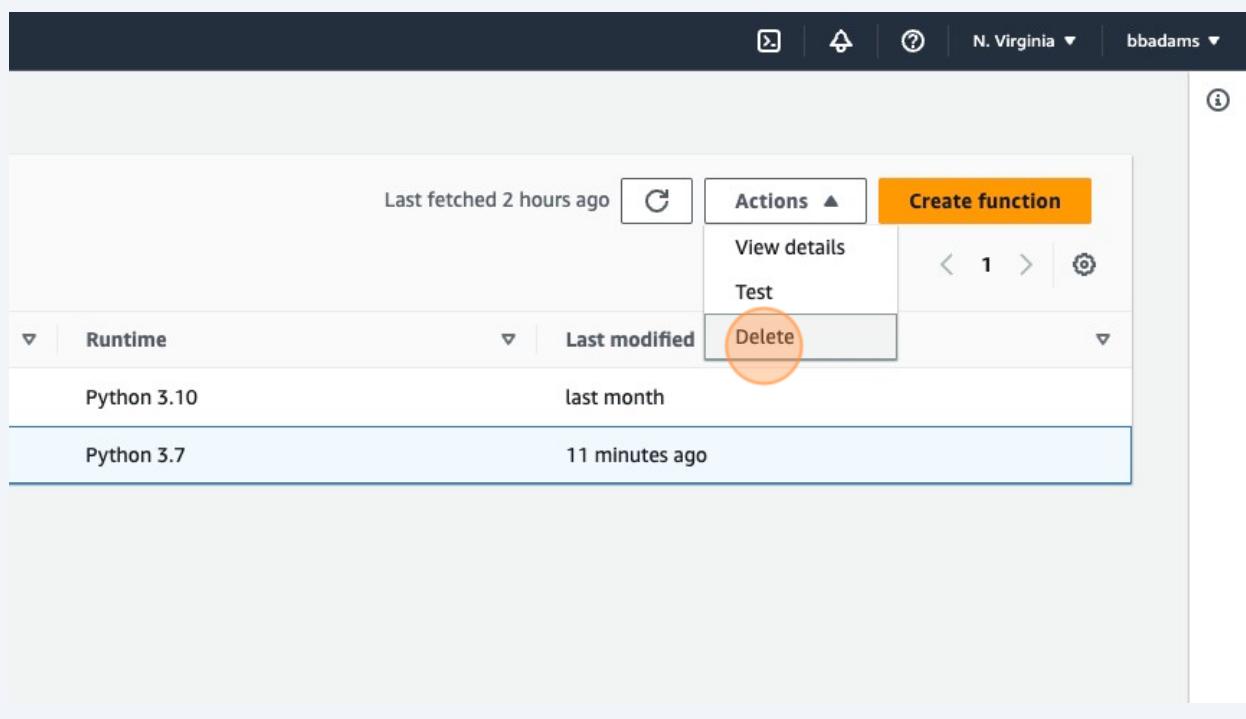


102 Click "PowerofMath"

103 Click "Actions"

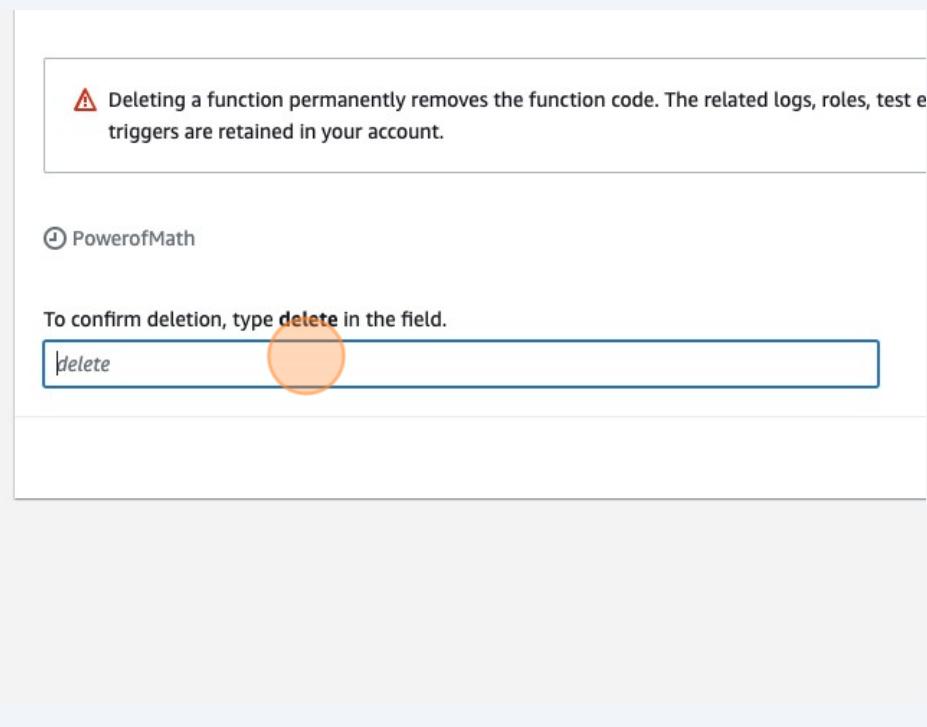


104 Click "Delete"



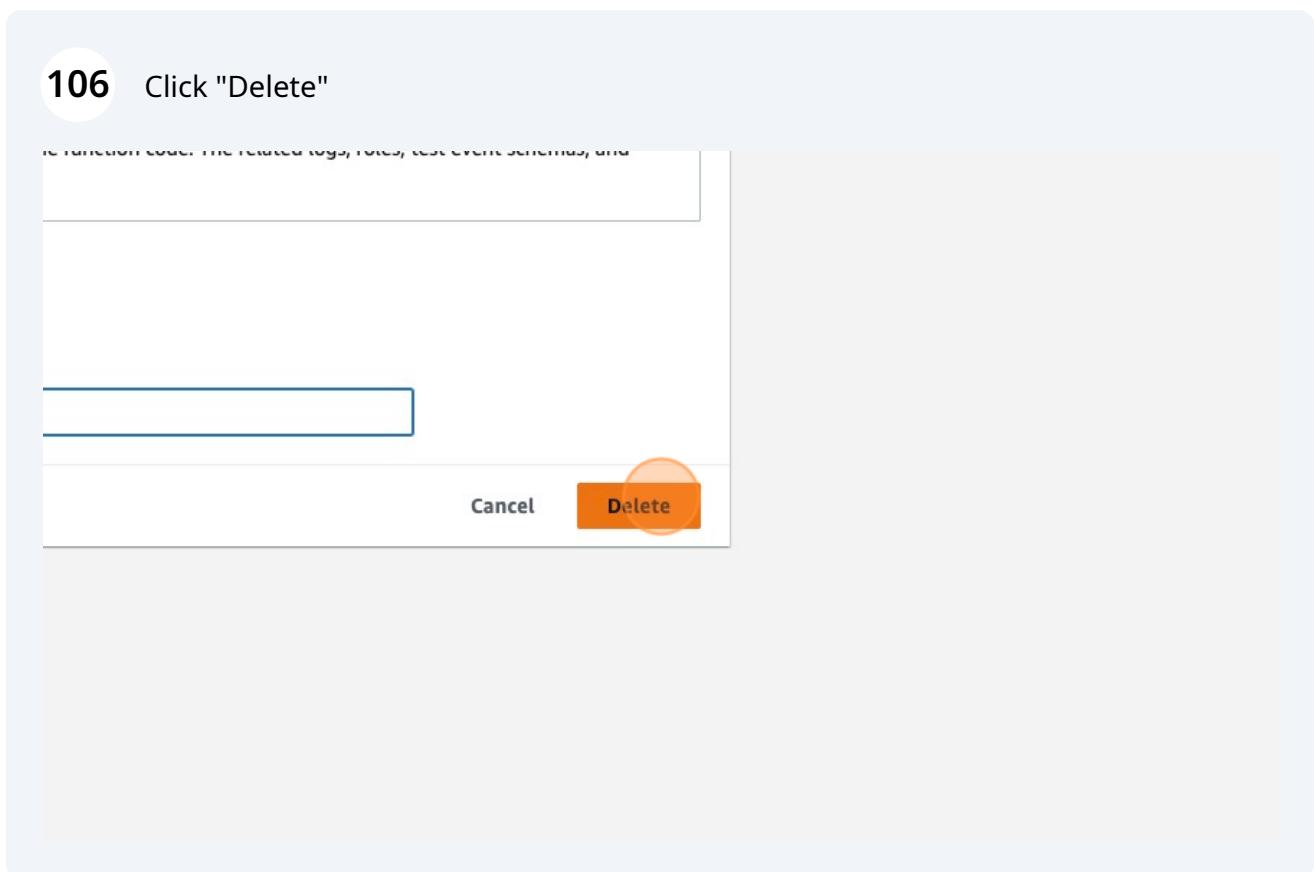
105

Click the "To confirm deletion, type delete in the field." field and type delete in the box.

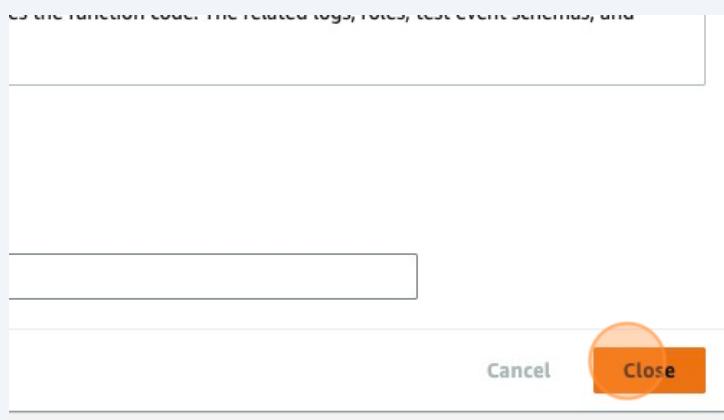


106

Click "Delete"



107 Click "Close"



108 Click "APIs"

A screenshot of the AWS API Gateway console. The top navigation bar shows 'aws', 'Services' (with 'dynamoDB' selected), and a search bar. Below the navigation, the 'APIs' tab is highlighted with a red circle. On the left sidebar, under 'API: PowerofMathAPI', the 'Resources' option is selected. In the main content area, the path '/OPTIONS' is shown, followed by a blue box containing the word 'POST'. To the right, there are sections for 'Method Execution', 'Path', 'Query Strings', and 'Headers', each with descriptive text. A blue arrow points from the 'Method Execution' section back to the 'POST' method in the path list.

109 Click this radio button.

The screenshot shows the AWS API Gateway console. The left sidebar has 'APIs' selected. The main area displays a table titled 'APIs (1)'. A single row is present, labeled 'PowerofMathAPI'. The first column contains a radio button icon, which is highlighted with an orange circle, indicating it is the target for the click action. The table has columns for 'Name' and 'Description'.

110 Click "Actions"

The screenshot shows the AWS API Gateway console. At the top, there is a toolbar with various icons and dropdown menus. The 'Actions' dropdown menu is highlighted with an orange circle, indicating it is the target for the click action. Below the toolbar, there is a table with columns: 'Protocol', 'Endpoint type', and 'Created'. The table shows one row with 'REST' in the Protocol column, 'Regional' in the Endpoint type column, and '2023-07-27' in the Created column.

111 Click "Delete"

A screenshot of a web-based API management tool. At the top, there are navigation icons and dropdown menus for 'N. Virginia' and 'bbadams'. Below the header, a table displays a single API entry. The columns are 'Protocol' (REST), 'Endpoint type' (Regional), and 'Created' (2023-07-27). Above the table, a toolbar includes a 'Create API' button and an 'Actions' dropdown menu. The 'Delete' option in the Actions menu is highlighted with a red circle.

| Protocol | Endpoint type | Created |
|----------|---------------|------------|
| REST | Regional | 2023-07-27 |

112 Click "Delete"

