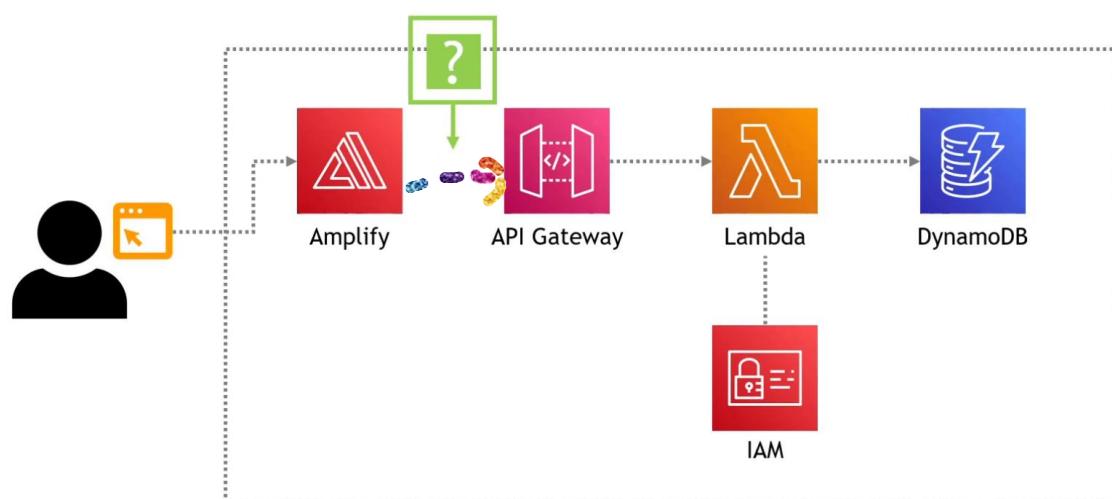


Project title: Building end-to-end web applications using AWS Serverless Architecture

Project statement: Using AWS serverless Architecture building end to end web applications.

Services Required: AWS Amplify, Amazon API Gateway, AWS Lambda, Amazon DynamoDB & AWS IAM

The Current Architecture



AWS Amplify: Amplify is everything you need to build full-stack web and mobile apps on AWS. Build and host your frontend, add features like auth and storage, connect to real-time data sources, deploy, and scale to millions of users.

Amazon API Gateway : Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API

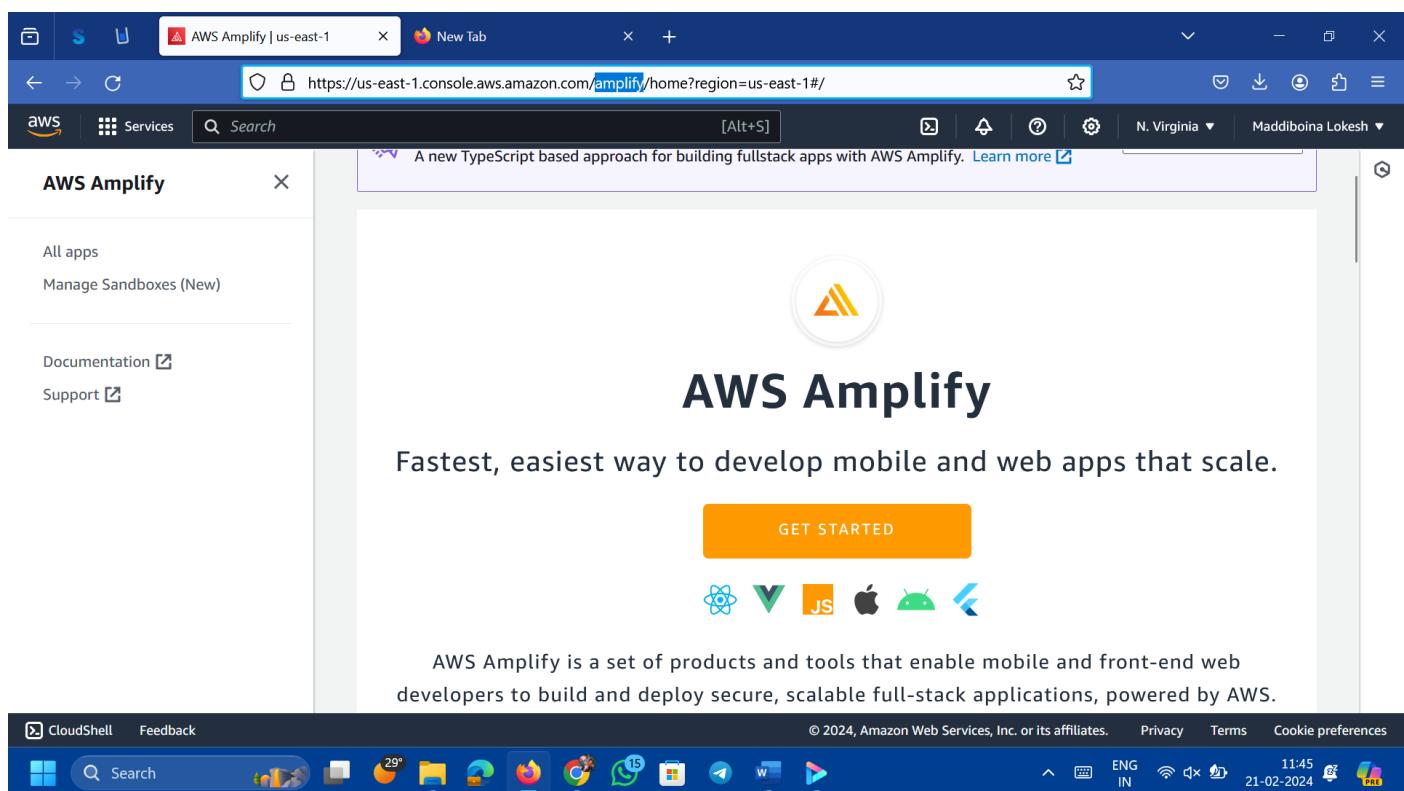
Gateway supports containerized and serverless workloads, as well as web applications.

AWS Lambda: AWS Lambda is a compute service that runs your code in response to events and automatically manages the compute resources, making it the fastest way to turn an idea into a modern, production, serverless applications.

Amazon DynamoDB: Amazon DynamoDB is a serverless, NoSQL database service that enables you to develop modern applications at any scale.

AWS IAM: Use IAM to manage and scale workload and workforce access securely supporting your agility and innovation in AWS.

Get start with the Amplify



AWS Amplify | us-east-1 New Tab https://us-east-1.console.aws.amazon.com/amplify/home?region=us-east-1#/create AWS Services Search [Alt+S] N. Virginia Maddiboina Lokesh

AWS Amplify All apps Manage Sandboxes (New) Documentation Support

Amplify Hosting is a fully managed hosting service for web apps. Connect your repository to build, deploy, and host your web app.

From your existing code

Connect your source code from a Git repository or upload files to host a web app in minutes.

GitHub 

Bitbucket 

GitLab 

AWS CodeCommit 

Deploy without Git provider 

Continue

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 11:49 21-02-2024

Select the Deploy without Git Provider because we uploading Zip file from our local drive

AWS Amplify | us-east-1 New Tab https://us-east-1.console.aws.amazon.com/amplify/home?region=us-east-1#/create AWS Services Search [Alt+S] N. Virginia Maddiboina Lokesh

AWS Amplify All apps Manage Sandboxes (New) Documentation Support

Amplify Hosting is a fully managed hosting service for web apps. Connect your repository to build, deploy, and host your web app.

From your existing code

Connect your source code from a Git repository or upload files to host a web app in minutes.

GitHub 

Bitbucket 

GitLab 

AWS CodeCommit 

Deploy without Git provider 

Continue

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 11:50 21-02-2024

AWS Amplify | us-east-1

Start a manual deployment

App name
Give this app a name or we will generate a default for you
PowerOfMath2

Environment name
Give this resource a meaningful environment name, like dev, test, or prod, or we will generate a default for you
dev

Method

Drag and drop

Amazon S3

Any URL

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Let

App name is PowerOfMath

Environment name is dev

AWS Amplify | us-east-1

Method

Drag and drop

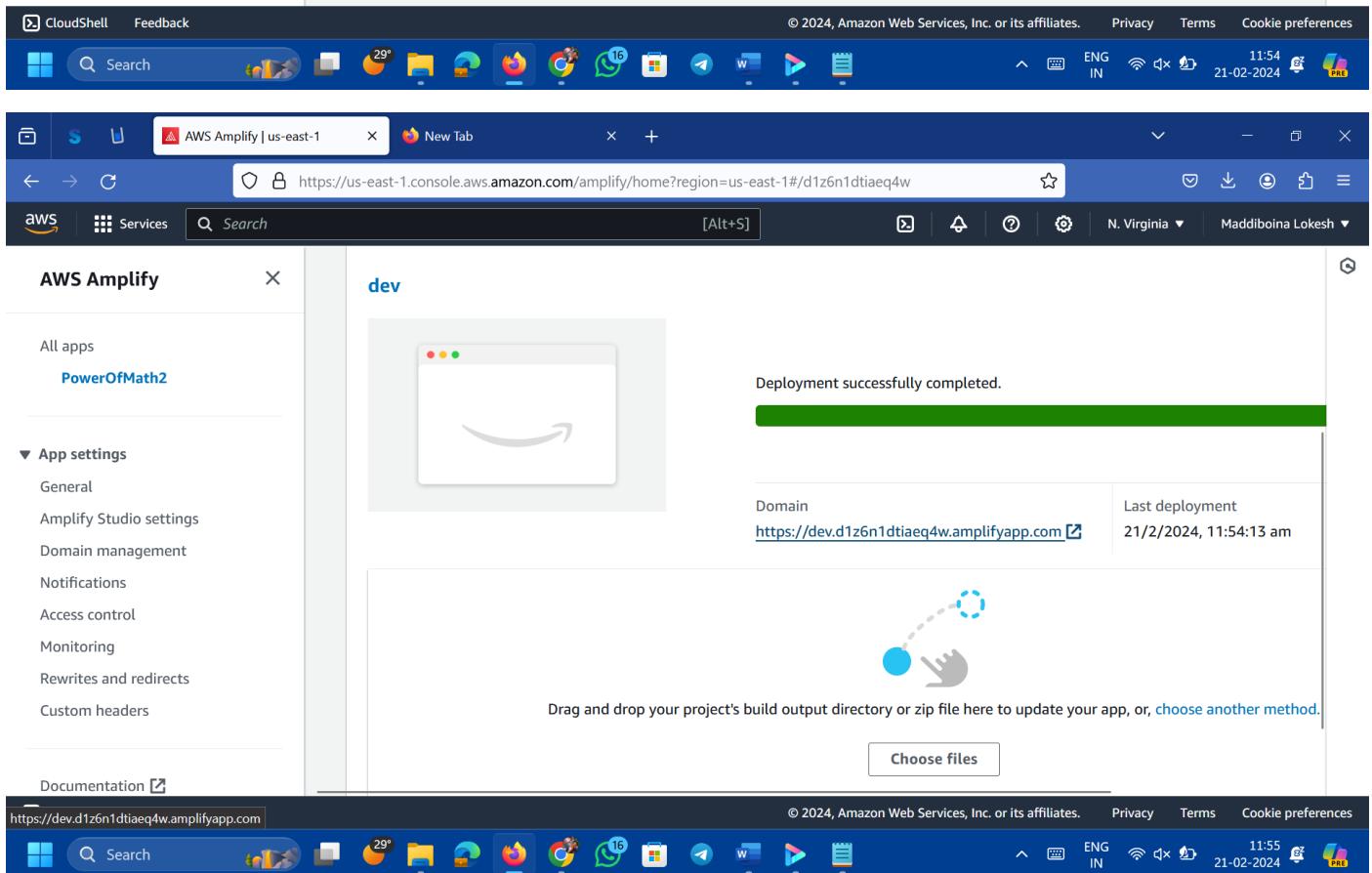
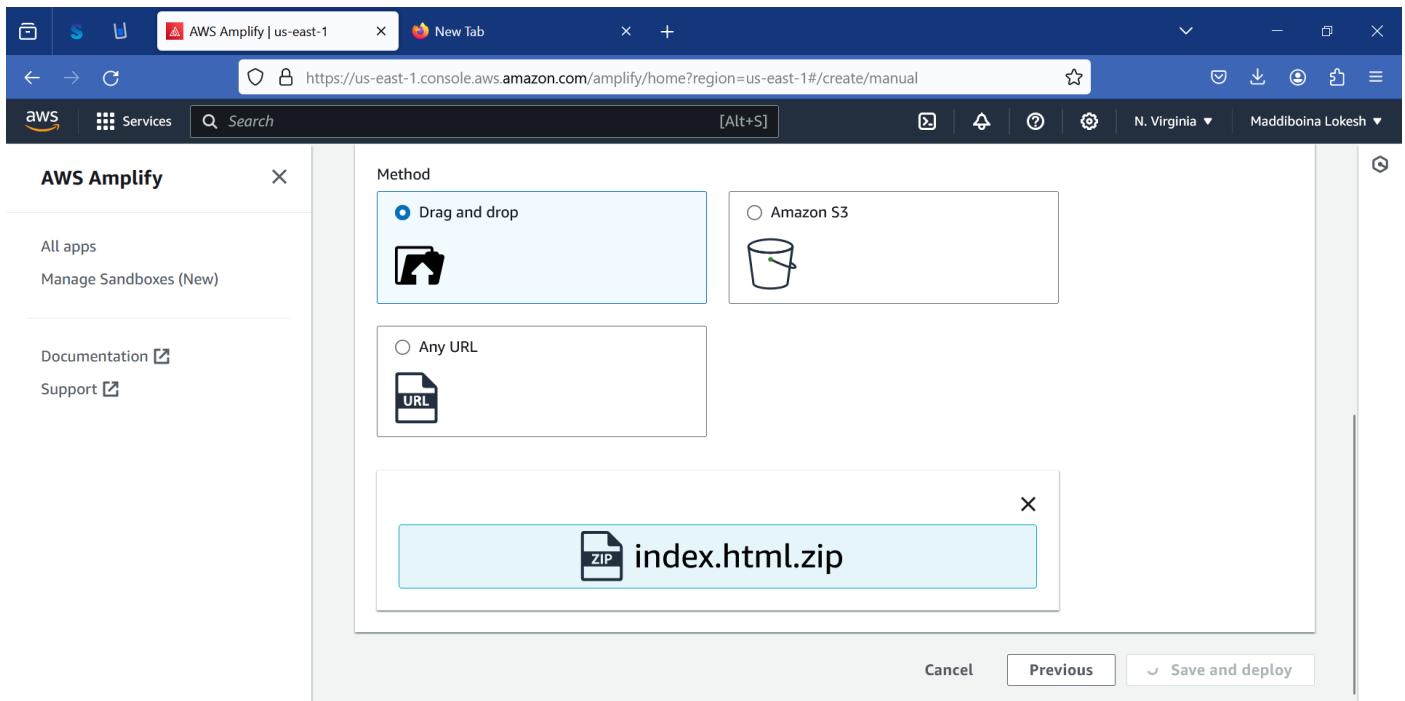
Amazon S3

Any URL

index.html.zip

Cancel Previous Save and deploy

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Now let create the Lambda function

The screenshot shows the AWS Lambda 'Create function - Lambda' wizard. In the 'Basic information' step, the function name is set to 'PowerOfMathFunction'. The runtime is chosen as 'Python 3.9' and the architecture as 'x86_64'. A 'Tutorials' sidebar on the right is open, showing a 'Create a simple web app' tutorial with steps to build a Lambda function that outputs a webpage.

Function name is PowerOfMathFunction

Runtime is Python 3.9

This is code for above function this is not finally code .it last we will change code

```
# import the JSON utility package
```

```
import json
```

```
# import the Python math library
```

```
import math
```

```
# define the handler function that the Lambda service will use as 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))
}

```

Successfully created the function **PowerOfMathFunction**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Code source [Info](#)

[Upload from](#)

[Test](#) [Deploy](#) Changes not deployed

lambda_function Environment Vari [+](#)

```

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    }

```

[CloudShell](#) [Feedback](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

ENG IN 16:00 21-02-2024

Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#) [Start tutorial](#)

The screenshot shows the AWS Lambda console interface. At the top, there are tabs for 'Info' and 'Tutorials'. A green banner at the top of the main content area says 'Successfully updated the function PowerOfMathFunction.' Below this, the 'Code source' tab is selected, showing the function's code in a code editor. The code is as follows:

```
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    }
```

The AWS Lambda logo is visible in the top left corner of the browser window.

The screenshot shows the 'Configure test event' dialog box overlaid on the AWS Lambda console. The dialog has the following fields:

- Test event action:** A radio button group with 'Create new event' selected and 'Edit saved event' as an option.
- Event name:** A text input field containing 'PowerOfMathtest'.
- Event sharing settings:** A radio button group with 'Private' selected and 'Shareable' as an option. A note below states: 'This event is only available in the Lambda console and to the event creator. You can configure a total of 10.' followed by a 'Learn more' link.

At the bottom right of the dialog are 'Cancel', 'Invoke', and 'Save' buttons. The AWS Lambda logo is visible in the top left corner of the browser window.

```
1 + {  
2     "base": 2,  
3     "exponent": 3  
4 }  
Format JSON  
Cancel Invoke Save
```

We are writing the JSON code to test the function

The test event PowerOfMathtest was successfully saved.

Code source Info

Execution result: Succeeded | Max memory used: 36 MB | Time: 1.65 ms

Test Event Name: PowerOfMathtest

Response:

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

Function Logs:

```
START RequestId: 0f32d6cc-711c-40b5-94e4-b5a754cf032 Version: $LATEST  
END RequestId: 0f32d6cc-711c-40b5-94e4-b5a754cf032  
REPORT RequestId: 0f32d6cc-711c-40b5-94e4-b5a754cf032 Duration: 1.65 ms Billed Duration: 1.65 ms
```

Request ID: 0f32d6cc-711c-40b5-94e4-b5a754cf032

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial

Now we are creating Gateway

The screenshot shows the AWS API Gateway - Create API interface. At the top, there are tabs for "AWS Amplify | us-east-1", "PowerOfMathFunction - Lambd", and "API Gateway - Create API". The main content area is titled "REST API" with the sub-section "REST API Private". A yellow circle highlights the "Build" button in the toolbar. Below the sections, there are "Import" and "Build" buttons. The status bar at the bottom indicates "CloudShell Feedback" and shows various system icons.

The screenshot shows the "Create REST API" page. The title bar includes tabs for "AWS Amplify | us-east-1", "PowerOfMathFunction - Lambd", and "API Gateway - Create REST API". The main content area is titled "API details". It features four options: "New API" (selected), "Clone existing API", "Import API", and "Example API". The "New API" section contains fields for "API name" (set to "My REST API") and "Description - optional". Below these, the "API endpoint type" section notes that regional APIs are deployed in the current AWS Region and edge-optimized APIs route requests to the nearest CloudFront Point of Presence. The status bar at the bottom indicates "CloudShell Feedback" and shows various system icons.

The screenshot shows the AWS API Gateway Resources page for a newly created REST API named 'PowerOfMath (rhh115i1pj)'. A green banner at the top indicates 'Successfully created REST API 'PowerOfMath (rhh115i1pj)''. The main area displays 'Resource details' for the root resource ('/'). The 'Methods' section shows 'No methods'. The left sidebar lists various API settings like Stages, Authorizers, and Models.

This screenshot is identical to the one above, but the 'Create method' button in the 'Methods' section is highlighted with a yellow box, indicating the next step in the process.

Screenshot of the AWS API Gateway - Create method page.

Method details

Method type: POST

Integration type:

- Lambda function**: Integrate your API with a Lambda function.
Icon: Lambda symbol.
- HTTP**: Integrate with an existing HTTP endpoint.
Icon: HTTP symbol.
- Mock**: Generate a response based on API Gateway mappings and transformations.
Icon: Mock symbol.

- AWS service**: Integrate with an AWS Service.
Icon: AWS service icon.
- VPC link**: Integrate with a resource that isn't accessible over the public internet.
Icon: VPC link icon.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 16:14 21-02-2024 PRE

Screenshot of the AWS API Gateway - Create method page, showing Lambda proxy integration configuration.

Lambda proxy integration

Send the request to: `arn:aws:lambda:us-east-1:975050273614:function:PowerOfMathFunction`

Lambda function

Provide the Lambda function ARN:
`arn:aws:lambda:us-east-1:975050273614:function:PowerOfMathFunction`

Region: us-east-1

Search bar: `arn:aws:lambda:us-east-1:975050273614:function:PowerOfMathFunction`

Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.

Default timeout
The default timeout is 29 seconds.

Cancel **Create method**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 16:15 21-02-2024 PRE

S U AWS Amplify | us-east-1 X PowerOfMathFunction - Lambd X API Gateway - Resources X +

https://us-east-1.console.aws.amazon.com/apigateway/main/apis/rhh115i1pj/resources?api=rhh115i1pj&experience=API

aws Services Search [Alt+S] N. Virginia Maddiboina Lokesh

API Gateway APIs Custom domain names VPC links

API: PowerOfMath Resources Stages Authorizers Gateway responses Models Resource policy Documentation Dashboard API settings

Successfully created method 'POST' in '/'.

API Gateway > APIs > Resources - PowerOfMath (rhh115i1pj)

Resources

Create resource / POST

/ - POST - Method execution

Update documentation Delete

ARN arn:aws:execute-api:us-east-1:975050273614:rhh115i1pj/*/POST/ Resource ID 1q2t9x9bql

Method request → Integration request →

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 16:15 21-02-2024

S U AWS Amplify | us-east-1 X PowerOfMathFunction - Lambd X API Gateway - Resources X +

https://us-east-1.console.aws.amazon.com/apigateway/main/apis/rhh115i1pj/resources?api=rhh115i1pj&experience=API

aws Services Search [Alt+S] N. Virginia Maddiboina Lokesh

API Gateway APIs Custom domain names VPC links

API: PowerOfMath Resources Stages Authorizers Gateway responses Models Resource policy Documentation Dashboard API settings

Enable CORS on a resource using the API Gateway console

API Gateway > APIs > Resources - PowerOfMath (rhh115i1pj)

Resources

Create resource / POST

Resource details Update documentation Enable CORS

Path / Resource ID 1q2t9x9bql

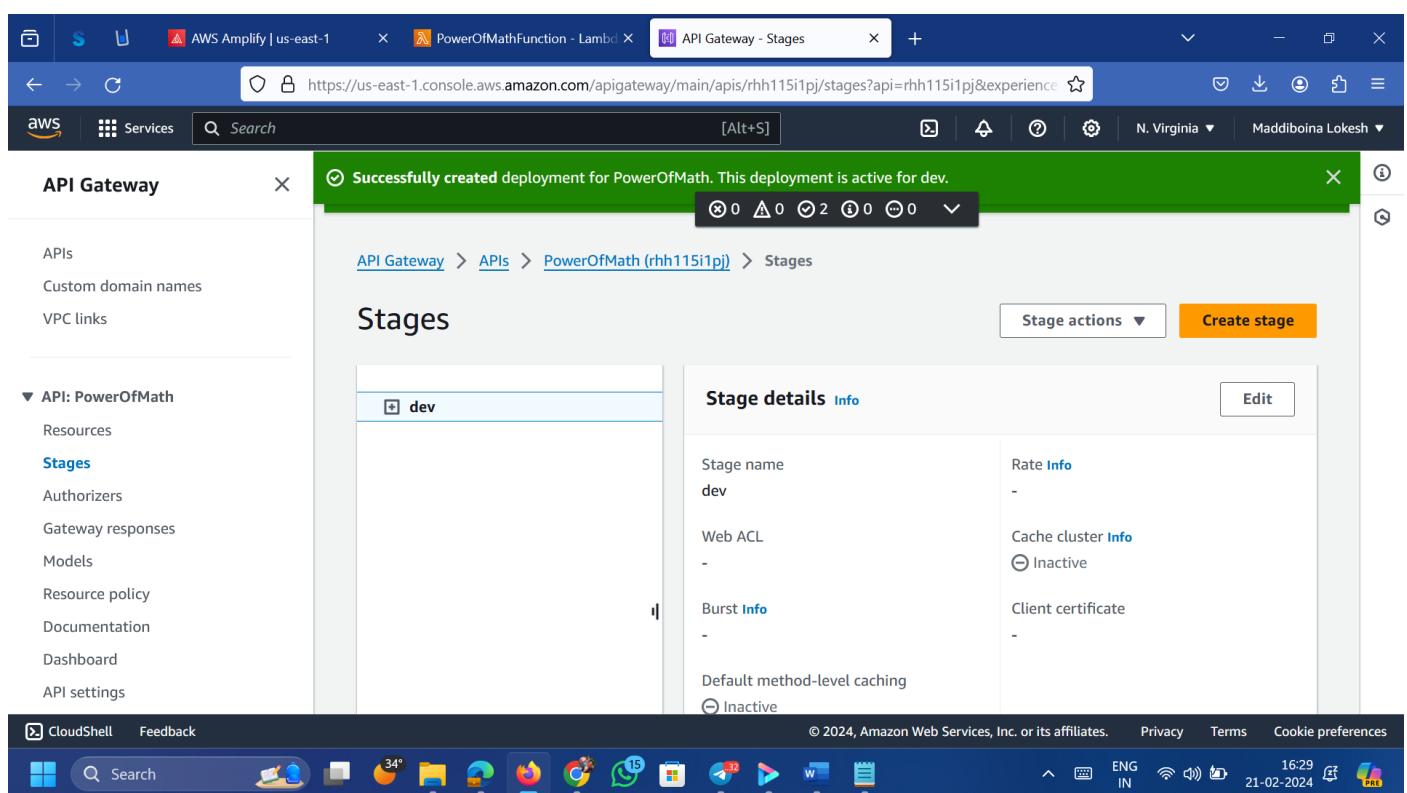
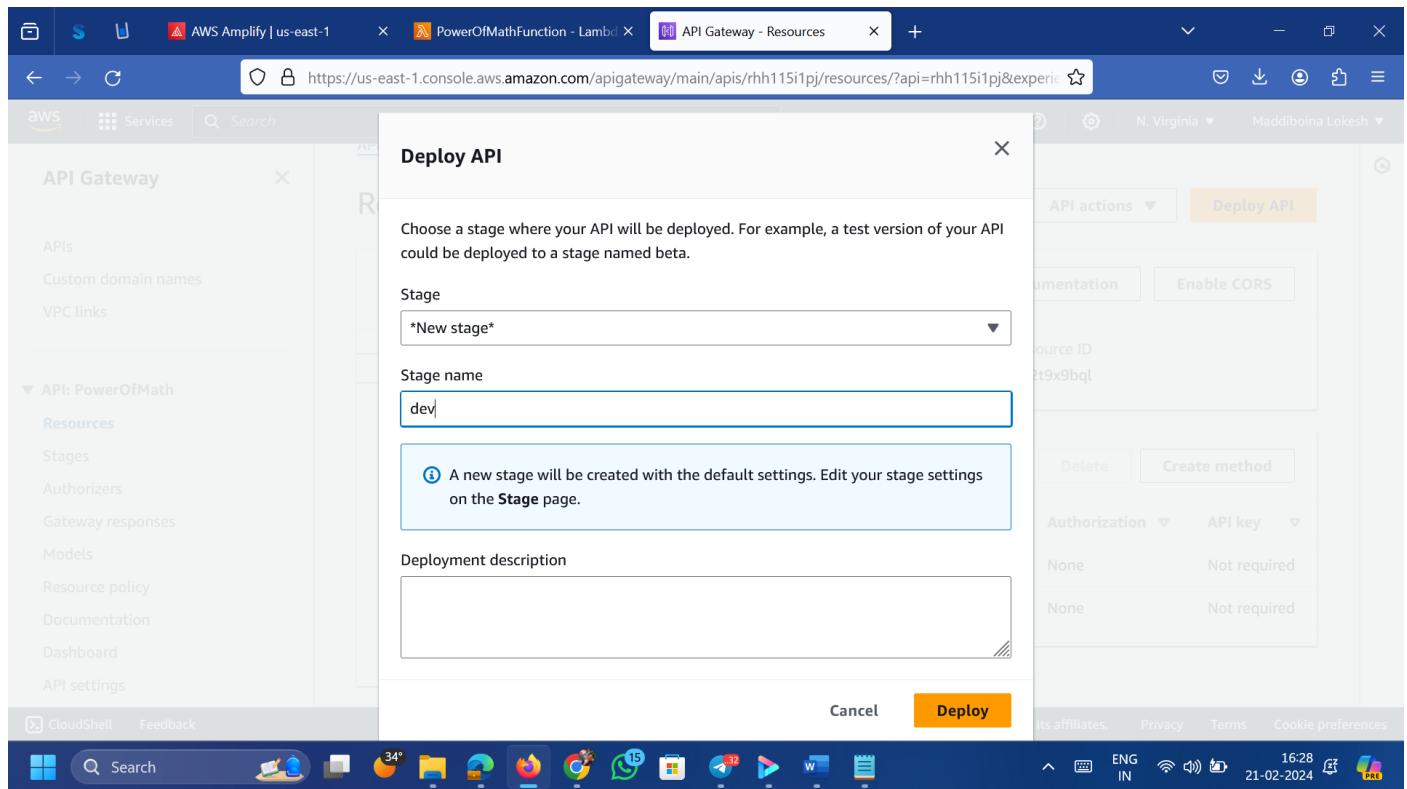
Methods (1)

Method type	Integration type	Authorization	API key
POST	Lambda	None	Not required

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 16:25 21-02-2024

The screenshot shows the AWS API Gateway CORS settings configuration page. It includes sections for 'Gateway responses' (checkboxes for Default 4XX and Default 5XX), 'Access-Control-Allow-Methods' (checkboxes for OPTIONS and POST, with POST selected), 'Access-Control-Allow-Headers' (a text input field containing 'Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token'), and 'Access-Control-Allow-Origin' (a text input field containing '*'). A success message at the top indicates that Access-Control-Allow-Headers, Access-Control-Allow-Methods, and Access-Control-Allow-Origin were added to the OPTIONS method.

The screenshot shows the AWS API Gateway CORS settings configuration page with a success message: "Added Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Integration Response Header Mappings to OPTIONS method." Below this message, the 'Enable CORS' section is visible, which contains the same configuration options as the previous screenshot. The browser's status bar at the bottom shows the date and time as 21-02-2024 16:27.



Screenshot of the AWS API Gateway Stages page for the 'PowerOfMath' API.

The left sidebar shows the API Gateway navigation menu with the 'Stages' option selected. The main content area displays the 'dev' stage configuration.

Burst Info: Default method-level caching (Inactive).

Invoke URL: <https://rhh115i1pj.execute-api.us-east-1.amazonaws.com/dev>

Logs and tracing: CloudWatch logs (Inactive), Detailed metrics (Inactive), X-Ray tracing (Inactive).

Deployment: Active deployment h92jaj on February 21, 2024, 16:29 (UTC+05:30).

CloudShell: Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

<https://rhh115i1pj.execute-api.us-east-1.amazonaws.com/dev>

Screenshot of the AWS API Gateway Resources page for the 'PowerOfMath' API.

The left sidebar shows the API Gateway navigation menu with the 'Resources' option selected. The main content area displays the 'POST/' resource configuration.

Method Request Flow: Client → Method request → Integration request → Integration response → Method response → Client.

Method Response Settings: Test tab selected.

Method request settings:

Authorization	API key required
NONE	False
Request validator	SDK operation name

CloudShell: Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS API Gateway Resources page. On the left, the navigation pane is open with the 'APIs' section selected. Under 'API: PowerOfMath', the 'Resources' section is active, showing a list of resources: Stages, Authorizers, Gateway responses, Models, Resource policy, Documentation, Dashboard, and API settings. The main content area displays a resource configuration for a POST method. The path is set to '/'. The method is defined as 'POST'. The 'Request body' section contains the following JSON payload:

```
1 ▼ {  
2   "base" : 2,  
3   "exponent" : 4  
4 }
```

The screenshot shows the AWS API Gateway Resources page after a test has been run. The 'POST' method test results are displayed on the right. The test details are as follows:

- Request: /
- Status: 200
- Latency: 252
- Response body: {"statusCode": 200, "body": "\"Your result is 16.0\""}
Detailed view:
- 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=1-65d5d92e-6f7cc4532012e9913566d9cf;Parent=564961af01  
    8c1067;Sampled=0;lineage=2f1e445b:0"  
}
```
- Log: (empty)

Now we are creating DynamoDB table

The screenshot shows the AWS Cloud Console search results for the query 'dynamoDB'. The search bar at the top contains the text 'Search results for 'dy''. Below it, a sidebar lists recent services and other resources. The main content area is titled 'Services' and shows four results:

- DynamoDB** ☆
Managed NoSQL Database
Top features: Tables, Imports from S3, Explore Items, Clusters, Reserved Capacity
- CloudFront** ☆
Global Content Delivery Network
- Athena** ☆
Serverless interactive analytics service
- AWS Cloud Map** ☆
Build a dynamic map of your cloud

At the bottom of the search results, there are buttons for 'View all services' and 'Go to myApplications'.

The screenshot shows the Amazon DynamoDB service homepage. The URL is <https://us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#service>. The page has a dark header with the AWS logo and a search bar. The main content area is titled 'Amazon DynamoDB' and describes it as 'A fast and flexible NoSQL database service for any scale'. It includes a 'Get started' section with a 'Create table' button and a 'Pricing' section. On the left, a sidebar lists various options like Dashboard, Tables, and DAX. At the bottom, there's a 'How it works' section and standard footer links.

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

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.

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.

1 to 255 characters and case sensitive.



Auto Scaling On Yes

Local secondary indexes	-	No
Global secondary indexes	-	Yes
Encryption key management	Owned by Amazon DynamoDB	Yes
Deletion protection	Off	Yes

Tags

Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

[Add new tag](#)

You can add 50 more tags.

Cancel

Create table



The screenshot shows the AWS DynamoDB service dashboard. On the left, a sidebar menu includes options like Dashboard, Tables (selected), Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. Below this is a section for DAX Clusters. The main content area displays a success message: "The PowerOfMath table was created successfully." It shows a table named "PowerOfMath" with one item: "ID (S)" with value "0". The table has "Active" status, "Off" deletion protection, and "Provisioned (5)" read capacity units.

This screenshot shows the "View table" page for the "PowerOfMath" table. The left sidebar is identical to the previous screen. The main content area shows detailed information for the table, including Replication Regions (0 Regions), Date created (February 21, 2024, 16:41:08 UTC+05:30), and Integrations (0). It also displays the Amazon Resource Name (ARN) which is highlighted in yellow: `arn:aws:dynamodb:us-east-1:975050273614:table/PowerOfMath`. Below this, an "Items summary" section shows Item count (0) and Table size (0 bytes).

`arn:aws:dynamodb:us-east-1:975050273614:table/PowerOfMath`

The screenshot shows the AWS Lambda console interface. The top navigation bar includes tabs for AWS Amplify, PowerOfMathFunction - Lambda, API Gateway - Resources, and View table | Amazon DynamoDB. The main content area has a green header bar stating "The test event PowerOfMathtest was successfully saved." Below this, there are tabs for Code, Test, Monitor, Configuration (which is highlighted in yellow), Aliases, and Versions. Under the Configuration tab, there's a "Code source" section with an "Info" link and an "Upload from" button. A toolbar above the code editor includes File, Edit, Find, View, Go, Tools, Window, Test (which is selected and highlighted in blue), Deploy, and a gear icon. The code editor shows a file named lambda_function.py containing the following Python code:

```
lambda_function.py
# PowerOfMathFunction
# lambda_function.py

def lambda_handler(event, context):
    result = int(event['number']) ** 2
    return {
        "statusCode": 200,
        "body": f"Your result is {result}"
    }
```

The status bar at the bottom indicates the URL as https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/PowerOfMathFunction?tab=configure, the year 2024, and the date 21-02-2024.

This screenshot shows the AWS Lambda console with the General configuration tab selected (highlighted in yellow). The left sidebar contains a navigation menu with options: Triggers, Permissions (which is highlighted in yellow), Destinations, Function URL, Environment variables, and Tags. The main content area displays the "General configuration" settings:

General configuration		
Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	SnapStart	
0 min 3 sec	Info	None

The status bar at the bottom indicates the URL as https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/PowerOfMathFunction?tab=general, the year 2024, and the date 21-02-2024.

The screenshot shows the AWS Lambda function configuration page for 'PowerOfMathFunction'. The 'Execution role' tab is selected. On the left, a sidebar lists 'General configuration', 'Triggers', 'Permissions' (which is selected), 'Destinations', 'Function URL', and 'Environment variables'. The main area displays the 'Role name' as 'PowerOfMathFunction-role-uxjcxr7u'. Below it is a 'Resource summary' section with a note: 'To view the resources and actions that your function has permission to access, choose a service.' At the top right, there are 'Edit' and 'View role document' buttons.

The screenshot shows the AWS IAM roles details page for 'PowerOfMathFunction-role-uxjcxr7u'. The 'Summary' section includes details like creation date (February 21, 2024, 15:59 UTC+05:30), ARN (arn:aws:iam::975050273614:role/service-role/PowerOfMathFunction-role-uxjcxr7u), last activity (43 minutes ago), and maximum session duration (1 hour). Below the summary are tabs for 'Permissions', 'Trust relationships', 'Tags', 'Access Advisor', and 'Revoke sessions'. The 'Permissions' tab is selected, showing 'Permissions policies (1) Info'. A note says 'You can attach up to 10 managed policies.' At the bottom right, there are 'Delete', 'Edit', 'Simulate', 'Remove', and 'Add permissions' buttons. The left sidebar of the IAM console shows 'Identity and Access Management (IAM)' with sections for 'Dashboard', 'Access management' (User groups, Users, Roles, Policies, Identity providers, Account settings), and 'Access reports' (Access Analyzer).

Screenshot of the AWS IAM Role details page for 'PowerOfMathFunction-role-uxjcxr7u'.

Identity and Access Management (IAM)

Permissions (selected)

Permissions policies (1) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AWSLambdaBasicExecution...	Customer managed	1

Add permissions ▲

- Attach policies
- Create inline policy (highlighted)

Creation date: February 21, 2024, 15:59 (UTC+05:30)

Last activity: 43 minutes ago

ARN: arn:aws:iam::975050273614:role/service-role/PowerOfMathFunction-role-uxjcxr7u

Maximum session duration: 1 hour

Screenshot of the 'Create policy | IAM' step in the IAM Role creation process.

Step 1: Specify permissions

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

Visual JSON Actions ▾

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "Statement1",  
6       "Effect": "Allow",  
7       "Action": [],  
8       "Resource": []  
9     }  
10   ]  
11 }
```

Edit statement

Statement1

Add actions

Choose a service:

Filter services

Available services:

- AMP
- API Gateway
- API Gateway V2
- ASC

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 16:50 21-02-2024

The screenshot shows the AWS IAM Policy Editor interface. At the top, there are tabs for 'Visual' (selected), 'JSON', and 'Actions'. Below the tabs, the JSON code for the policy is displayed:

```
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 }
```

To the right of the JSON code, there is a sidebar with the title 'Edit statement' and a sub-section 'Select a statement' with the instruction 'Select an existing statement in the policy or add a new statement'. A button '+ Add new statement' is also present.



```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "VisualEditor0",  
      "Effect": "Allow",  
      "Action": [  
        "dynamodb:PutItem",  
        "dynamodb>DeleteItem",  
        "dynamodb:GetItem",  
        "dynamodb:Scan",  
        "dynamodb:Query",  
        "dynamodb:UpdateItem"  
      ],  
      "Resource": "arn:aws:dynamodb:us-east-1:975050273614:table/PowerOfMath"  
    },  
    {  
      "Sid": "VisualEditor1",  
      "Effect": "Allow",  
      "Action": [  
        "lambda:InvokeFunction"  
      ],  
      "Resource": "arn:aws:lambda:us-east-1:975050273614:function:PowerOfMathFunction"  
    }  
  ]  
}
```

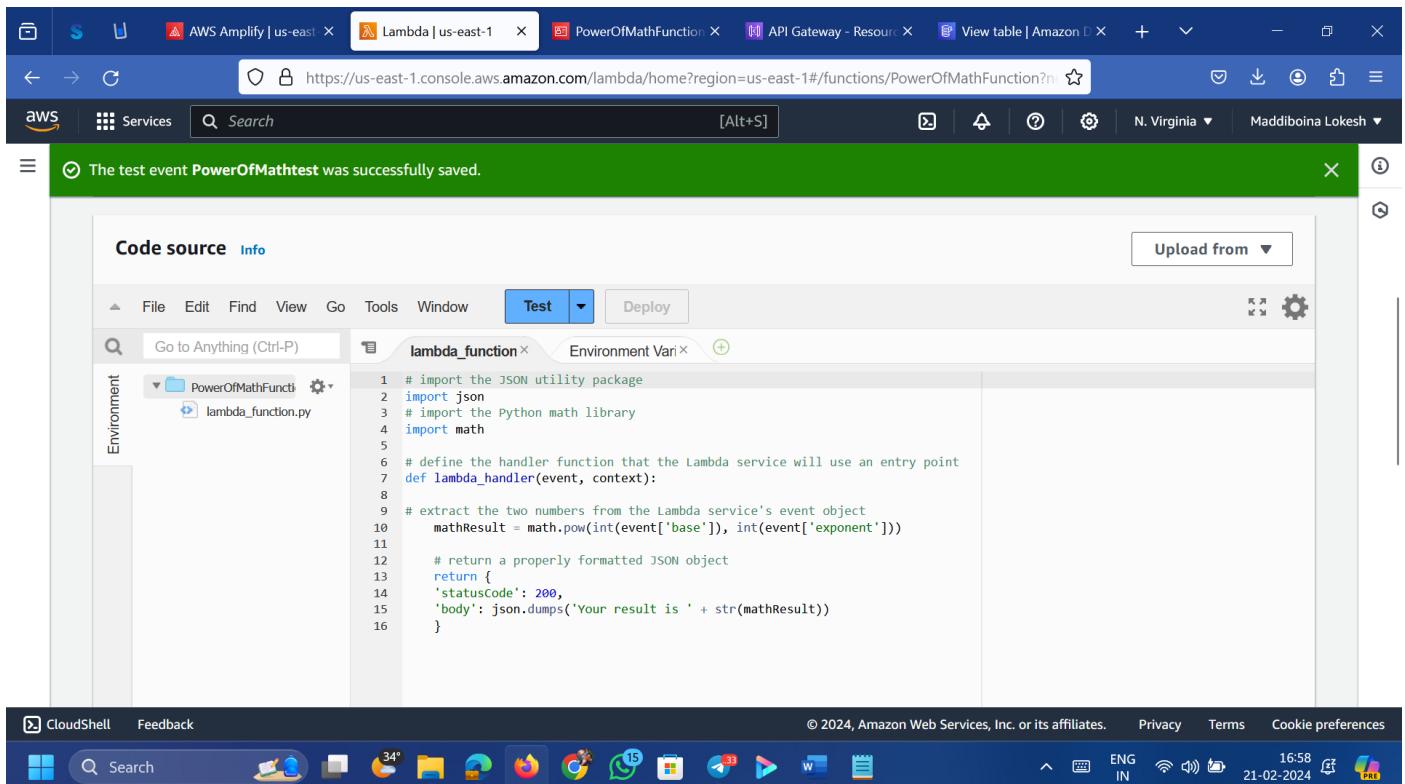
The screenshot shows the AWS IAM console interface. A new policy is being created with the following details:

- Policy name:** PowerOfMathDynamoPolicy
- Permissions defined in this policy:** Allow (1 of 404 services)
- Service:** DynamoDB
- Access level:** Limited: Read, Write
- Resource:** region| string like |us-east-1, TableName| string like |PowerOfMath
- Request conditions:** None

The screenshot shows the AWS IAM console interface after the policy has been created. The newly created policy is listed under the 'Permissions policies' section:

- Permissions policies (2):** AWSLambdaBasicExecution... (Customer managed) and PowerOfMathDynamoPolicy (Customer inline).

Updating code

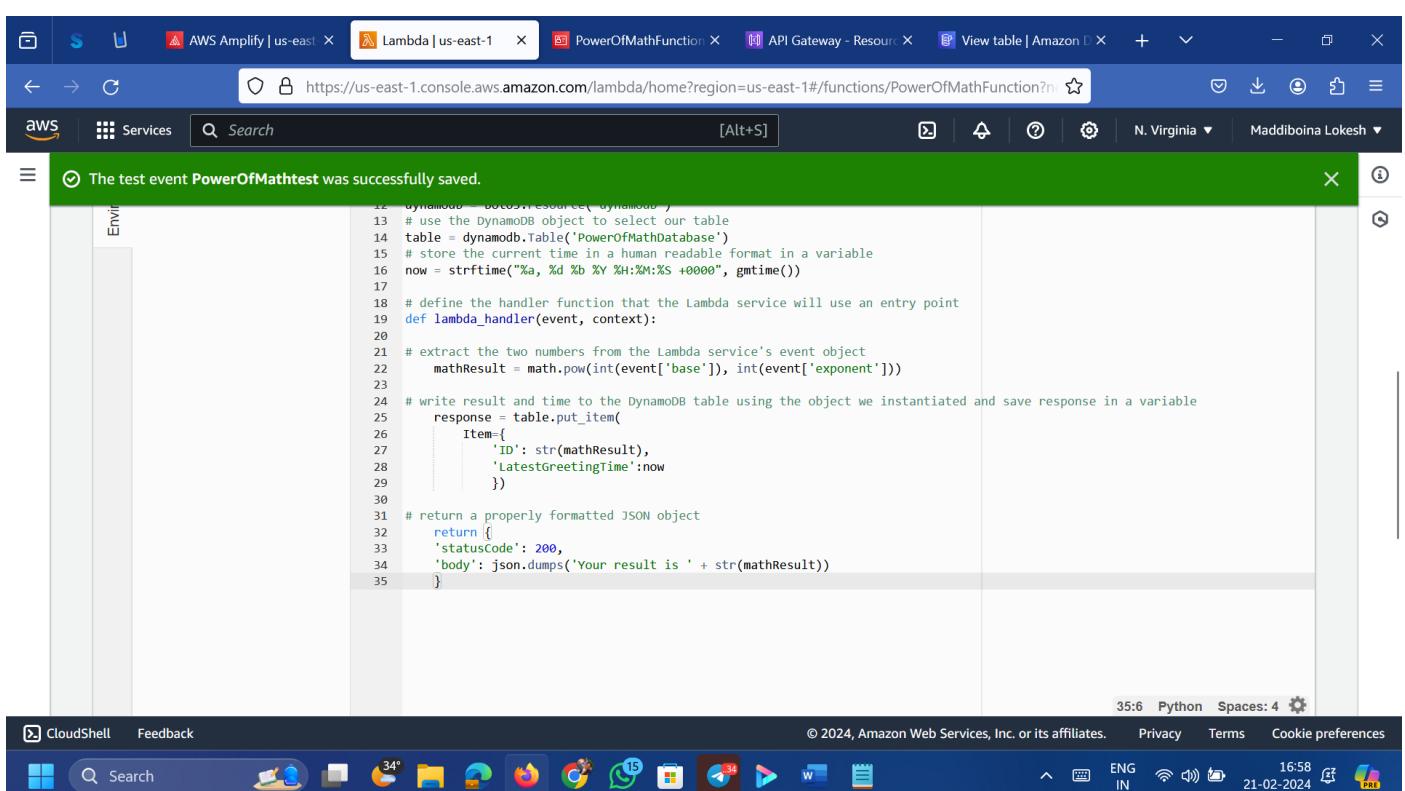


```
# import the JSON utility package
import json
# import the Python math library
import math

# define the handler function that the Lambda service will use as 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))
    }
```



```
dynamodb = boto3.resource('dynamodb')
# use the DynamoDB object to select our table
table = dynamodb.Table('PowerOfMathDatabase')
# store the current time in a human readable format in a variable
now = strftime("%a, %d %b %Y %H:%M:%S +0000", gmtime())
# define the handler function that the Lambda service will use as 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']))

    # write result and time to the DynamoDB table using the object we instantiated and save response in a variable
    response = table.put_item(
        Item={
            'ID': str(mathResult),
            'LatestGreetingTime':now
        })
    # return a properly formatted JSON object
    return {
        'statusCode': 200,
        'body': json.dumps('Your result is ' + str(mathResult))
    }
```

```
# import the JSON utility package
import json
# import the Python math library
import math

# import the AWS SDK (for Python the package name is boto3)
```

```

import boto3

# import two packages to help us with dates and date formatting
from time import gmtime, strftime

# create a DynamoDB object using the AWS SDK
dynamodb = boto3.resource('dynamodb')

# use the DynamoDB object to select our table
table = dynamodb.Table('PowerOfMath')

# store the current time in a human readable format in a variable
now = strftime("%a, %d %b %Y %H:%M:%S +0000", gmtime())

# define the handler function that the Lambda service will use as 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']))

    # write result and time to the DynamoDB table using the object we instantiated and save response in a variable
    response = table.put_item(
        Item={
            'ID': str(mathResult),
            'LatestGreetingTime': now
        }
    )

    # return a properly formatted JSON object
    return {
        'statusCode': 200,
        'body': json.dumps('Your result is ' + str(mathResult))
    }

```

The screenshot shows the AWS Lambda console interface. A green success message at the top states "Successfully updated the function PowerOfMathFunction." Below this, the "Code" tab is selected in the navigation bar. The main area displays the "Code source" tab with the "Info" sub-tab selected. A toolbar above the code editor includes "File", "Edit", "Find", "View", "Go", "Tools", "Window", "Test" (which is currently selected), and "Deploy". The code editor window shows a file named "lambda_function.py" containing Python code for a Lambda function. The code uses the DynamoDB object to interact with a table named "PowerofMath". It defines a handler function "lambda_handler" that takes an event and context, extracts base and exponent values from the event, and writes the result to the DynamoDB table. The code editor also shows environment variables and a sidebar for the function's configuration.

The screenshot shows the AWS Lambda console interface after testing the function. The "Execution result" tab is selected in the navigation bar. The results show a successful execution with a status of "Succeeded", a maximum memory used of 71 MB, and a duration of 271.81 ms. The "Test Event Name" is listed as "PowerOfMathtest". The "Response" field displays a JSON object with a status code of 200 and a body containing the string "Your result is 8.0". The "Function Logs" section shows the request and response details, including RequestId, Version, and Duration. The "Request ID" is listed as "31e83de5-5743-4bd4-8ec4-6619c9c417c9".

Now Result is in table

DynamoDB

PowerOfMath

Completed. Read capacity units consumed: 0.5

ID (String)	LatestGreetingTime
8.0	Wed, 21 Feb 2024 11:34:30 +0000

```

D: > 3 year even sem > serverless csc > project > index-ORIGINAL.html > ...
52     // callAPI function that takes the base and exponent numbers as parameters
53     var callAPI = (base, exponent) => {
54         // instantiate a headers object
55         var myHeaders = new Headers();
56         // add content type header to object
57         myHeaders.append("Content-Type", "application/json");
58         // using built in JSON utility package turn object to string and store in a variable
59         var raw = JSON.stringify({ "base": base, "exponent": exponent });
60         // create a JSON object with parameters for API call and store in a variable
61         var requestOptions = {
62             method: 'POST',
63             headers: myHeaders,
64             body: raw,
65             redirect: 'follow'
66         };
67         // make API call with parameters and use promises to get response
68         fetch("YOUR API GATEWAY ENDPOINT", requestOptions)
69             .then(response => response.text())
70             .then(result => alert(JSON.parse(result).body))
71             .catch(error => console.log('error', error));
72     }
73     </script>
74 </head>
75 <body>
76     <h1>TO THE POWER OF MATH!</h1>
77     <form>
78         <label>Base number:</label>
79         <input type="text" id="base">

```

Updating Amplify code

File Edit Selection View Go Run ... ← → Search

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

```

1 index-ORIGINAL.html •
D: > 3 year even sem > servless csc > project > index-ORIGINAL.html > html > head > script > callAPI
52 // callAPI function that takes the base and exponent numbers as parameters
53 var callAPI = (base, exponent) => {
54   // instantiate a Headers object
55   var myHeaders = new Headers();
56   // add content type header to object
57   myHeaders.append("Content-Type", "application/json");
58   // using built in JSON utility package turn object to string and store in a variable
59   var raw = JSON.stringify({ "base": base, "exponent": exponent });
60   // create a JSON object with parameters for API call and store in a variable
61   var requestOptions = {
62     method: 'POST',
63     headers: myHeaders,
64     body: raw,
65     redirect: 'follow'
66   };
67   // make API call with parameters and use promises to get response
68   fetch("https://rhh115iipj.execute-api.us-east-1.amazonaws.com/dev", requestOptions)
69     .then(response => response.text())
70     .then(result => alert(JSON.parse(result).body))
71     .catch(error => console.log('error', error));
72 }
73 </script>
74 </head>
75 <body>
76   <h1>TO THE POWER OF MATH!</h1>
77   <form>
78     <label>Base number:</label>
79     <input type="text" id="base">

```

Screen Reader Optimized | LIn 68, Col 78 | Spaces: 4 | UTF-8 | CRLF | HTML | Prettier

File Upload 34° ENG IN 17:13 21-02-2024

File Upload

File Upload

Organize New folder

Name Date modified

- Desktop
- Documents
- Pictures
- This PC
- Windows (C)
- Local Disk (D)
- Network
- Linux

File name: All Files Open Cancel

API Gateway - Resource Items | Amazon Dynan region=us-east-1#/dh8s2t5nha05g N. Virginia Maddiboina Lokesh

Deployment successfully completed.

Domain https://dev.dh8s2t5nha05g.amplifyapp.com Last deployment 21/2/2024, 3:53:43 pm

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

Choose files

Monitoring
Rewrites and redirects
Custom headers

Documentation

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

File Upload 34° ENG IN 17:15 21-02-2024

S U AWS Amplify | us-east-1 X PowerOfMathFunction X PowerOfMathFunction X API Gateway - Resource X Items | Amazon Dynan X + - ×

https://us-east-1.console.aws.amazon.com/amplify/home?region=us-east-1#/dh8s2t5nha05g

AWS Services Search [Alt+S] N. Virginia Maddiboina Lokesh

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

AWS Amplify

All apps **PowerOfMath**

App settings

- General
- Amplify Studio settings
- Domain management
- Notifications
- Access control
- Monitoring
- Rewrites and redirects
- Custom headers

Documentation

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

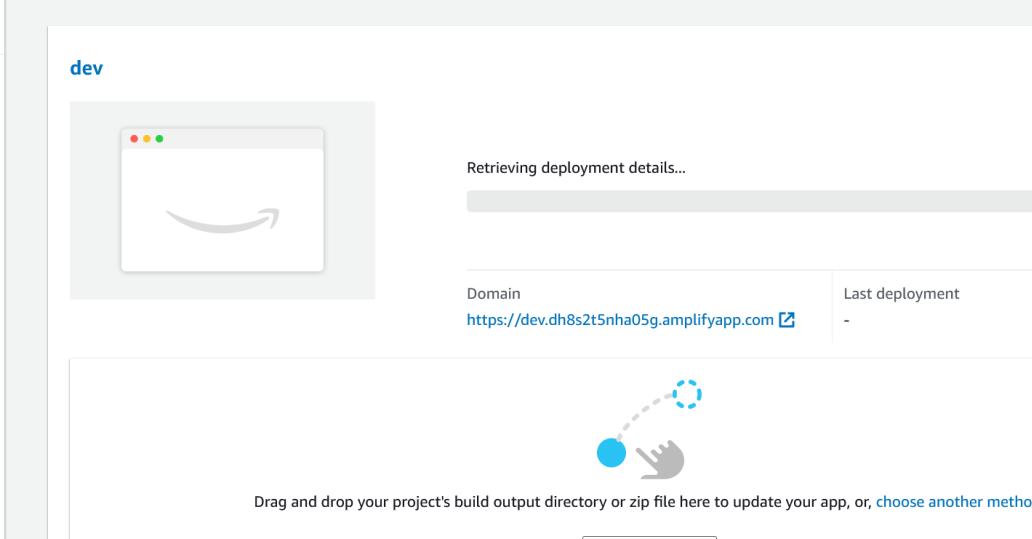
34° 17:16 21-02-2024 ENG IN

dev

Retrieving deployment details...

Domain https://dev.dh8s2t5nha05g.amplifyapp.com Last deployment -

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

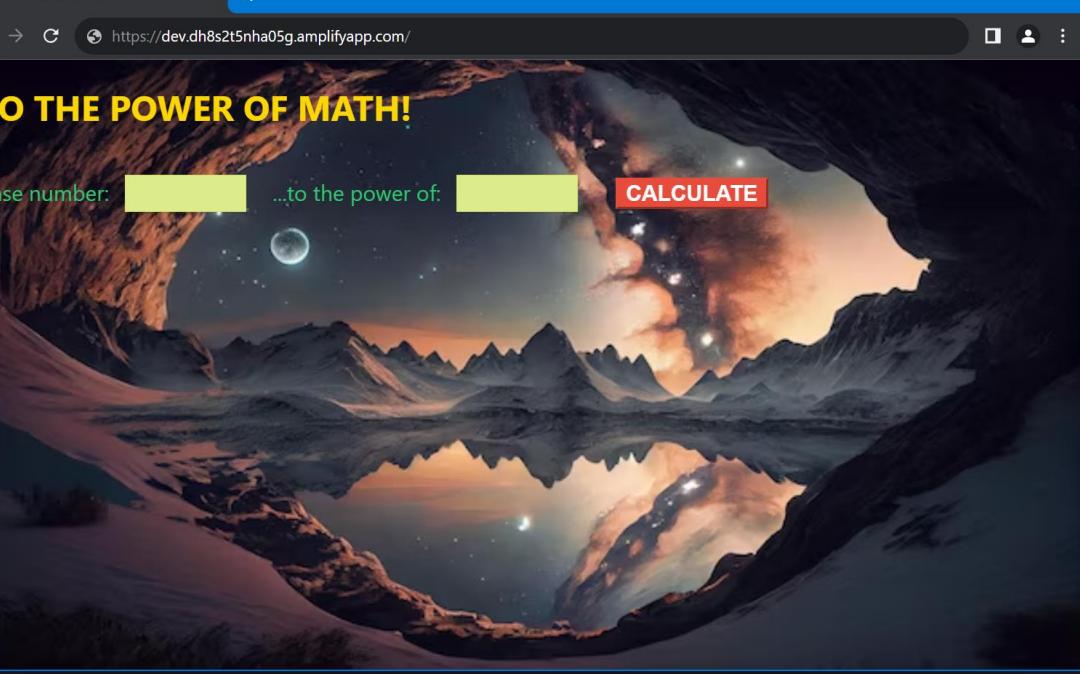


To the Power of Math! x +

https://dev.dh8s2t5nha05g.amplifyapp.com/

TO THE POWER OF MATH!

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



34° 17:17 21-02-2024 ENG IN

