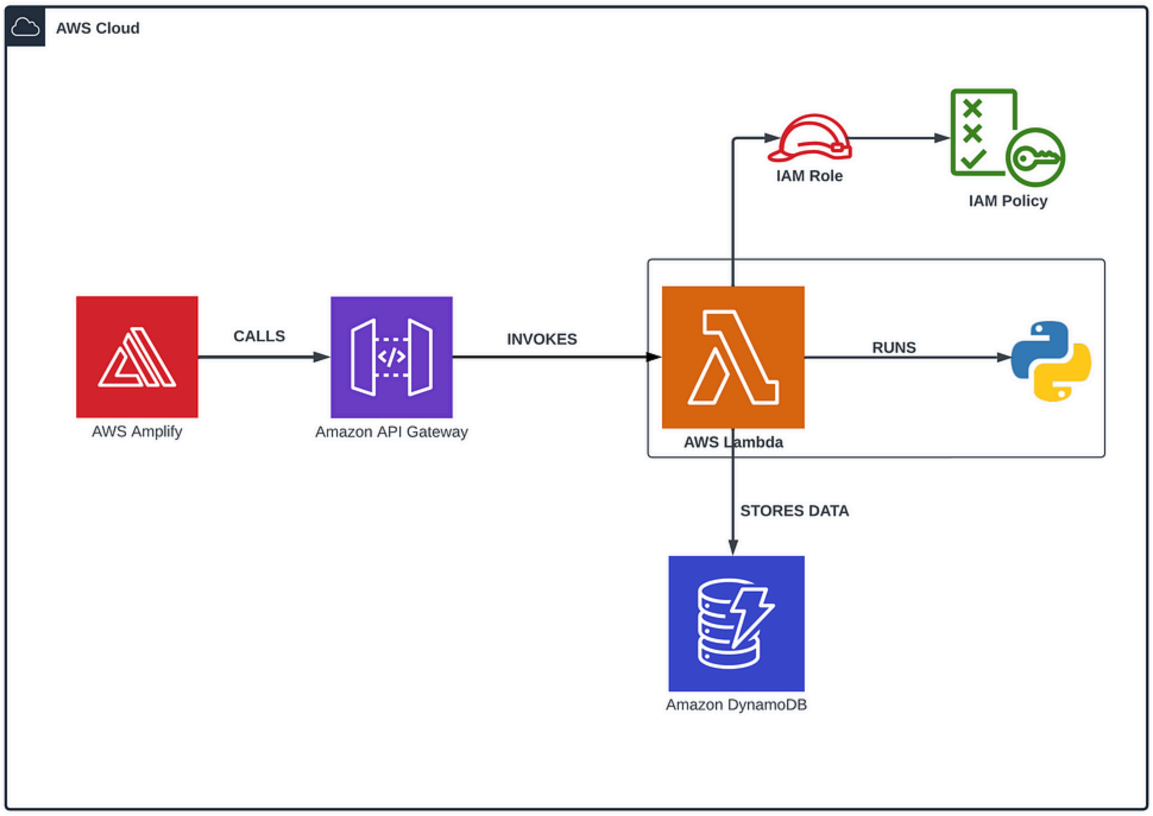
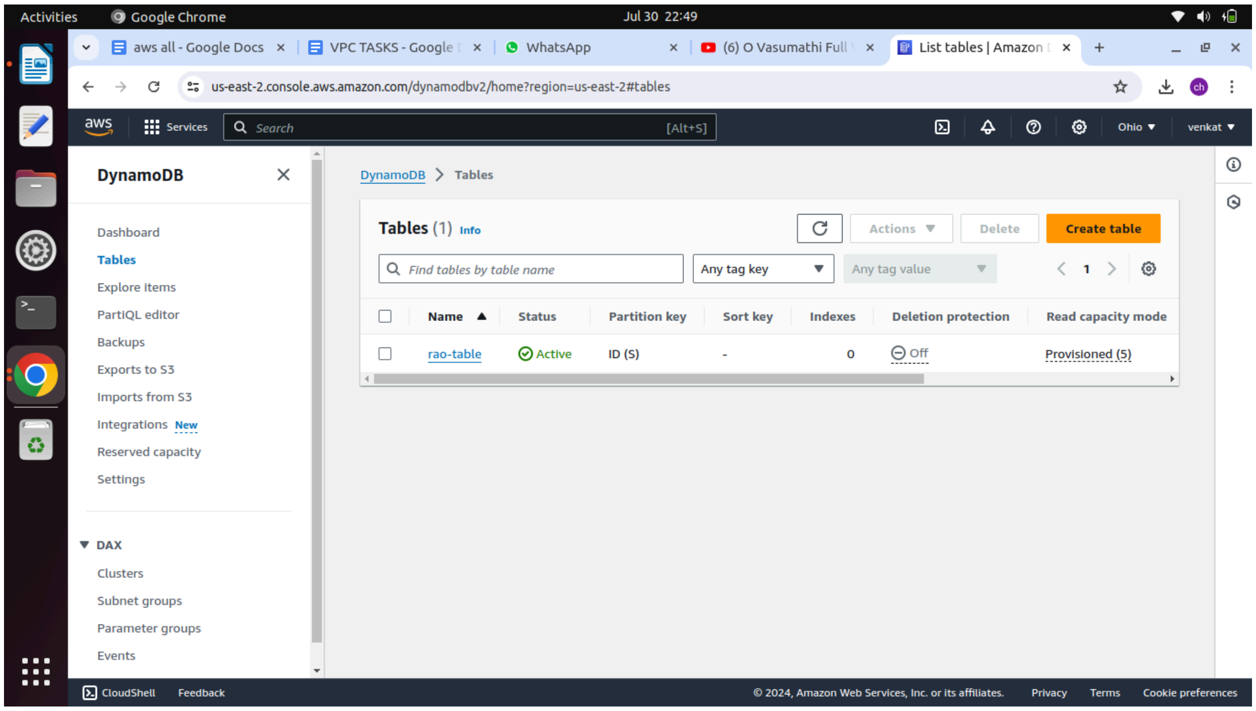
lambda and dynamo db and api gateway



1) Go to dynamoDB and click create table
2) Table name= rao-table
3) Partition key= ID
4) Click create table

5) 

Activities   Google Chrome   Jul 30 22:49

aws all - Google Docs | VPC TASKS - Google | WhatsApp | (6) O Vasumathi Full | List tables | Amazon

us-east-2.console.aws.amazon.com/dynamodbv2/home?region=us-east-2#tables

DynamoDB > Tables

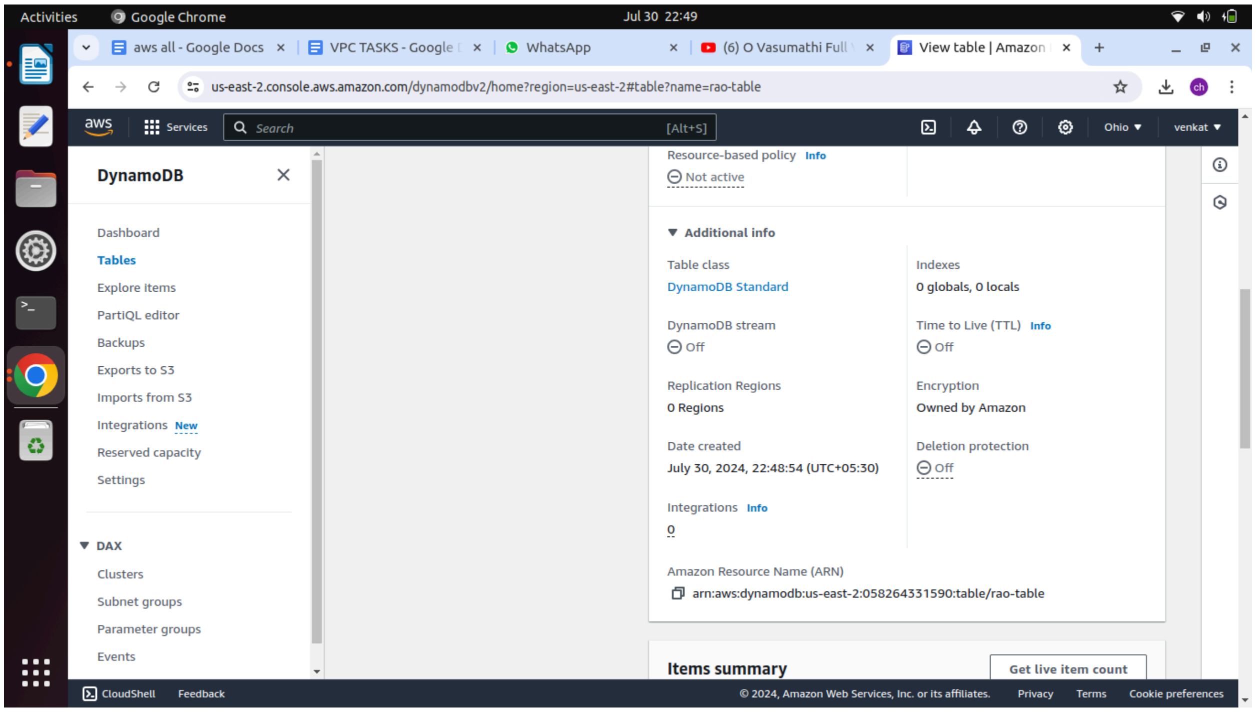
Tables (1) Info

Actions | Delete | Create table

Find tables by table name | Any tag key | Any tag value

Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode
rao-table	Active	ID (S)	-	0	Off	Provisioned (5)

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

6) 

Activities   Google Chrome   Jul 30 22:49

aws all - Google Docs | VPC TASKS - Google | WhatsApp | (6) O Vasumathi Full | View table | Amazon

us-east-2.console.aws.amazon.com/dynamodbv2/home?region=us-east-2#table?name=rao-table

Resource-based policy Info
Not active

Additional info

Table class: DynamoDB Standard
Indexes: 0 globals, 0 locals
DynamoDB stream: Off
Time to Live (TTL) Info: Off
Replication Regions: 0 Regions
Encryption: Owned by Amazon
Date created: July 30, 2024, 22:48:54 (UTC+05:30)
Deletion protection: Off
Integrations Info: 0

Amazon Resource Name (ARN)
arn:aws:dynamodb:us-east-2:058264331590:table/rao-table

Items summary   Get live item count

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

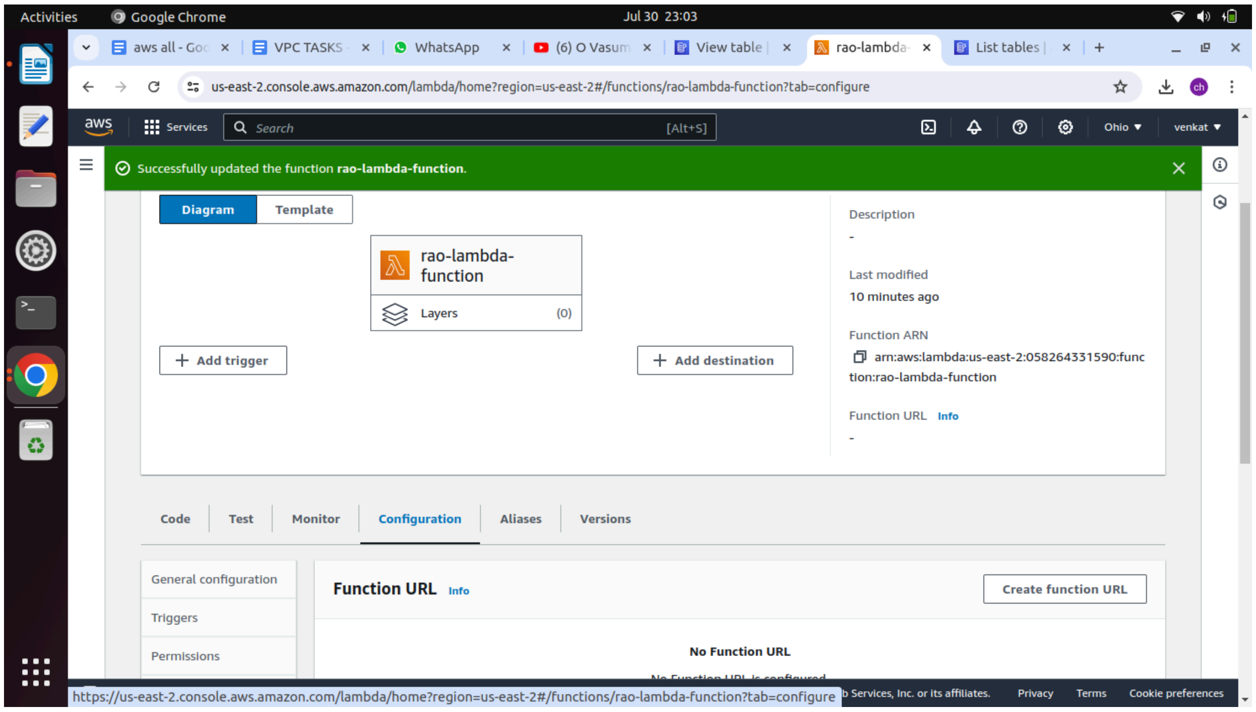
7) Open rao-table copy arn= arn:aws:dynamodb:us-east-2:058264331590:table/rao-table
8) Go to lambda click create function
9) Select = author from scratch
10) Function name= rao-lambda-function
11) runtime= python
12) Click create function
13) Open rao-lambda-function put code

```
# 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('rao-table')
# 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 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))
    }
```

14) Click deploy
15) Click test
16) Test event action =create new event
17) Event name = test1
18) Event json=

```
{
  "base": "2",
  "exponent": "4"
}
```

19) Click save



us-east-2.console.aws.amazon.com/lambda/home?region=us-east-2#/functions/rao-lambda-function?tab=configure

Successfully updated the function **rao-lambda-function**.

rao-lambda-function

Description
-

Last modified
10 minutes ago

Function ARN
arn:aws:lambda:us-east-2:058264331590:function:rao-lambda-function

Function URL
-

Function URL

No Function URL

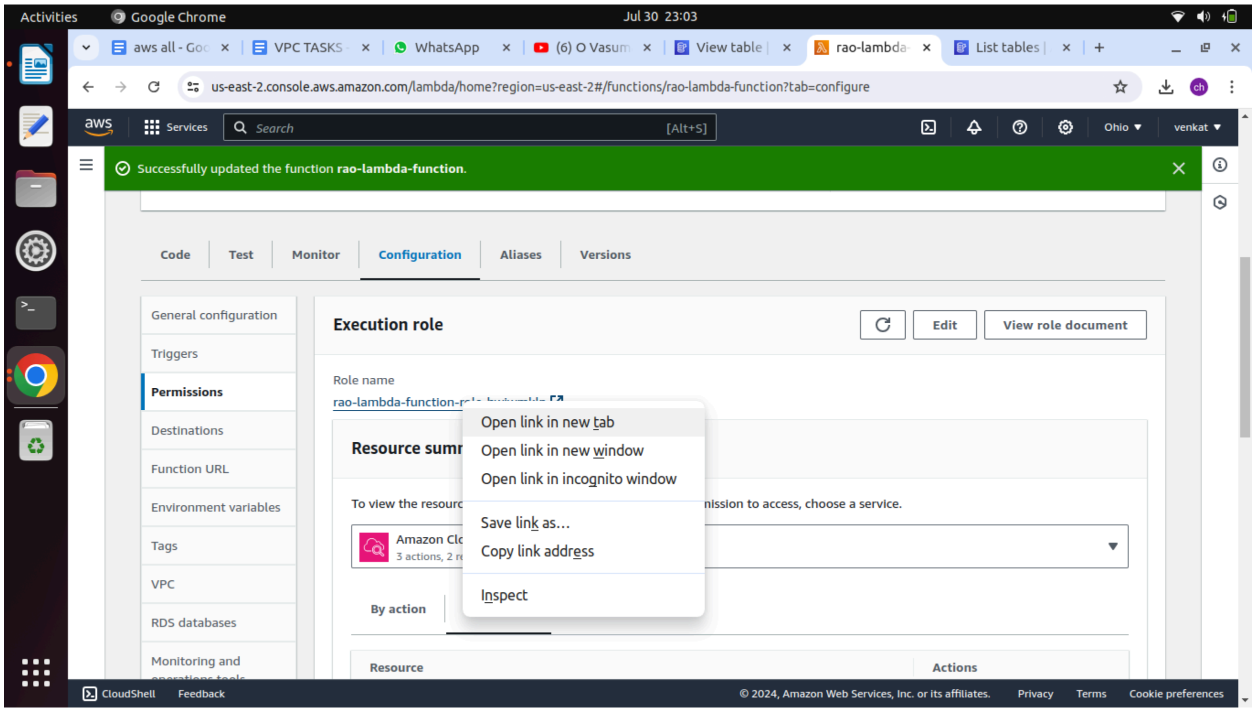
No Function URL is configured

https://us-east-2.console.aws.amazon.com/lambda/home?region=us-east-2#/functions/rao-lambda-function?tab=configure

20)

21) Click configuration

22) Click permission rao-lambda-function-role right click open new tab



us-east-2.console.aws.amazon.com/lambda/home?region=us-east-2#/functions/rao-lambda-function?tab=configure

Successfully updated the function **rao-lambda-function**.

Execution role

Role name
rao-lambda-function-role-

Resource summ

To view the resourc ... ission to access, choose a service.

Amazon Cl
3 actions, 2 r

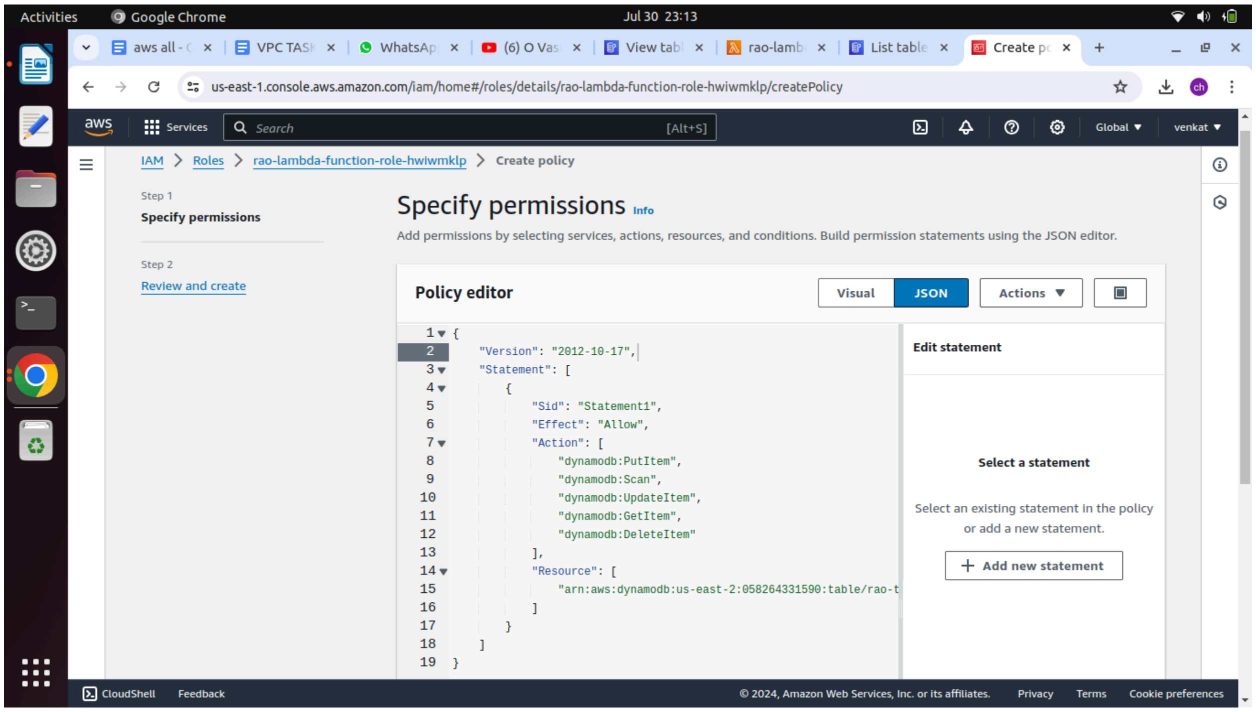
© 2024, Amazon Web Services, Inc. or its affiliates.

23)

24) Click add permissions click create inline policy

25) Select json

26) Dynamodb and add permissions = PutItem, Scan , UpdateItem , GetItem , DeleteItem
27) Add resource = here use dynamoDB arn
28) Click next

29) 

us-east-1.console.aws.amazon.com/iam/home#/roles/details/rao-lambda-function-role-hwiwmklp/createPolicy

IAM > Roles > rao-lambda-function-role-hwiwmklp > Create policy

Step 1
Specify permissions

Step 2
Review and create

Specify permissions Info

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

Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [
                "dynamodb:PutItem",
                "dynamodb:Scan",
                "dynamodb:UpdateItem",
                "dynamodb:GetItem",
                "dynamodb:DeleteItem"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-2:058264331590:table/rao-t
            ]
        }
    ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

==

or use this

```
{
        "Version": "2012-10-17",
        "Statement": [
                {
                        "Sid": "Statement1",
                        "Effect": "Allow",
                        "Action": [
                                "dynamodb:PutItem",
                                "dynamodb:Scan",
                                "dynamodb:UpdateItem",
                                "dynamodb:GetItem",
                                "dynamodb:DeleteItem"
                        ],
                        "Resource": ["arn:aws:dynamodb:us-east-2:058264331590:table/rao-table"]
                }
        ]
}
```

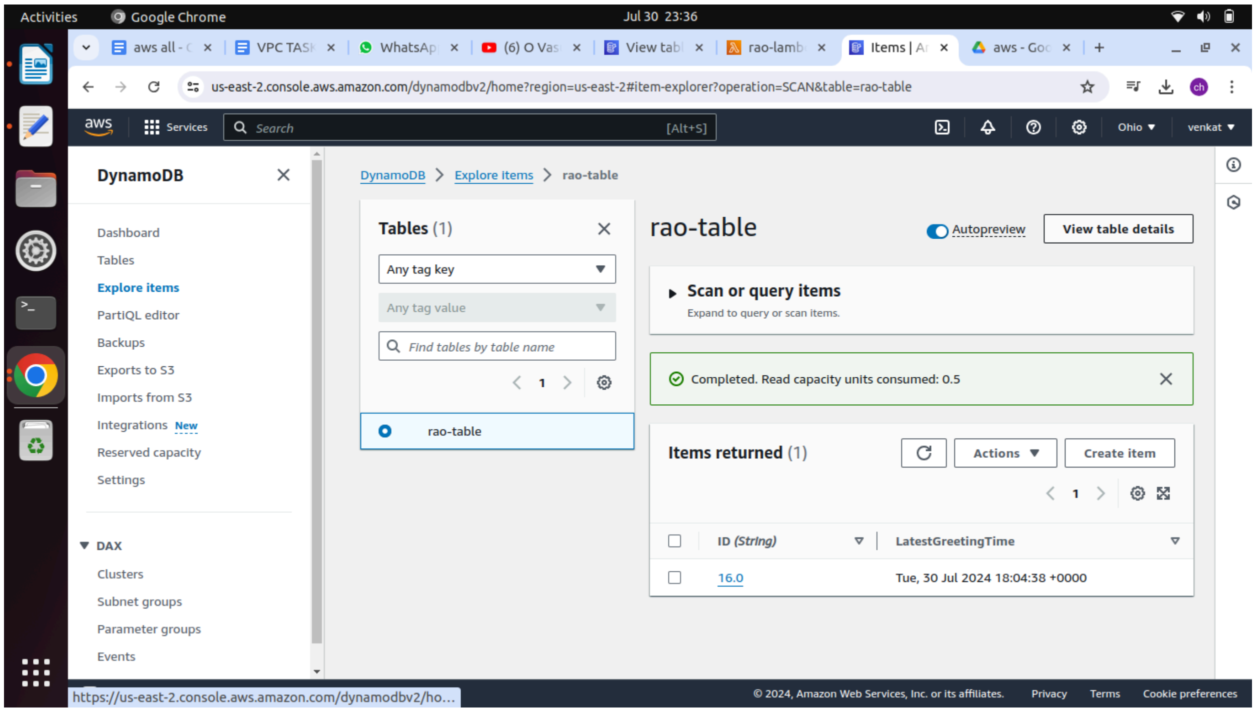
==

30) Policy name = rao-lambda-dynamo-policy
31) Clik crete policy
32) Go to lambda click code click deploy clik test give name json

```
{
  "base": "2",
  "exponent": "4"
}
```

33) Now click test

34) Now go to dynamoDB click explore items check value =16 available or not



us-east-2.console.aws.amazon.com/dynamodbv2/home?region=us-east-2#item-explorer?operation=SCAN&table=rao-table

DynamoDB > Explore items > rao-table

rao-table

Completed. Read capacity units consumed: 0.5

Items returned (1)

ID (String)	LatestGreetingTime
16.0	Tue, 30 Jul 2024 18:04:38 +0000

https://us-east-2.console.aws.amazon.com/dynamodbv2/ho...

35) Now go to amazon api gateway select Restapi click build

36) New api and api name= raoapi

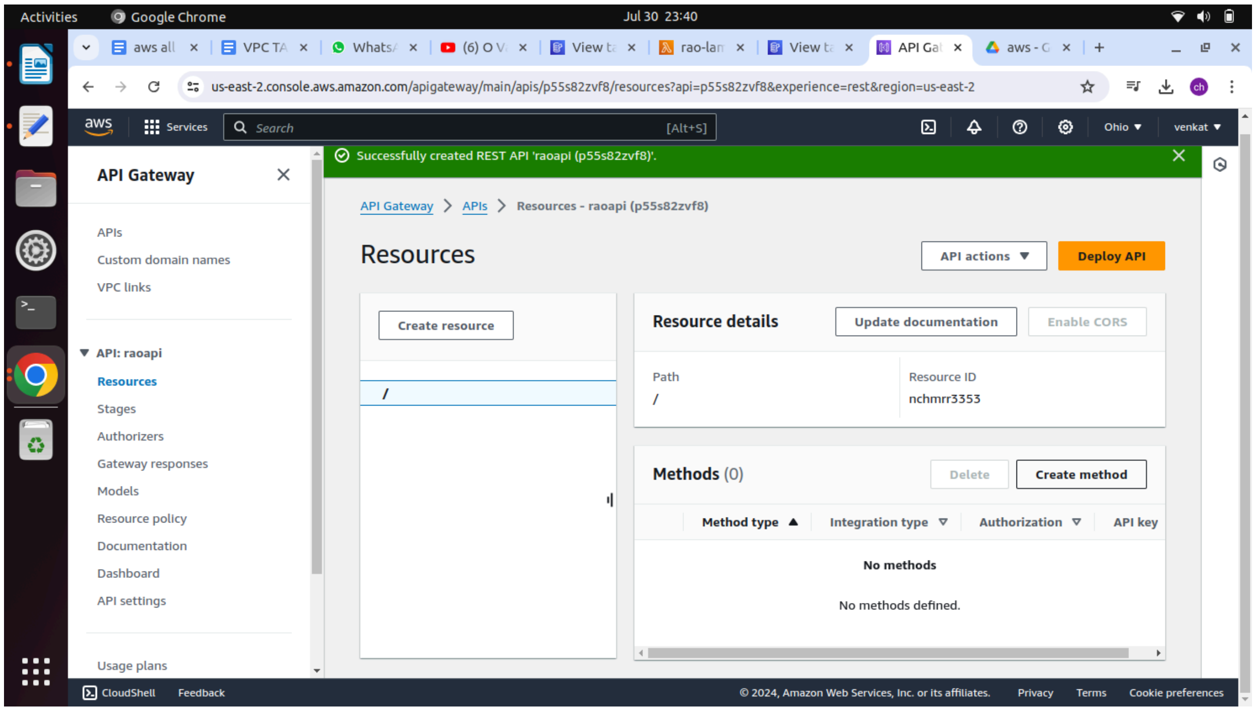
37) End point= edge click crete api

38) Click crete method and Method type= post

39) Integration type= lambda function and choose rao-lambda-function

40) Click create method and Click deploy api and select stage= new stage

41) Stage name= dev and Click deploy


Successfully created REST API 'raoapi (p55s82zvf8)'.

API Gateway > APIs > Resources - raoapi (p55s82zvf8)

Resources

Resource details

Path: /

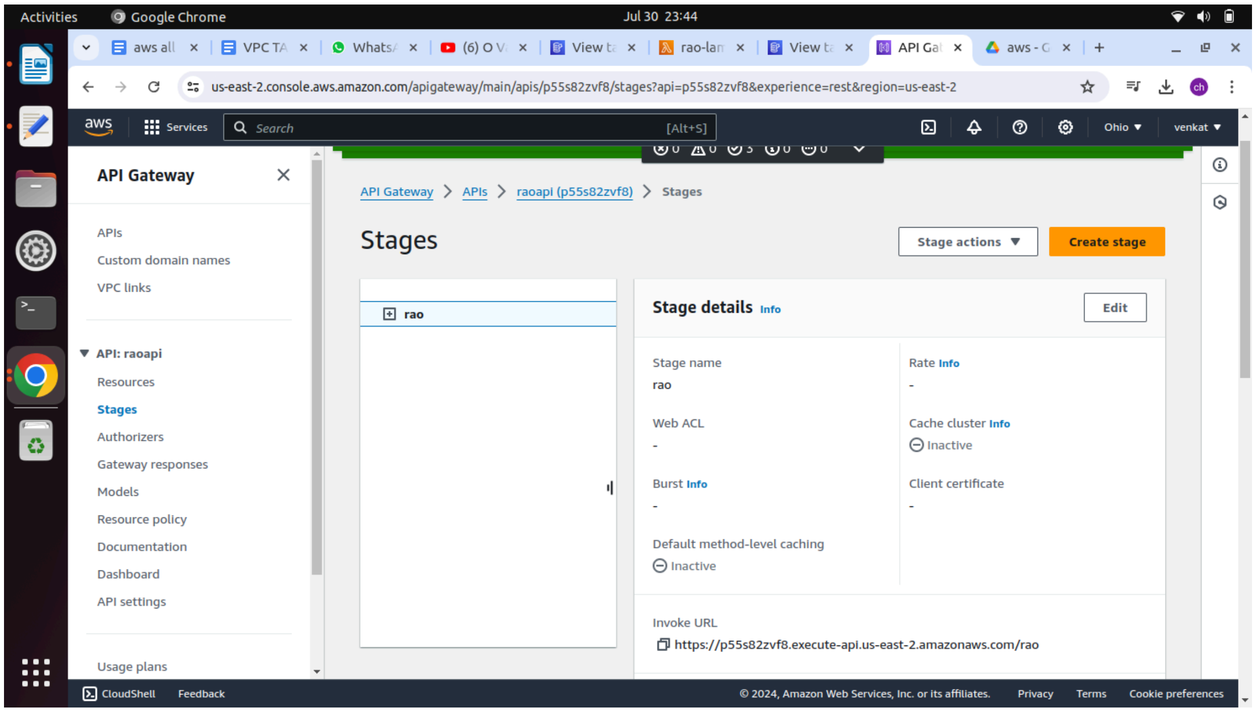
Resource ID: nchmrr3353

Methods (0)

Method type	Integration type	Authorization	API key

No methods

No methods defined.


API Gateway > APIs > raoapi (p55s82zvf8) > Stages

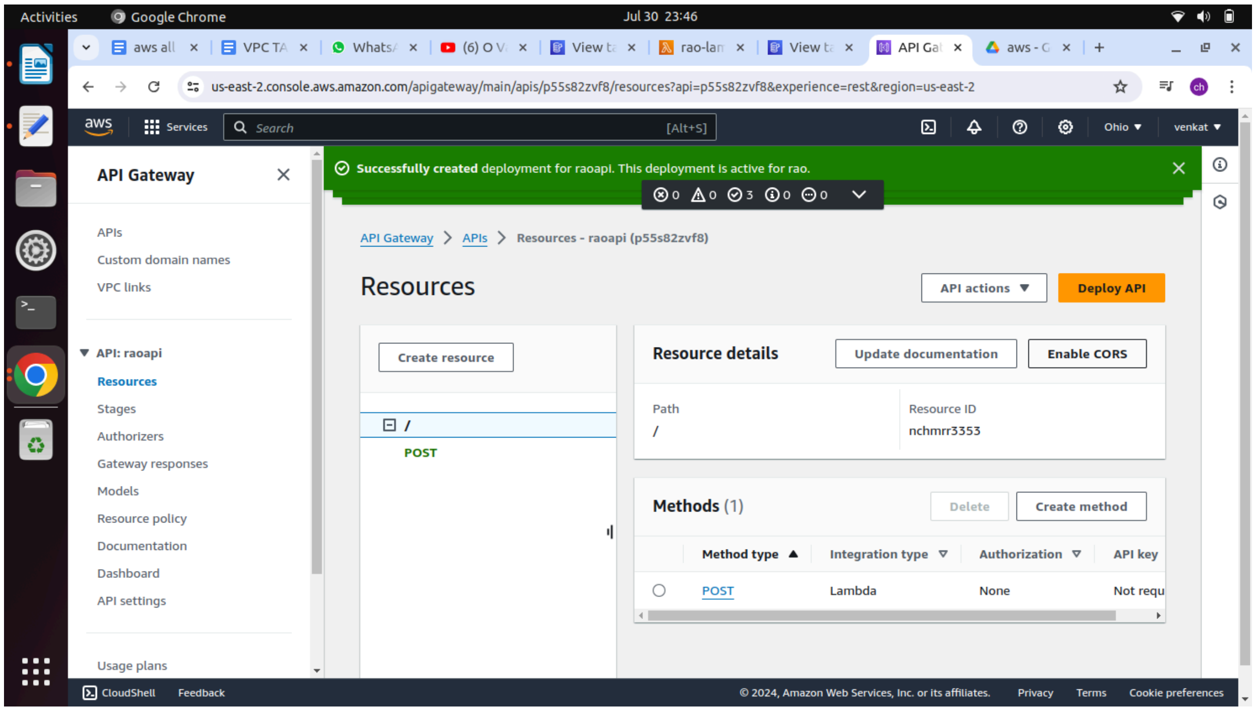
Stages

Stage details

Stage name: rao	Rate: -
Web ACL: -	Cache cluster: Inactive
Burst: -	Client certificate: -
Default method-level caching: Inactive	

Invoke URL: https://p55s82zvf8.execute-api.us-east-2.amazonaws.com/rao

42) Now take invoke url copy it
43) Resource click enable cros



Successfully created deployment for raoapi. This deployment is active for rao.

API Gateway > APIs > Resources - raoapi (p55s82zvf8)

/
POST

Resource details

Path	Resource ID
/	nchmrr3353

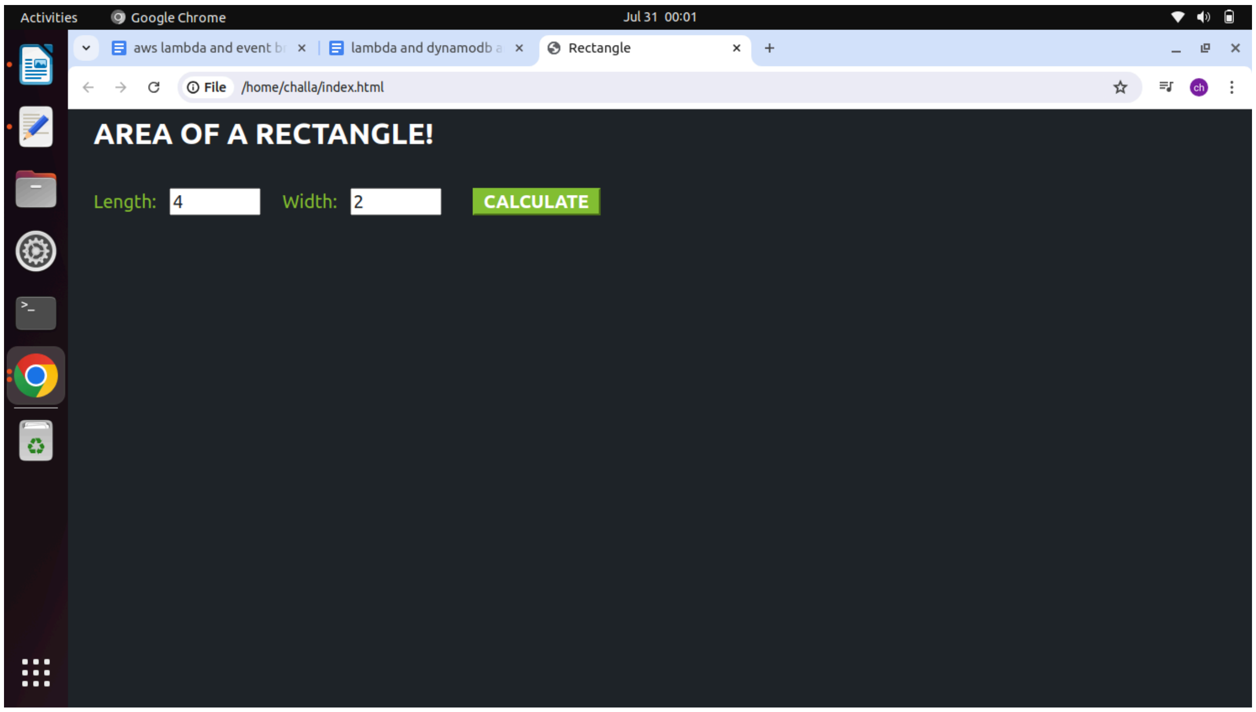
Methods (1)

	Method type	Integration type	Authorization	API key
○	POST	Lambda	None	Not requ

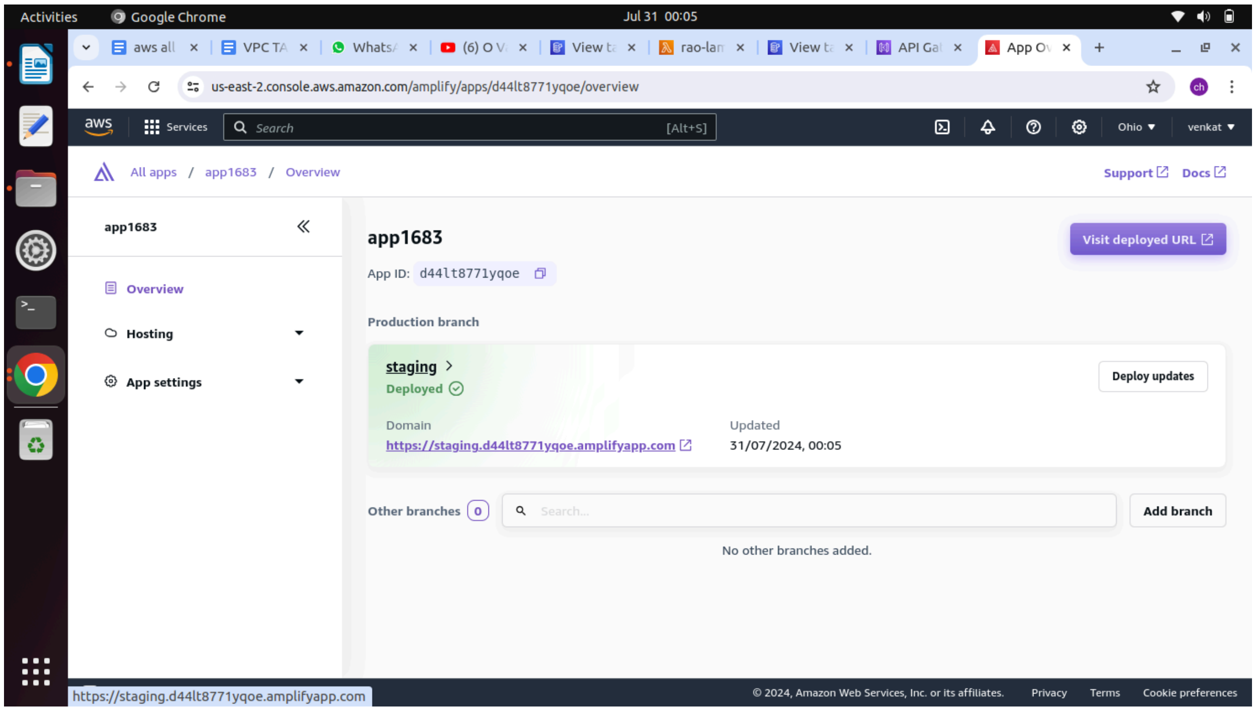
44) Gateway response= 4xx
45) method=post click save
46) Open nate pad use below code

```
<html>
<head>
    <meta charset="UTF-8">
    <title>Rectangle</title>
    <!-- Styling for the client UI -->
    <style>
    h1 {
        color: #FFFFFF;
        font-family: system-ui;
  margin-left: 20px;
        }
 body {
        background-color: #222629;
        }
    label {
        color: #86C232;
        font-family: system-ui;
        font-size: 20px;
        margin-left: 20px;
  margin-top: 20px;
        }
     button {
        background-color: #86C232;
  border-color: #86C232;
  color: #FFFFFF;
        font-family: system-ui;
        font-size: 20px;
  font-weight: bold;
        margin-left: 30px;
  margin-top: 20px;
  width: 140px;
        }
  input {
        color: #222629;
        font-family: system-ui;
        font-size: 20px;
        margin-left: 10px;
  margin-top: 20px;
  width: 100px;
        }
    </style>
    <script>
        // callAPI function that takes the length and width numbers as parameters
        var callAPI = (length,width)=>{
            // instantiate a headers object
            var myHeaders = new Headers();
            // add content type header to object
            myHeaders.append("Content-Type", "application/json");
            // using built in JSON utility package turn object to string and store in a variable
            var raw = JSON.stringify({"length":length,"width":width});
            // create a JSON object with parameters for API call and store in a variable
            var requestOptions = {
                method: 'POST',
                headers: myHeaders,
                body: raw,
                redirect: 'follow'
            };
            // make API call with parameters and use promises to get response
            fetch("https://p55s82zvf8.execute-api.us-east-2.amazonaws.com/rao", requestOptions)
            .then(response => response.text())
            .then(result => alert(JSON.parse(result).body))
            .catch(error => console.log('error', error));
        }
    </script>
</head>
<body>
    <h1>AREA OF A RECTANGLE!</h1>
 <form>
        <label>Length:</label>
        <input type="text" id="length">
        <label>Width:</label>
        <input type="text" id="width">
        <!-- set button onClick method to call function we defined passing input values as parameters
-->
        <button type="button"
onclick="callAPI(document.getElementById('length').value,document.getElementById('width').value)"
>CALCULATE</button>
    </form>
</body>
</html>
```

47) Above code use rao-api invoke url save file as index.html
48) Open the file you see



49) The html file create zip file name= rao.zip (right click on file click compress)
50) Now go to amplify
51) Click create new app click Deploy without git click next
52) Choose zip folder select rao.zip click save and deploy
53) Now you see domain below link click it



app1683

App ID: d44lt8771yqoe

Production branch

staging
Deployed

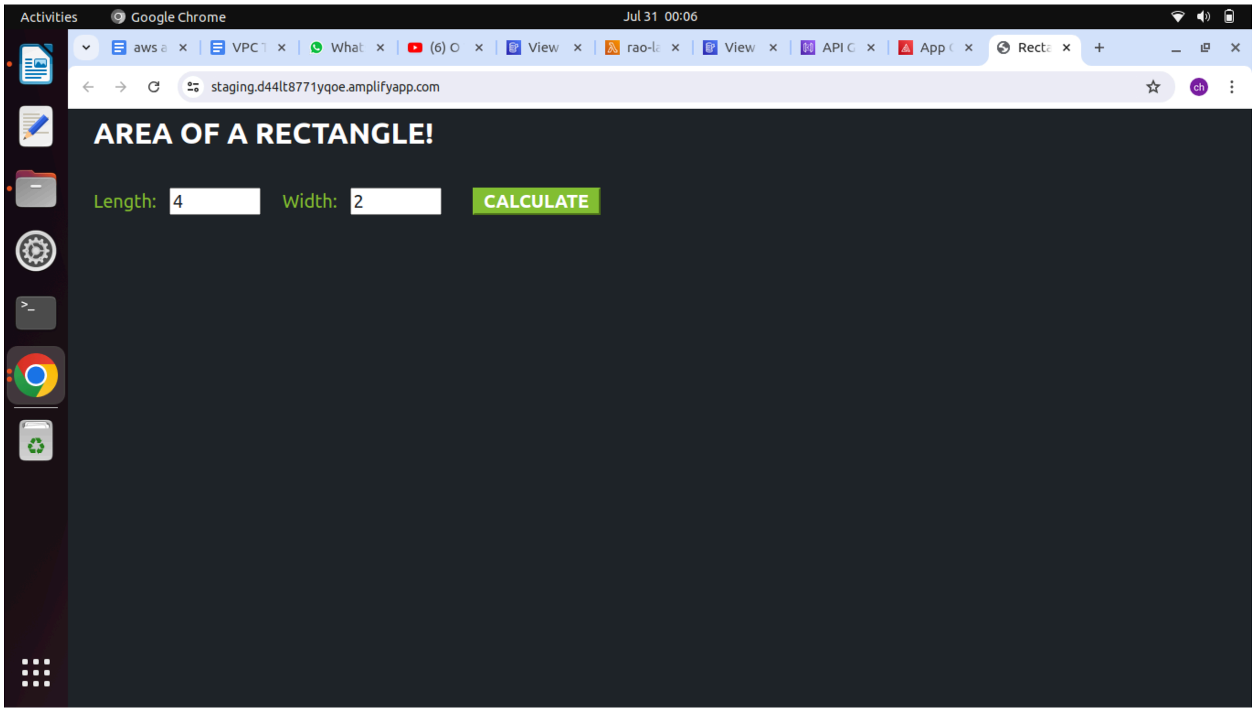
Domain https://staging.d44lt8771yqoe.amplifyapp.com

Updated 31/07/2024, 00:05

Other branches (0)

No other branches added.

Activities
Google Chrome
Jul 31 00:05
us-east-2.console.aws.amazon.com/amplify/apps/d44lt8771yqoe/overview
All apps / app1683 / Overview
Overview
Hosting
App settings
Visit deployed URL
Deploy updates
Add branch
https://staging.d44lt8771yqoe.amplifyapp.com



AREA OF A RECTANGLE!

Length: 4 Width: 2 CALCULATE

54) It take minum half hour to work after go to dynamoDB check items lit update or not

==