**Server-less Web App Development using AWS Amplify, DynamoDB, Lambda and API Gateway**

**High level Diagram:**

A diagram of a software flow

Description automatically generated

**>>** creating a dynamic web app that calculates the area of a rectangle based on user-provided length and width values

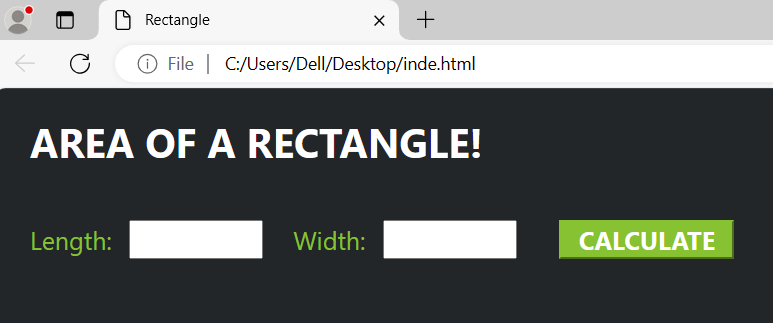
>> AWS Amplify for web hosting, AWS Lambda functions for real-time calculations, DynamoDB for storing and retrieving results, and API Gateway for seamless communication

**\*Creating a front end**

>>copy the below html code into notepad and save it with .html extension

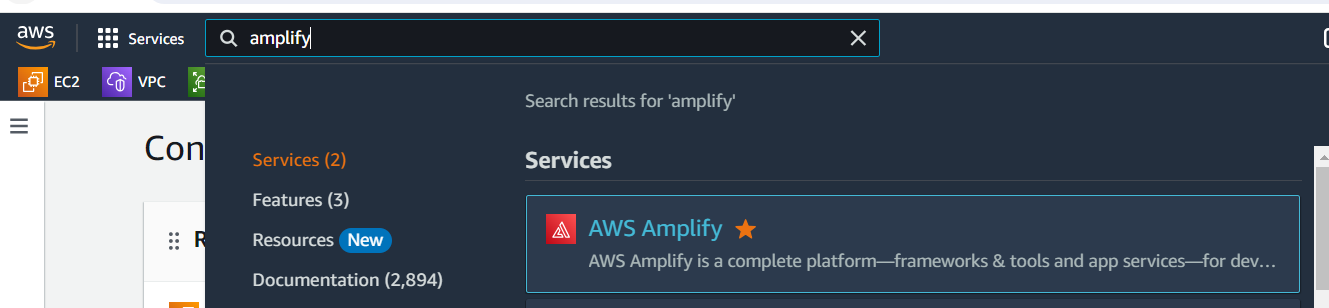
<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>Rectangle</title>  
 <!-- Styling for the client UI -->  
 <style>  
 h1 {  
 color: #FFFFFF;  
 font-family: system-ui;  
 margin-left: 20px;  
 }  
 body {  
 background-color: #222629;  
 }  
 label {  
 color: #86C232;  
 font-family: 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("YOUR API URL", 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>

>>now open the html file and it should look below

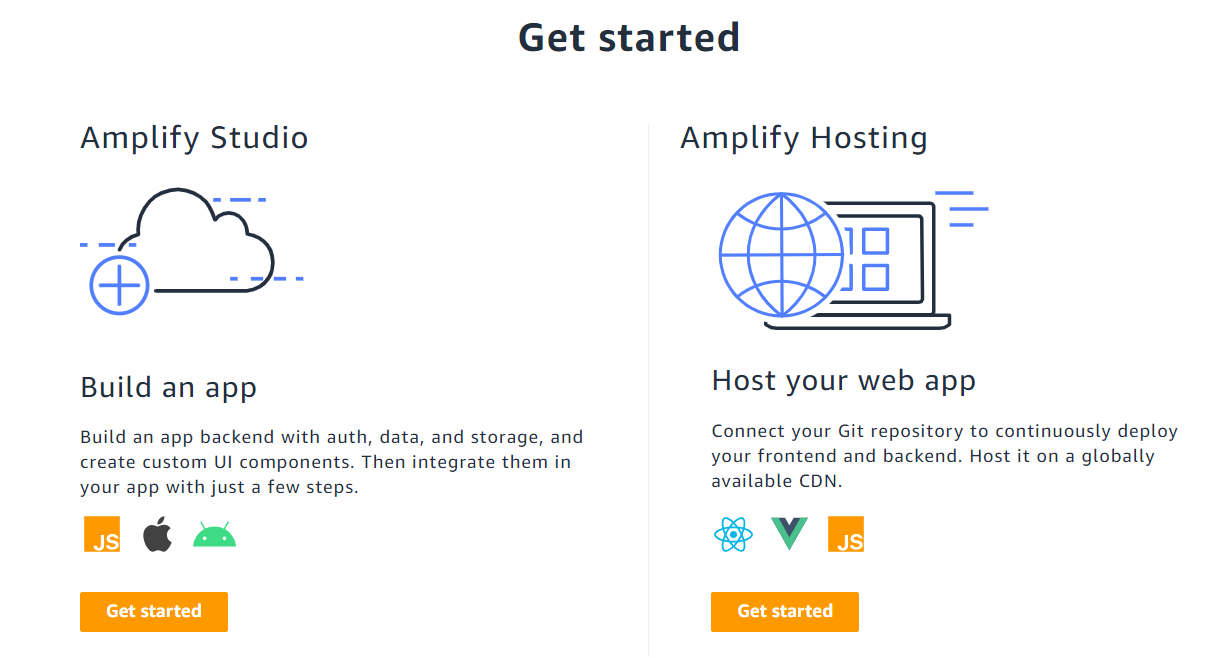


\*Hosting our APP on AWS Amplify

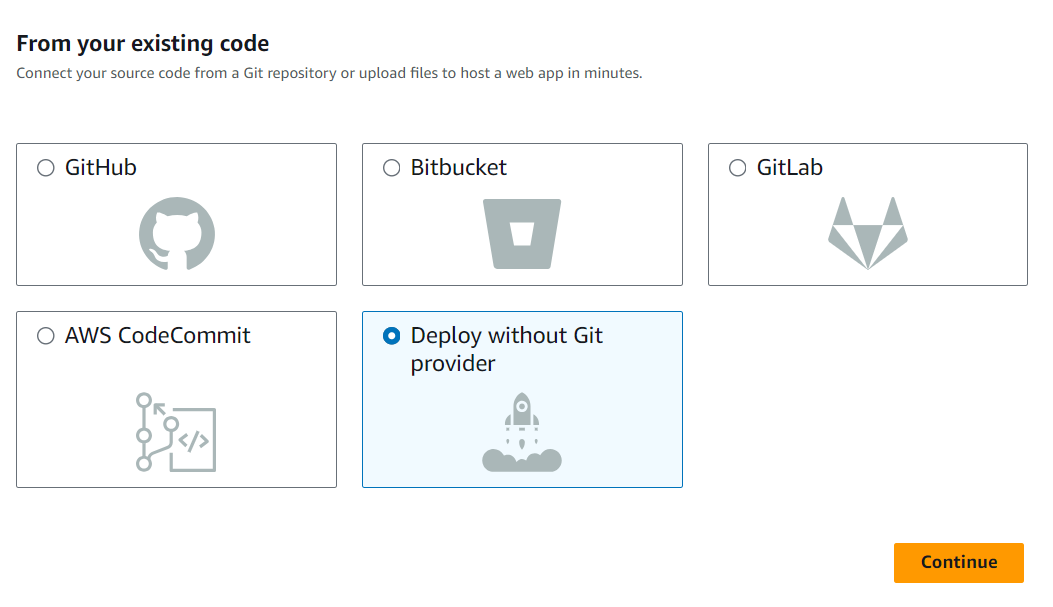
* Search for Amplify in search bar of AWS console and click on AWS Amplify



* Click on Get started and click on Get started for Amplify Hosting to host our Web APP

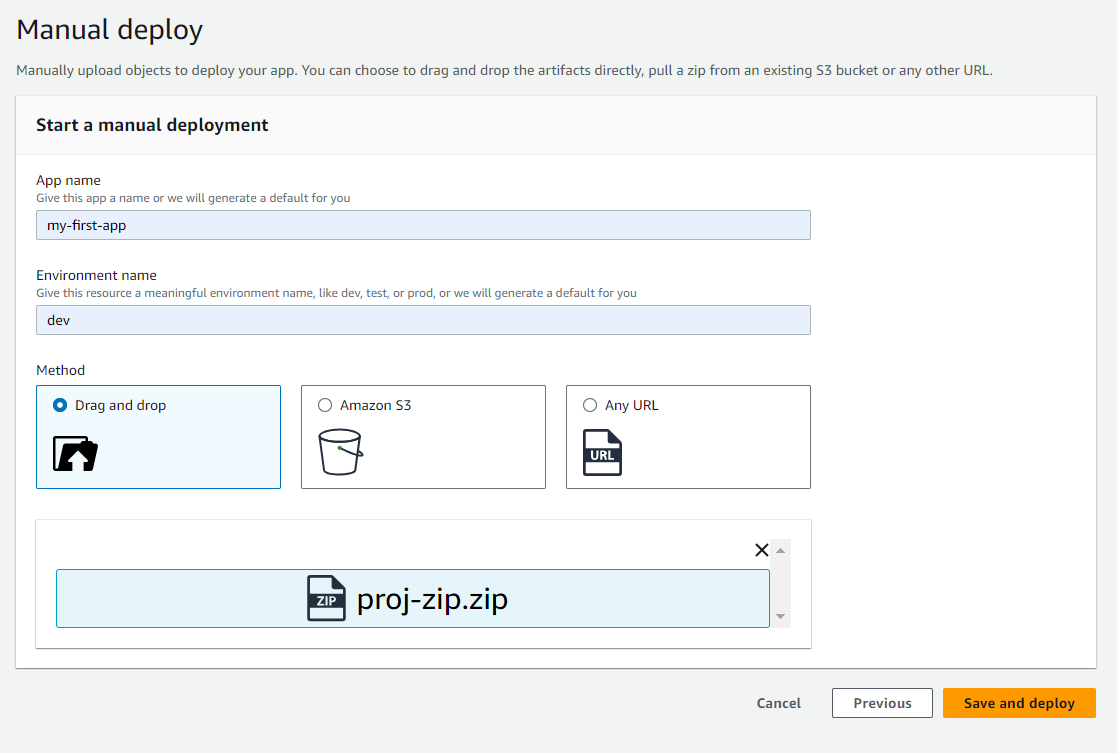


* Select Deploy without GIT provider option and click on Continue

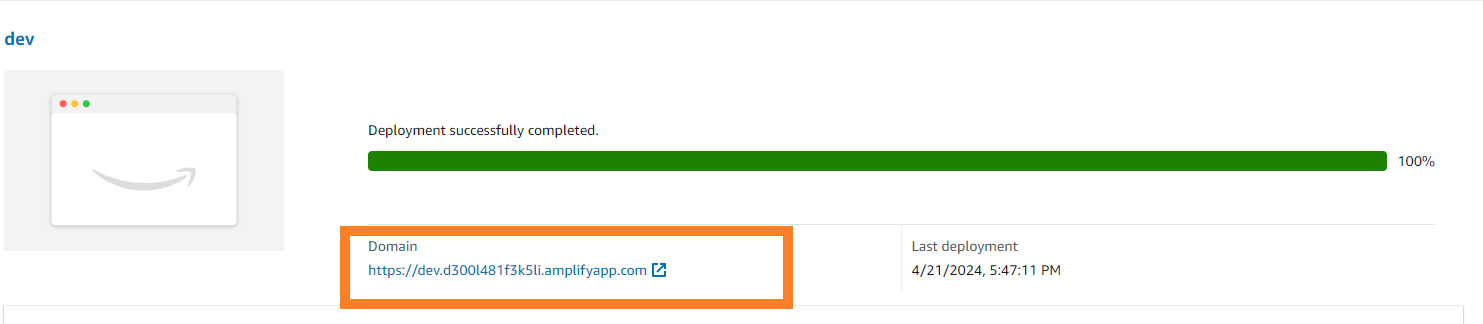


* Give name to the app 🡪 give environment name 🡪select Drag and drop Method 🡪click on Choose files and then upload the zip file 🡪finally click on Save and deploy button

\*note: Amplify will accept zip files only, so Zip our index.html to upload in Amplify



* After deployed, click on Domain link to access the App



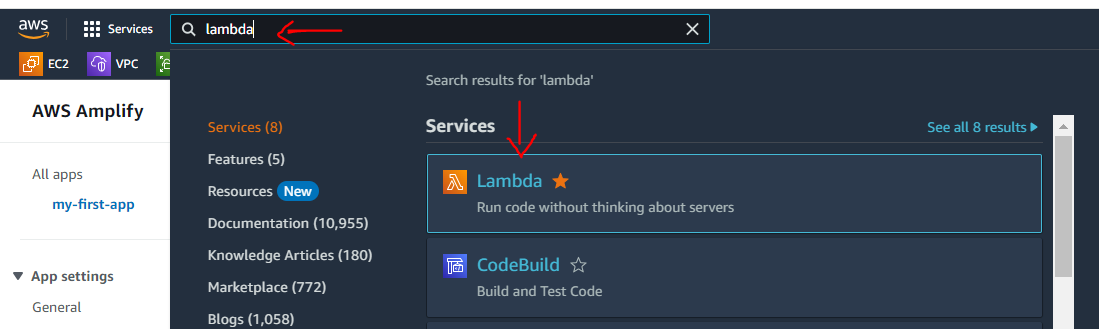
* The app that opened on the browser looks like below



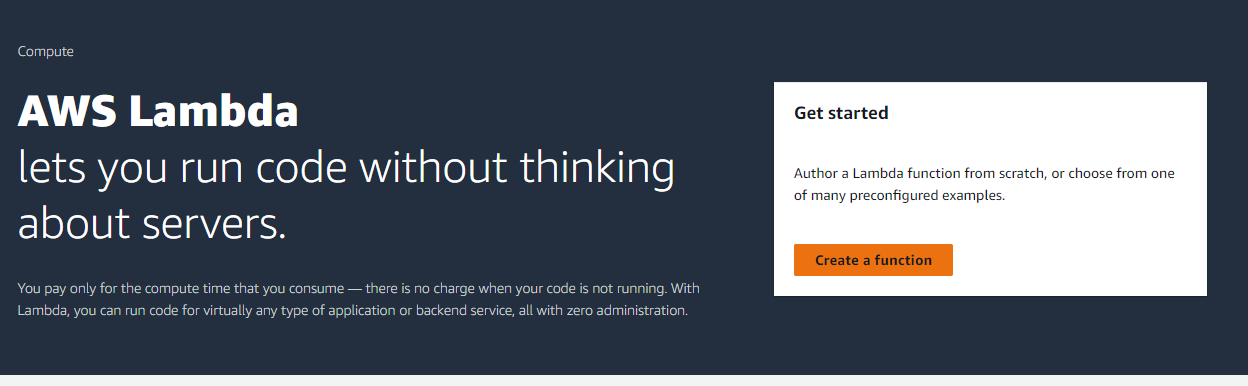
* We are done with first level of our task
* Now create a Lambda function to perform calculation.

**\*Creating Lambda function**

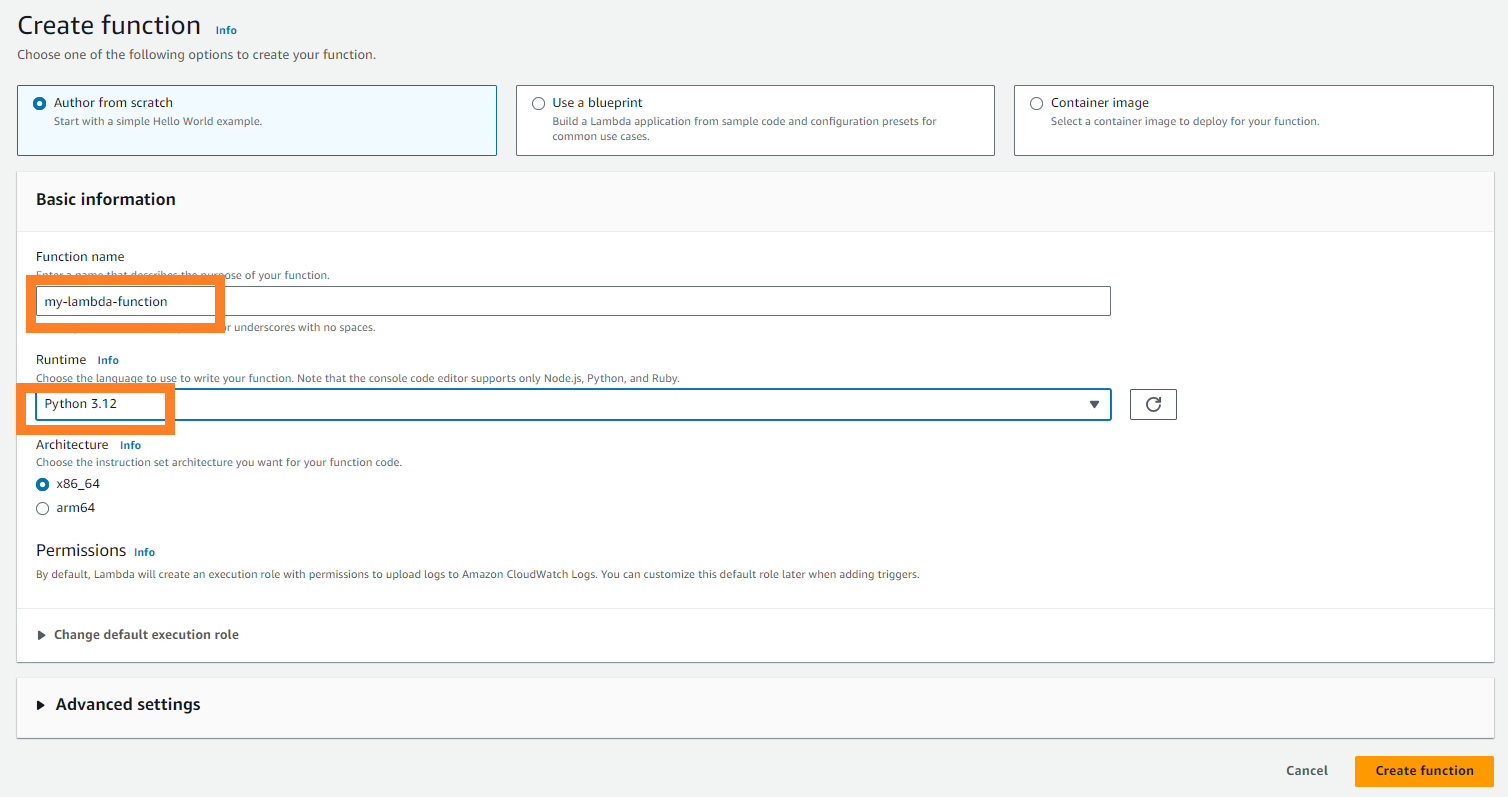
* Search for Lambda in AWS search bar and click on Lambda



* Now click on Create function to crate function



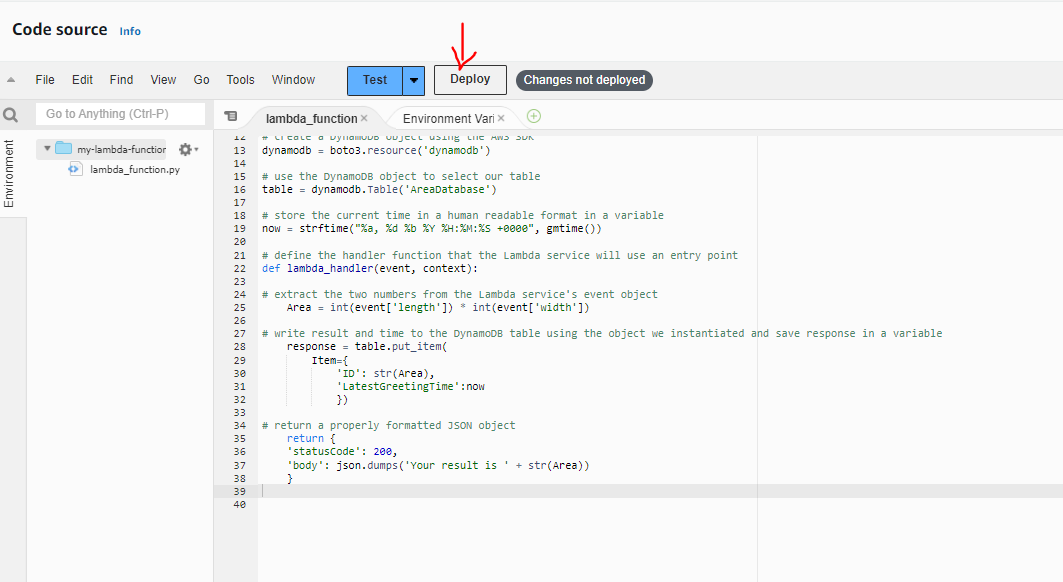
* Give function name and select python (latest version) as Runtime and click on Create function



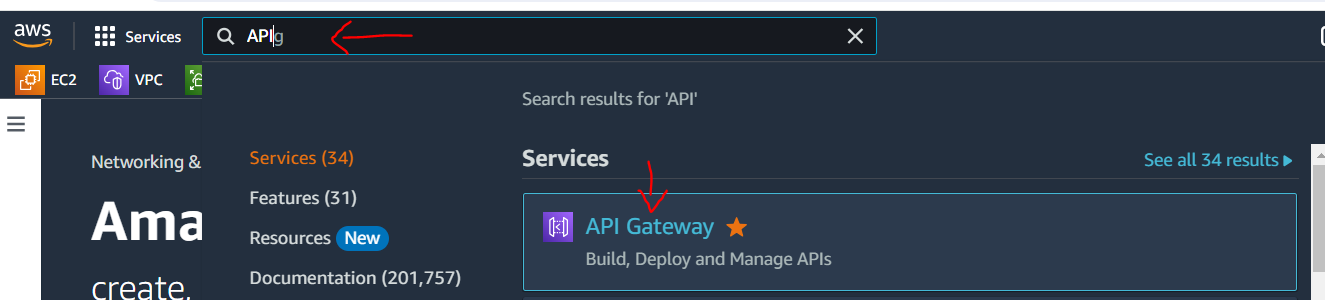
* Copy the below code and paste it on lambda\_function.py file

# import the JSON utility package  
import json  
  
# 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('AreaDatabase')  
  
# 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  
 Area = int(event['length']) \* int(event['width'])  
  
# 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(Area),  
 'LatestGreetingTime':now  
 })  
  
# return a properly formatted JSON object  
 return {  
 'statusCode': 200,  
 'body': json.dumps('Your result is ' + str(Area))  
 }

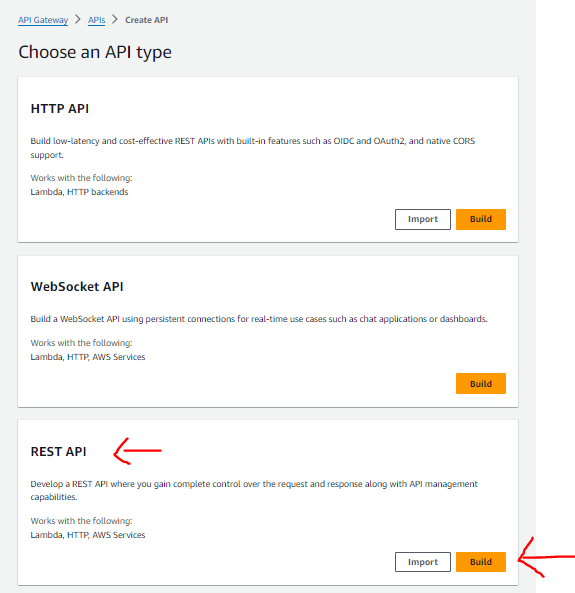
* Now click on Deploy



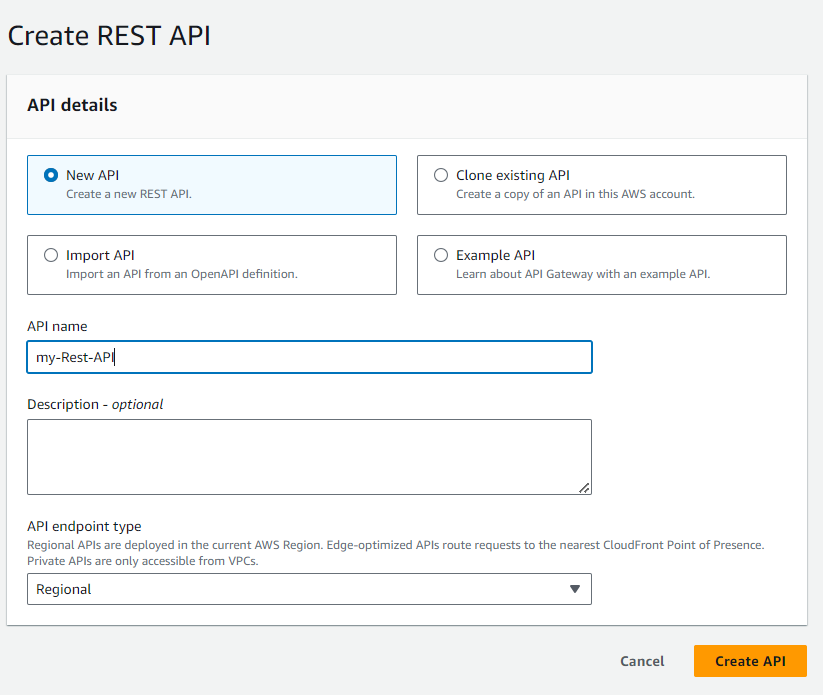
* We are don with Lambda function and now create API gateway
* Search for API and click on API Gateway



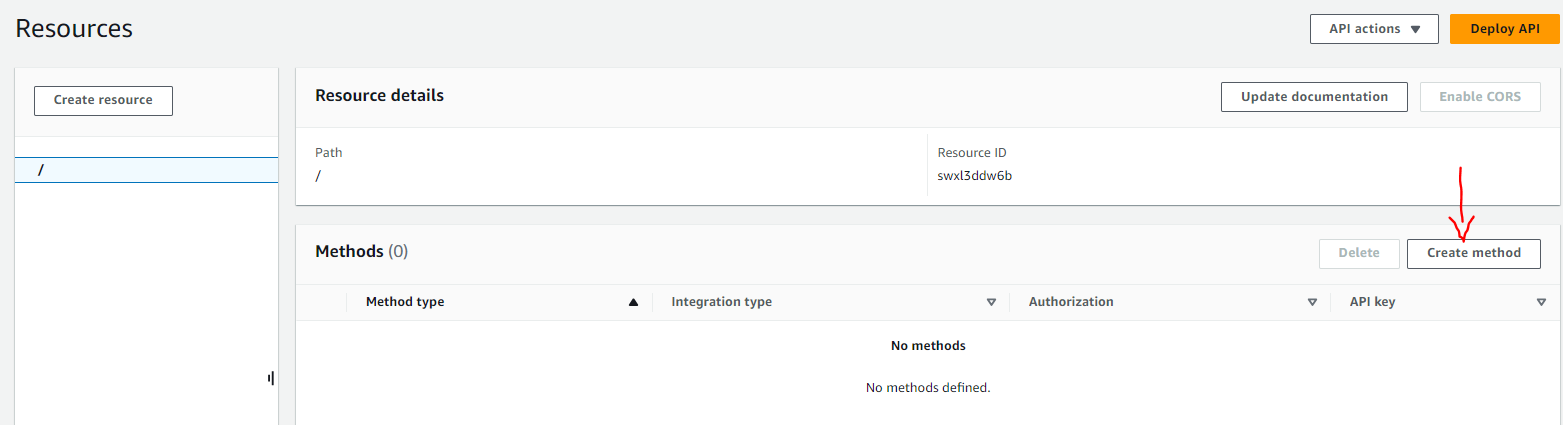
* Form the list for Choose an API type, click on Build for Rest API



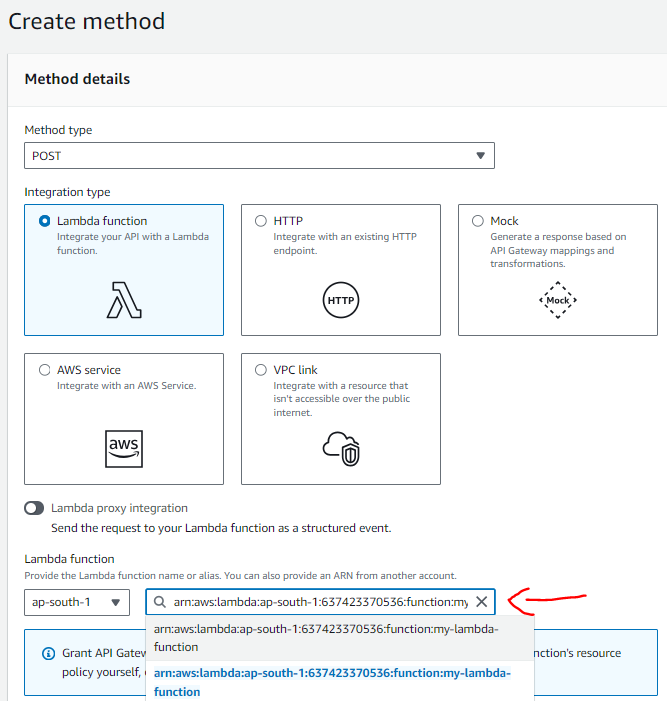
* On creation page, just give the name of API (under API name) and click on Create API

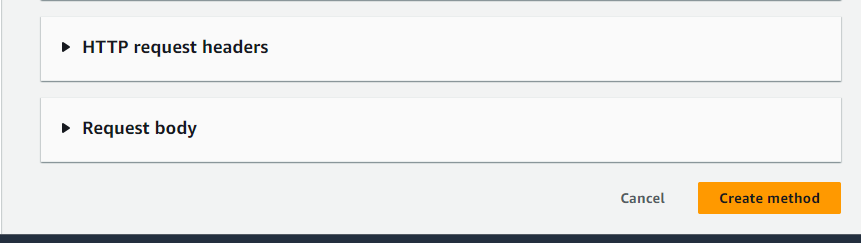


* Now click on Create Method

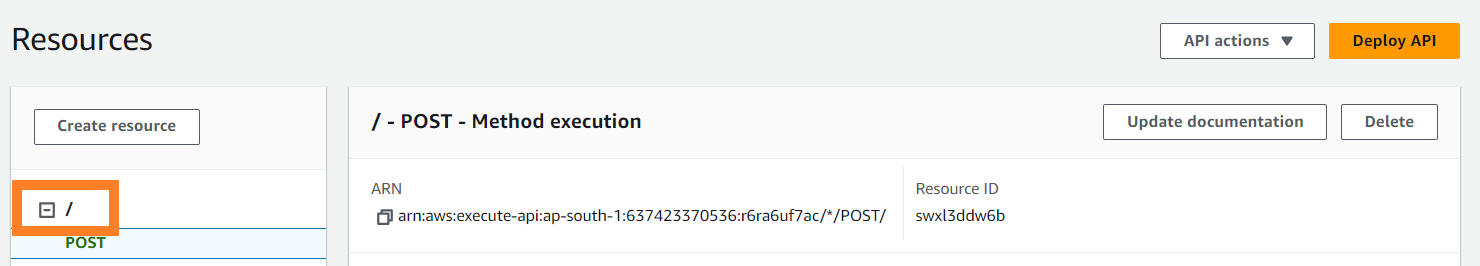


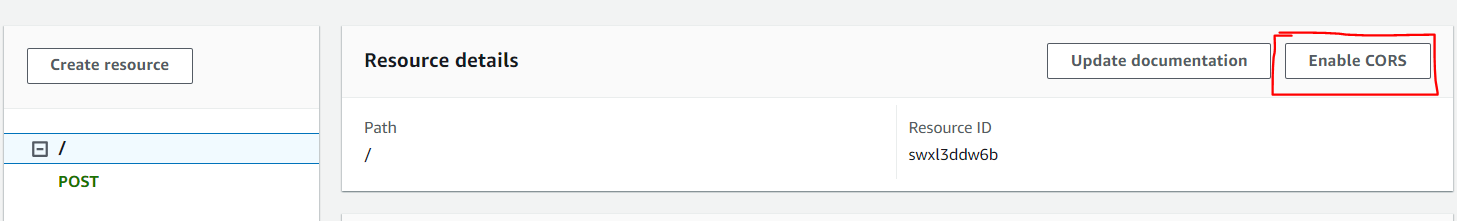
* Under Method type, select POST option from drop down and select Lambda function for Integration type. Under function field, select our Lambda function which we already created. And finally click on Create Method



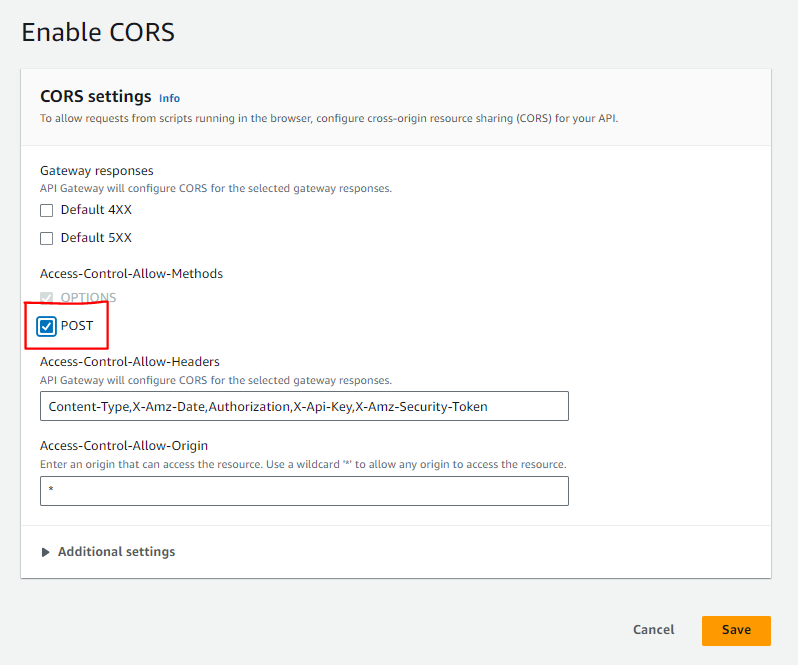


* Now click on / (slash) that appears under Create resources and click on Enable CORS

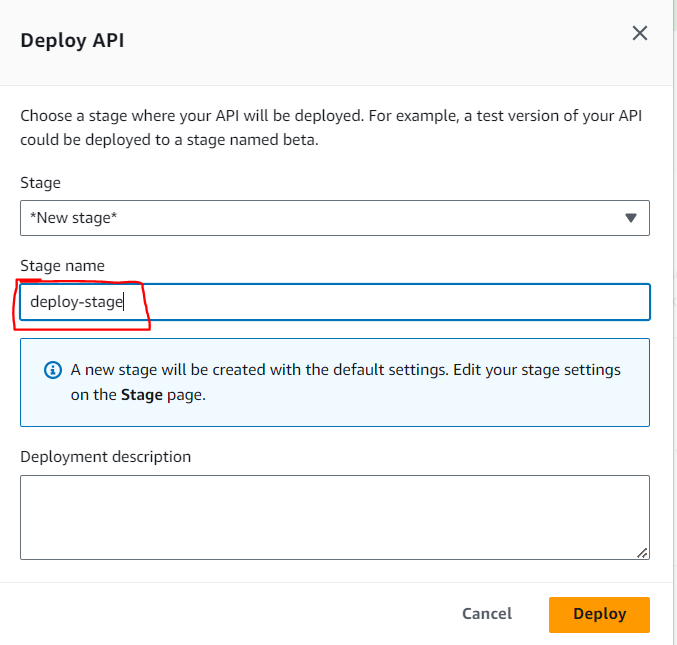




* Select POST method, click on Save



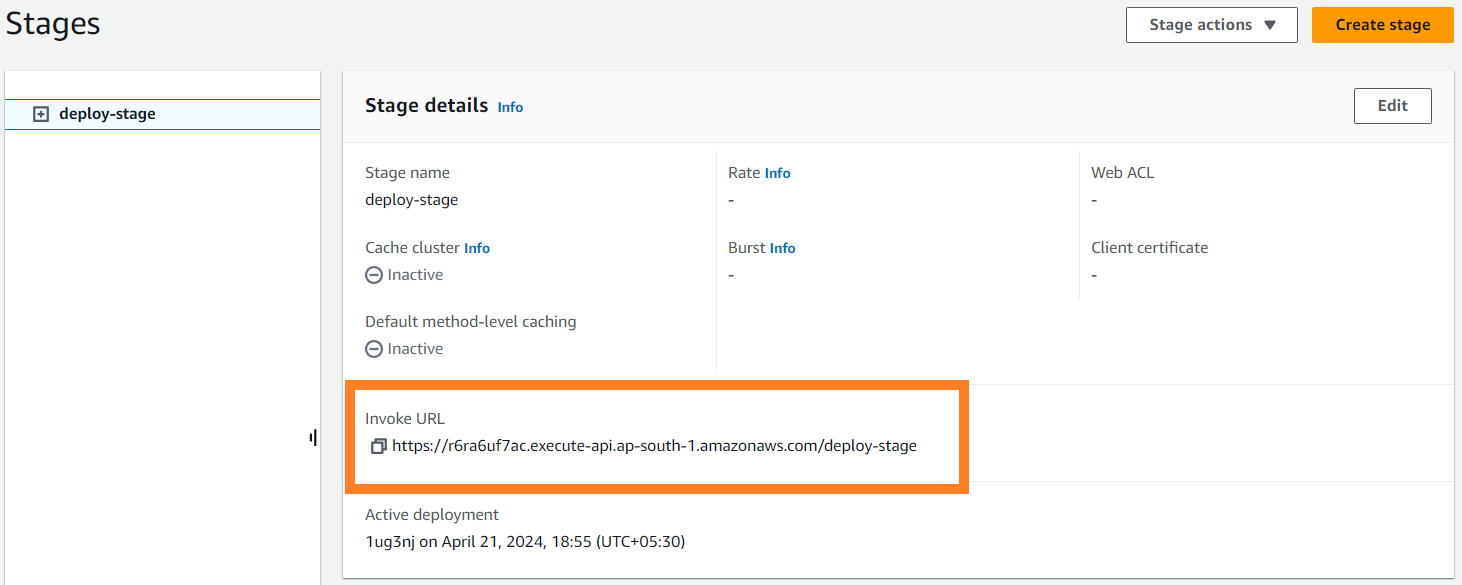
* Click on Deploy API. Now Deploy API page will appear, here select New stage under Stage field from drop down and give stage name and click on Deploy



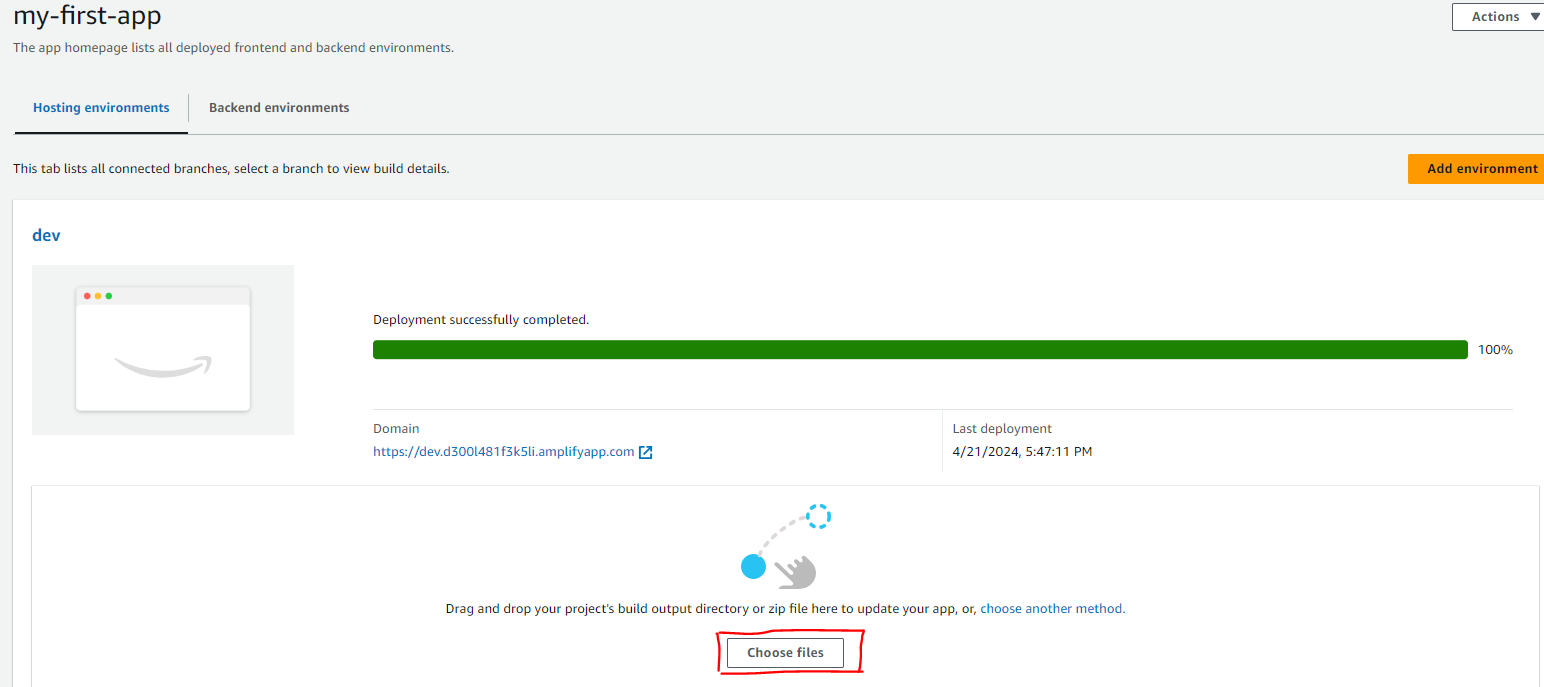
* Now we will get an Invoke URL, copy that URL and paste it in the index.html file, where you can see YOUR API URL. Save it with index.html and make a zip of it and re upload it to Amplify APP.

🡪in index.html, we can see --fetch("YOUR API URL", requestOptions)

🡪instead of YOUR API URL, paste the invoke URL. Should something look like below



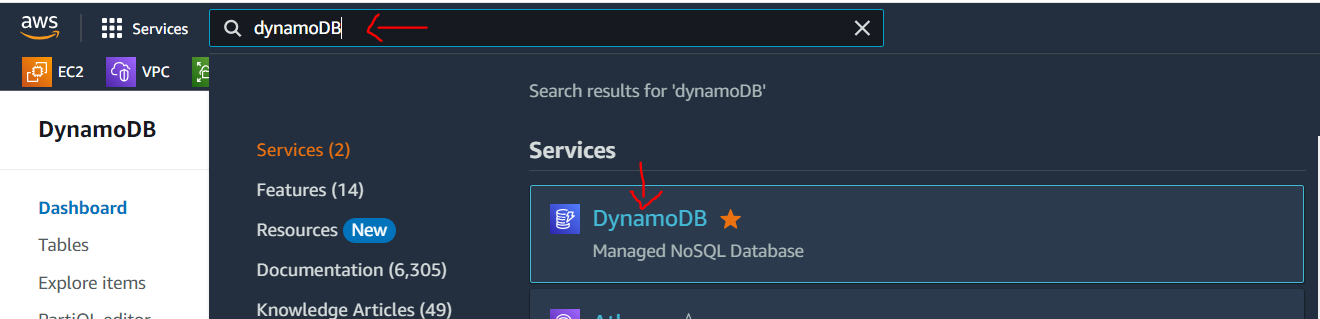
* To re-upload, go to Amplify page, click on Choose files option and then upload the zip file. (project zip file)

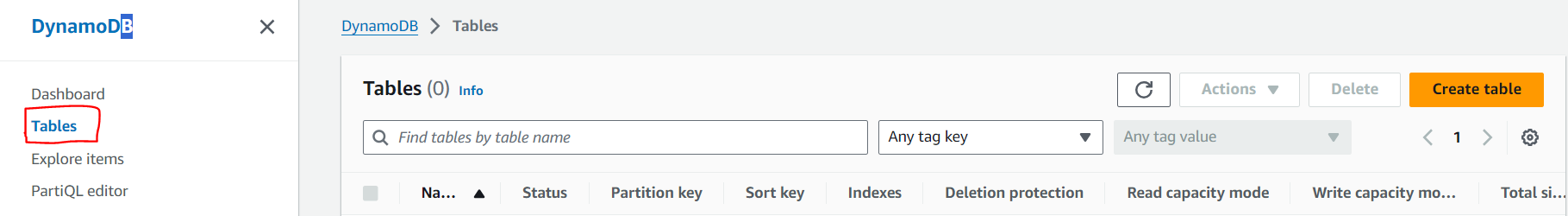


* Now create DynamoDB to store results

**\*DynamoDB**

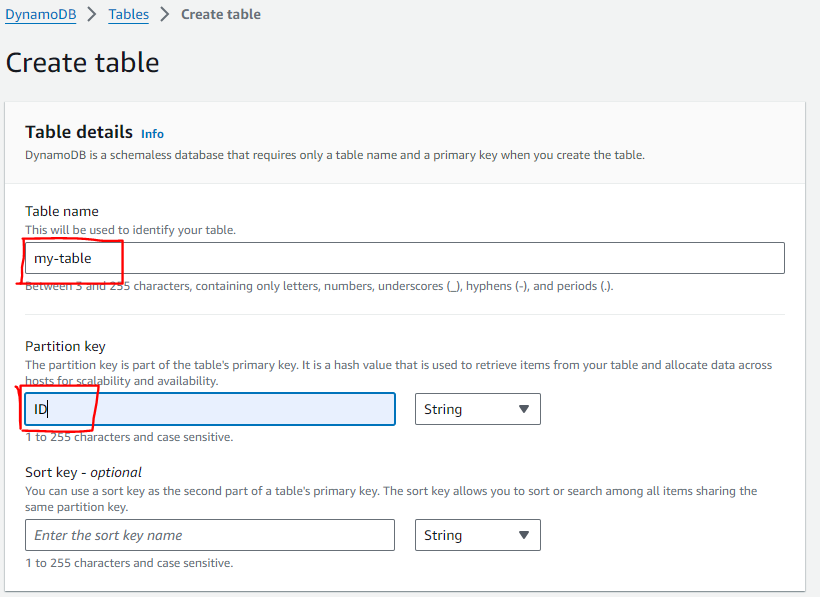
* Search for DynamoDB and click on DynamoDB
* Click on Tables option and click on Create table

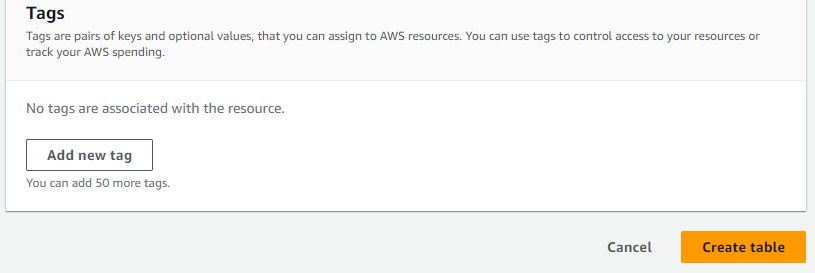




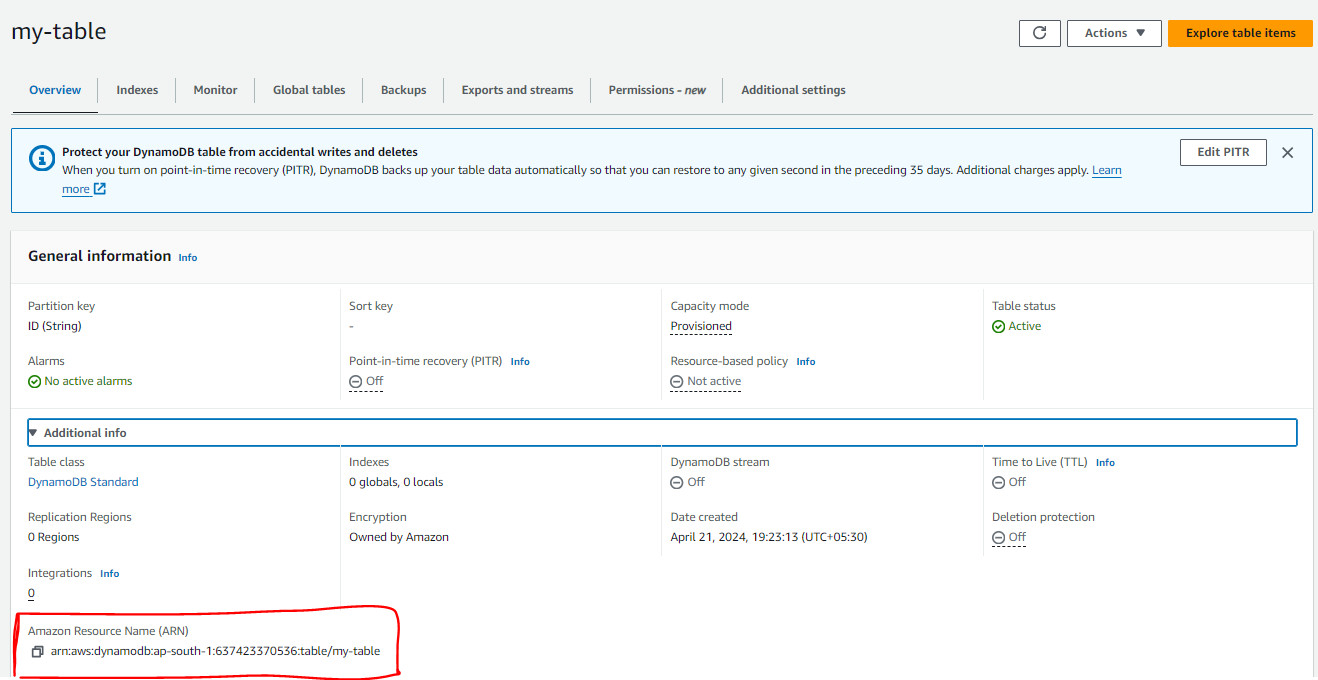
* On table creation page, give name of the table and give ID as partion key. Scroll down and click on Create table

\*Note: Please do remember table name that we have given here, as it is required to give the same in the lambda function code to access the table.

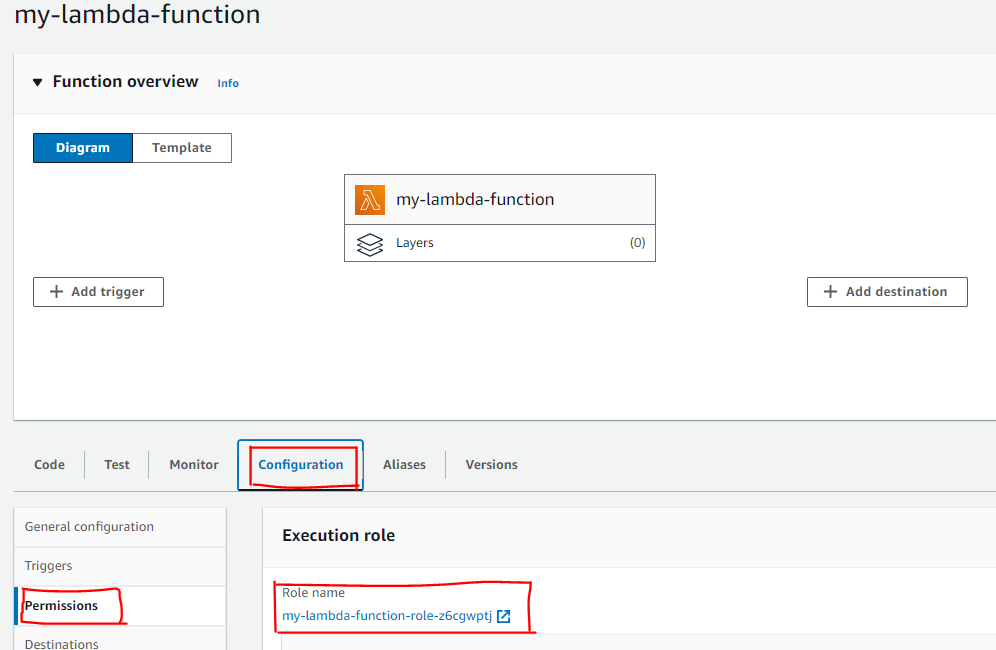




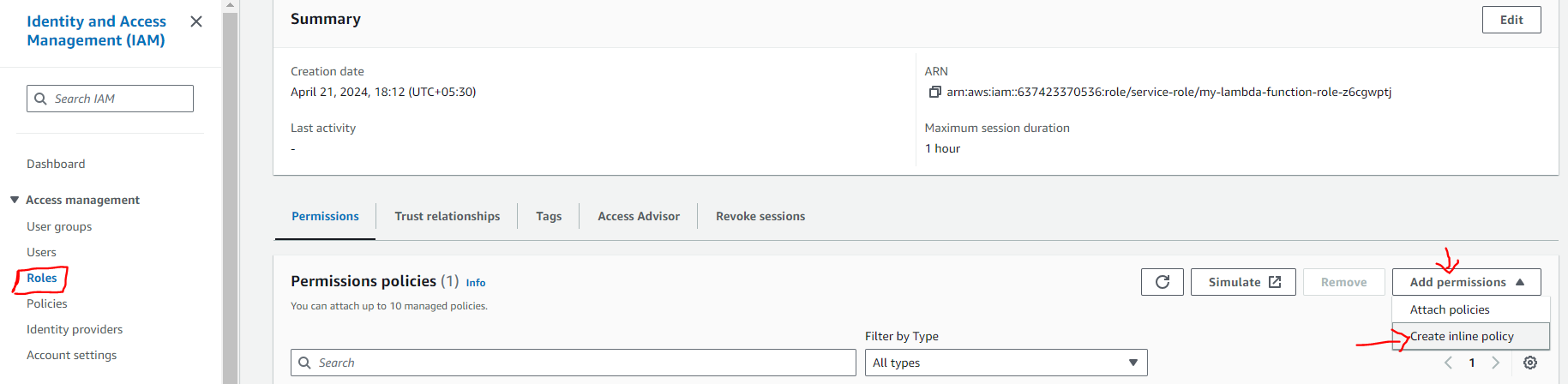
* After table created, click on our table, under General information, explore the Additional information option and take a note the ARN (copy and paste it somewhere)



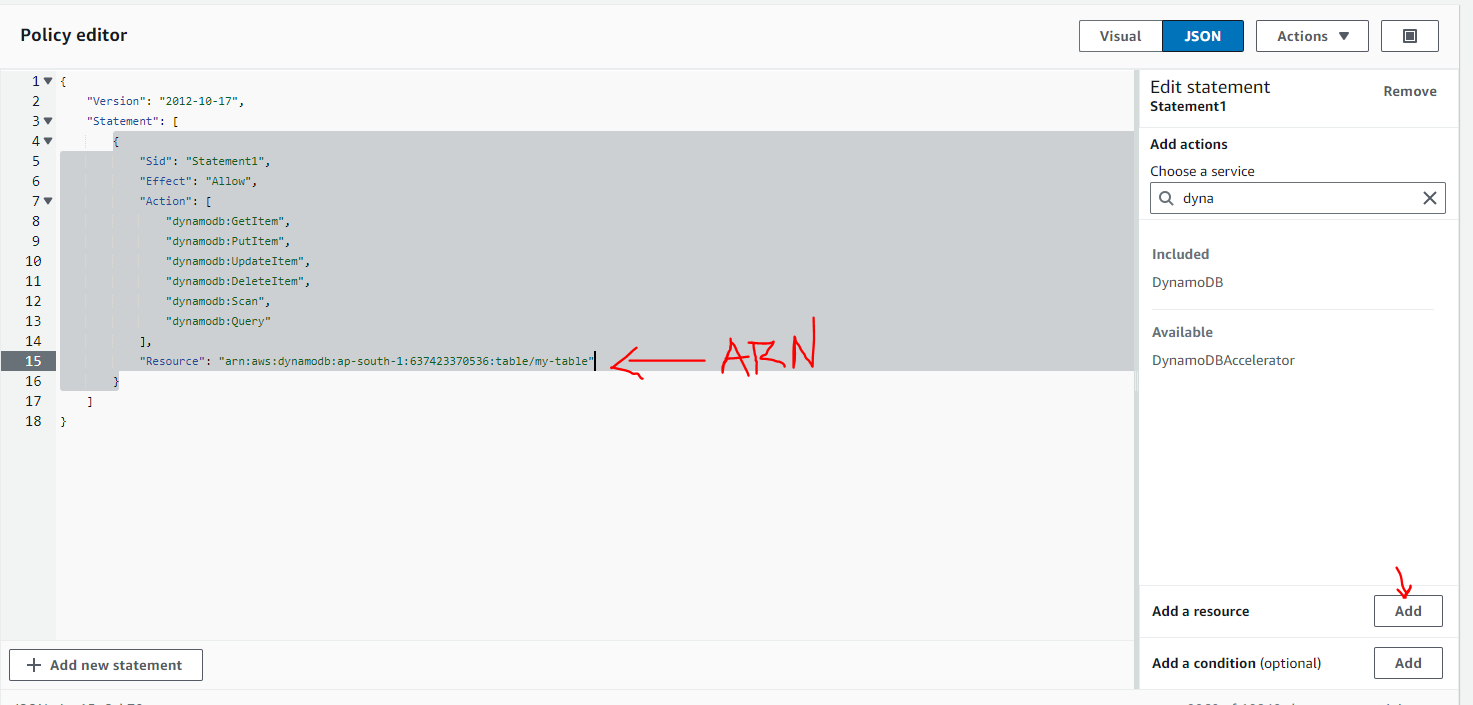
* Its time to add permissions to our Lambda function to Access the DynamoDB
* Go to Lambda function page, click on Configuration tab then click on Permissions. And click on the link under Role name, it will open new IAM tab.



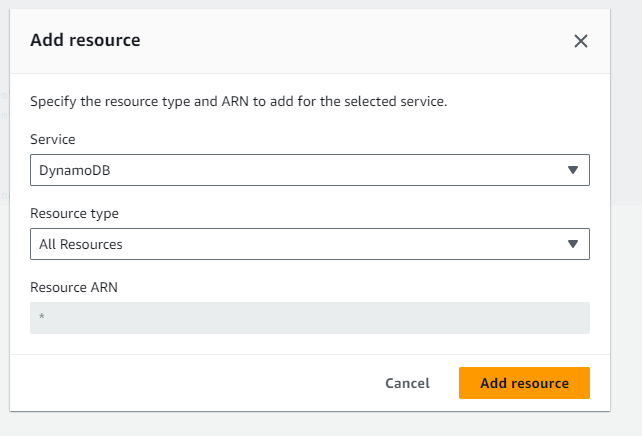
* In new IAM tab, click on Roles, click on Add permissions and then click on Create inline policy



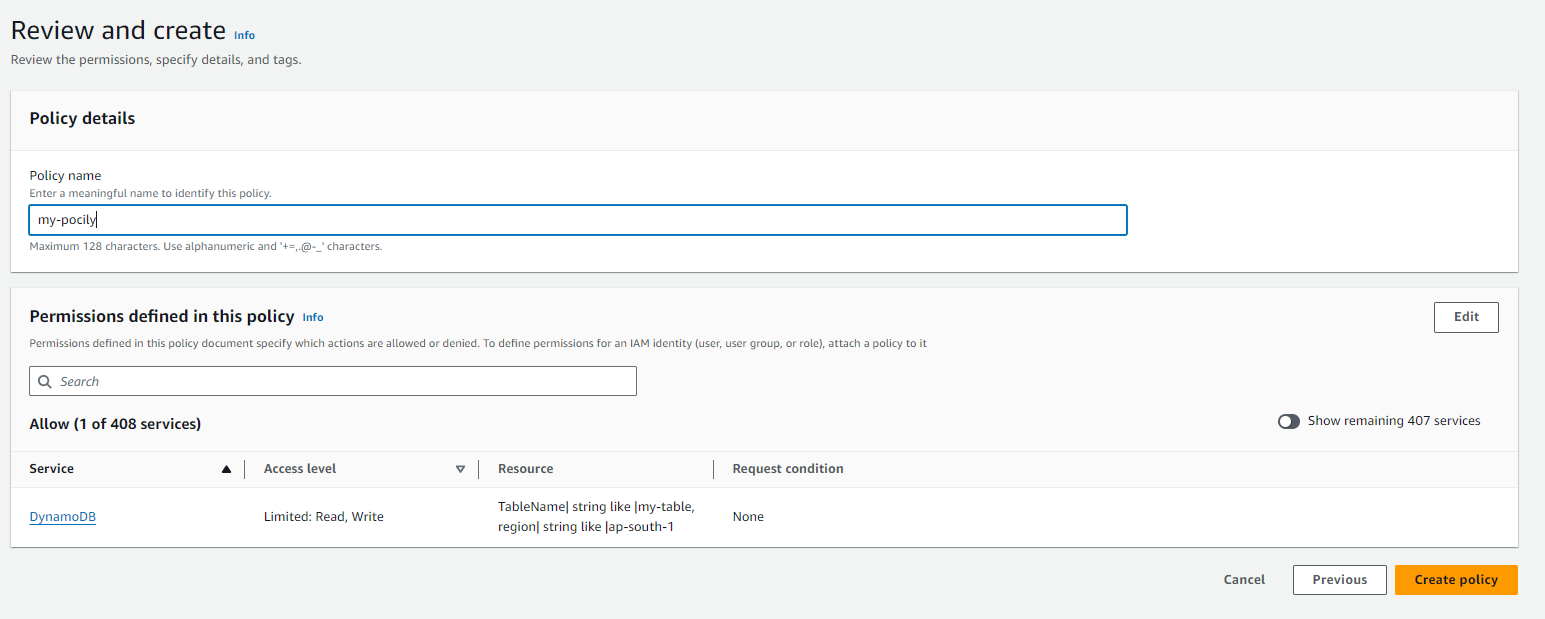
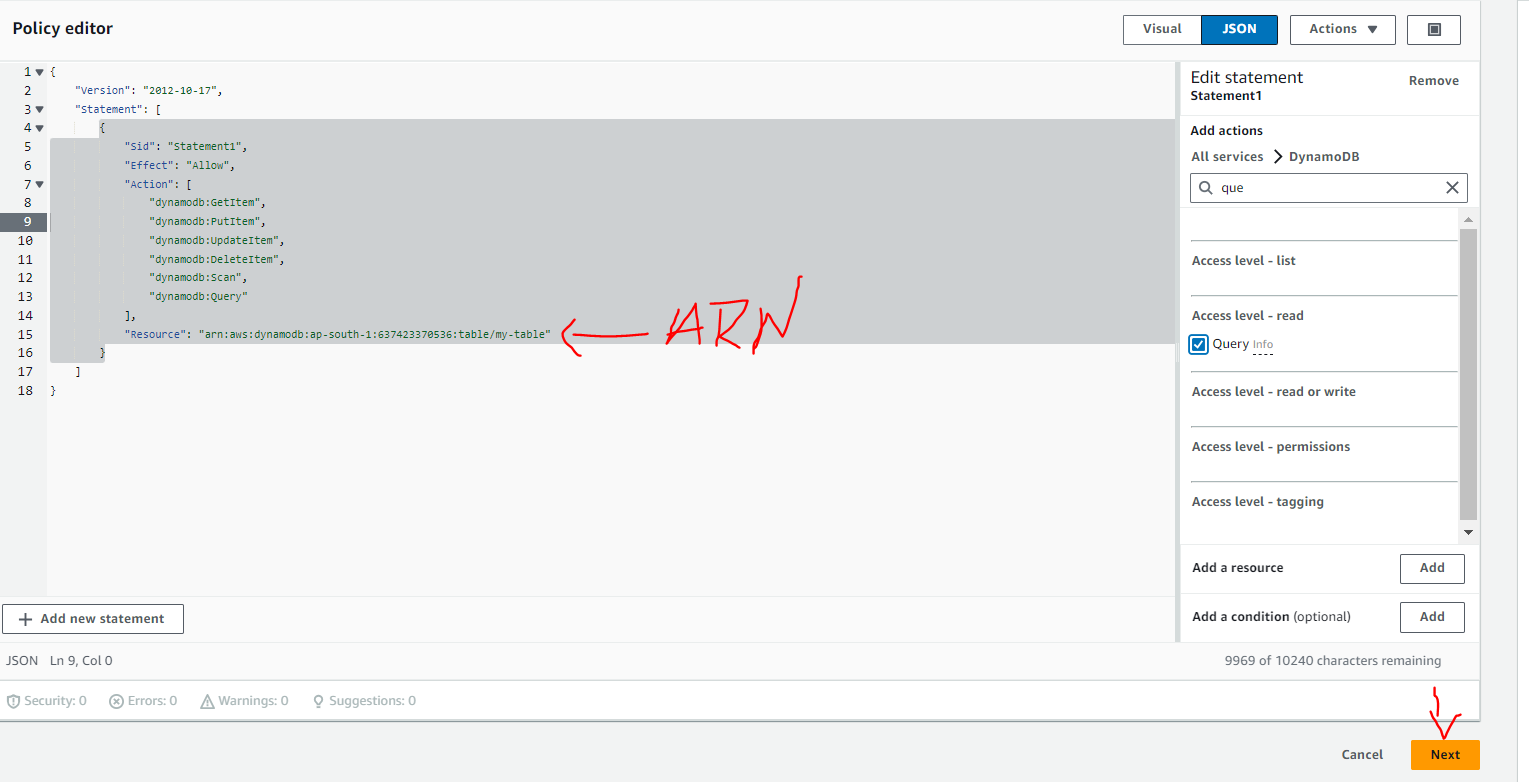
* Change permission mode to JSON, under Choose a service, search Dynamodb and select it. And give the permissions we want.
* And important point here is, give the ARN here for Resource and click on Add resource



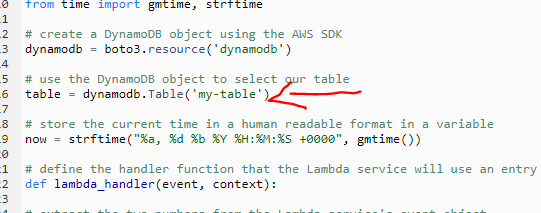
* Select DynamoDB as service and All resources under Resource type and click on Add resource



* Click on Next and give name to policy and click on Create policy



* Finally do one thing, go to lambda function and go to our source code and change the table name with the table name that you given at the time of creating data base table (I have give table name as my-table)



\*now test the app

* Go to Amplify page, click on Domain link to open the app. Give values of your choice and click on Calculate button, it will give the results

