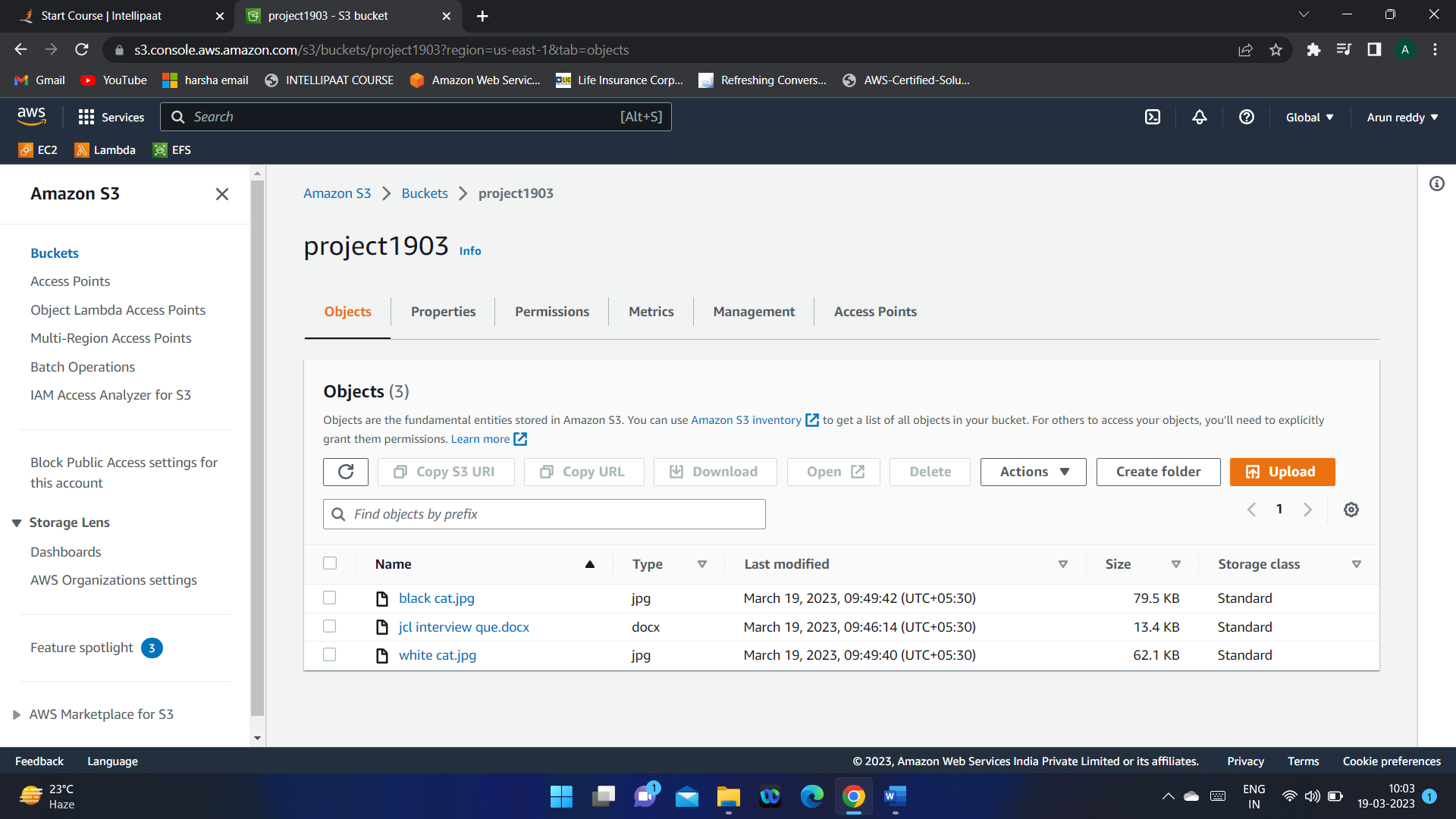
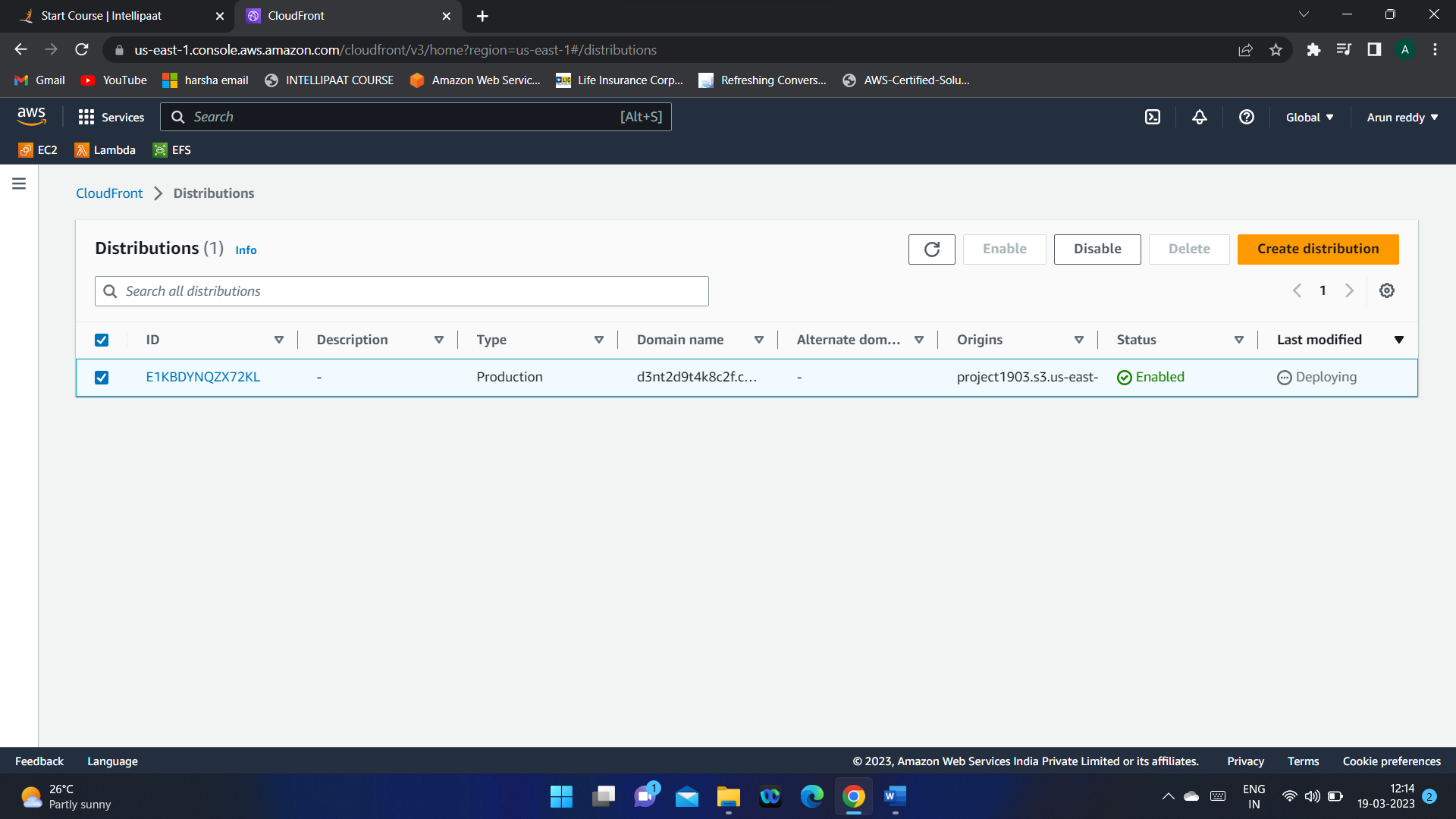
**1.CLOUDFRONT PROJECT**

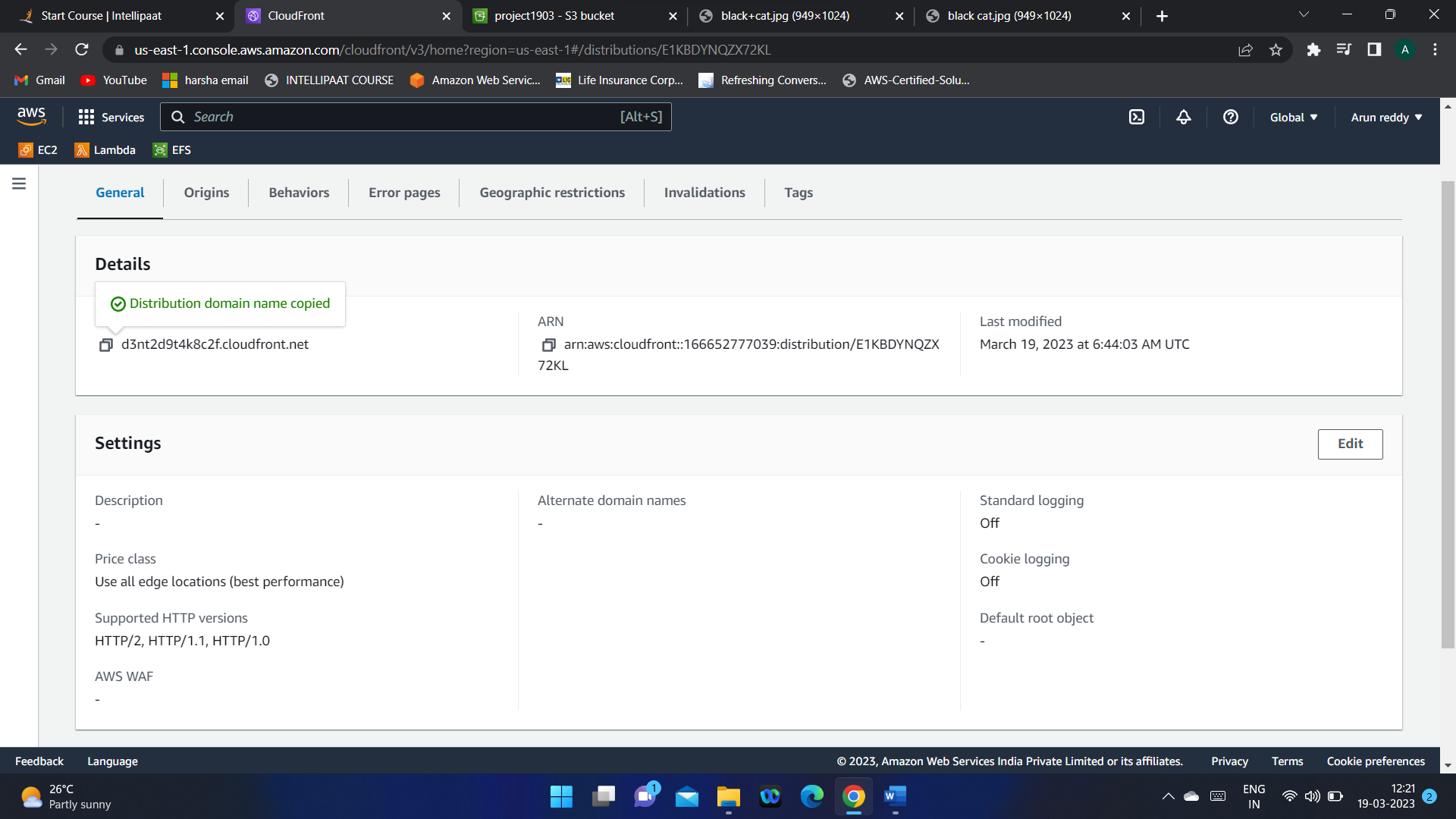
1.create an S3 bucket and upload objects in it , give public access to bucket and objects



2.Create Cloud front distribution



3.use the distribution domain name URL to access the object (in the url give **“/object name**”)



4.if you update the object in S3 bucket , it will take some time to reflect the new image in DDN url for others. But u can view the new object in object URL

5.the new image will reflect after TTL time

6.If you want to change the memory of cache server before TTL you can do that in Invalidations.

**2.Lambda function project**

**Hosting a static website that renders “Hello World”. Then we will add functionality to the web app so the text that displays is based on custom input you provide.**

**I)Create Web app**

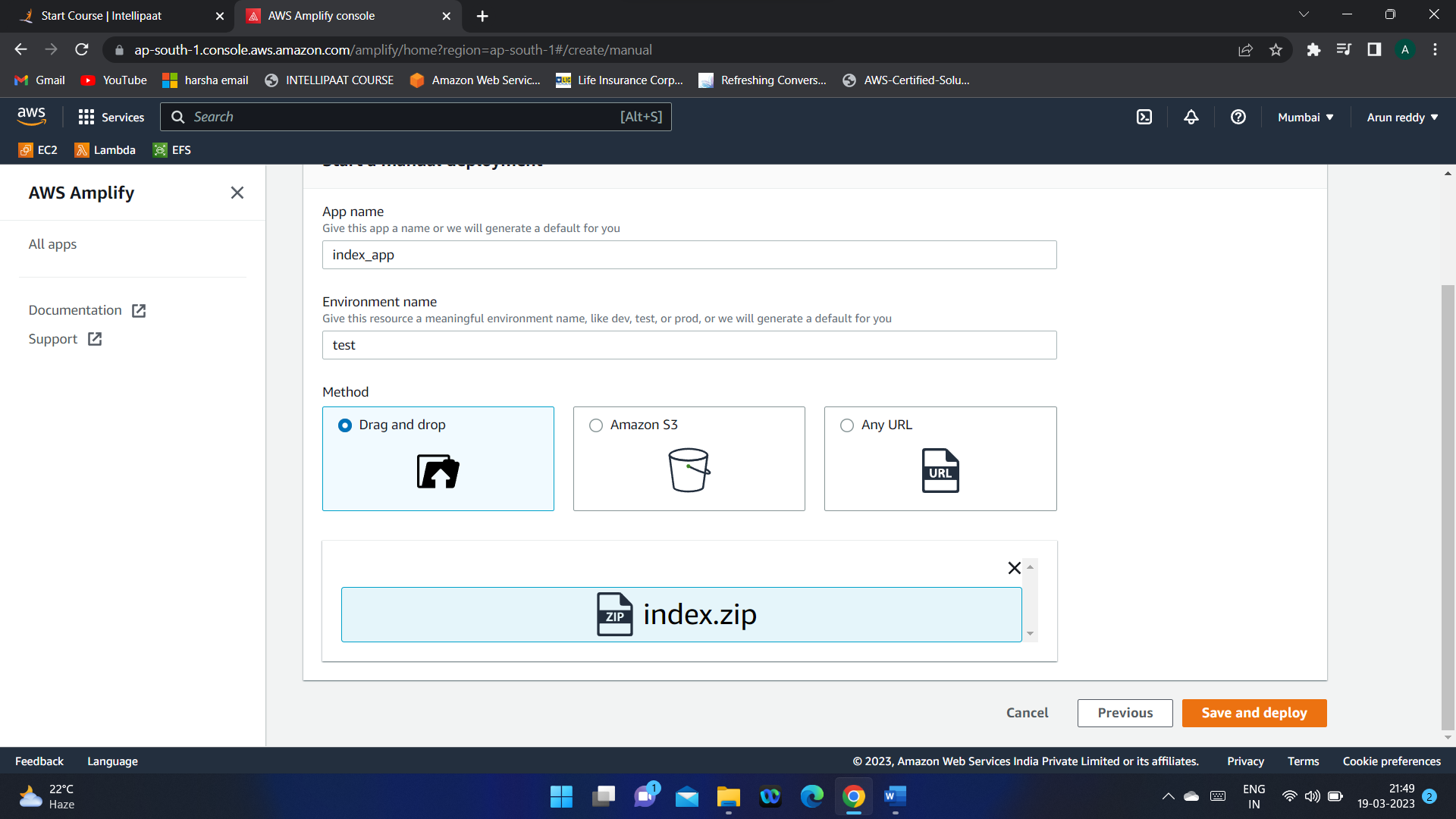
**II) Build serverless Function**

**III)link lambda function to app**

**IV) Create Data Table**

**V)Add interactivity to web app**

**I)1.**In AWS Amplify service drag and drop your code in a zip file

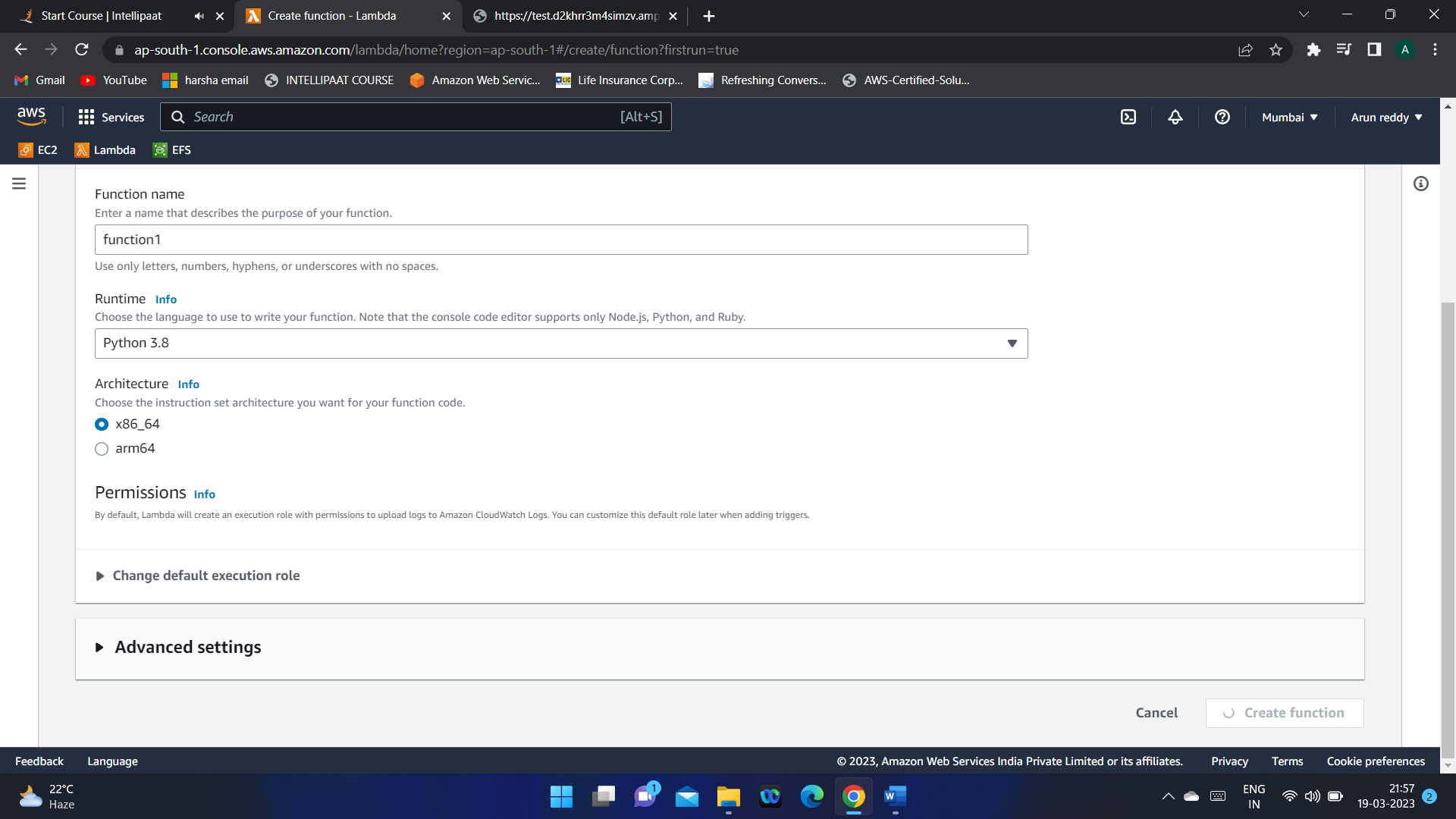
****

2. By clicking on the domain URL you can access the website

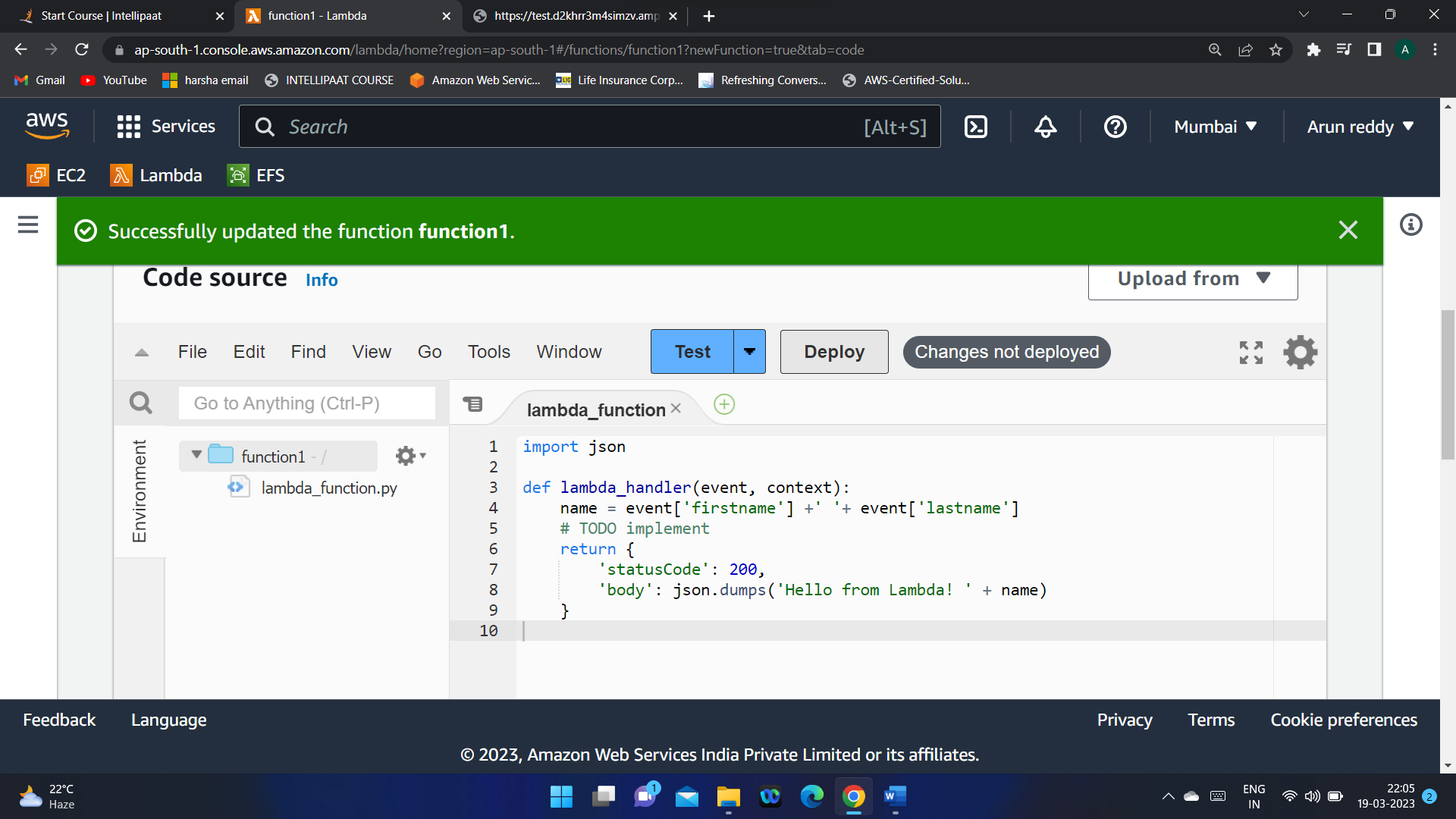
**II)Create lambda function**

**In this module we will create a Lambda function in AWS console and create (JSON) events**

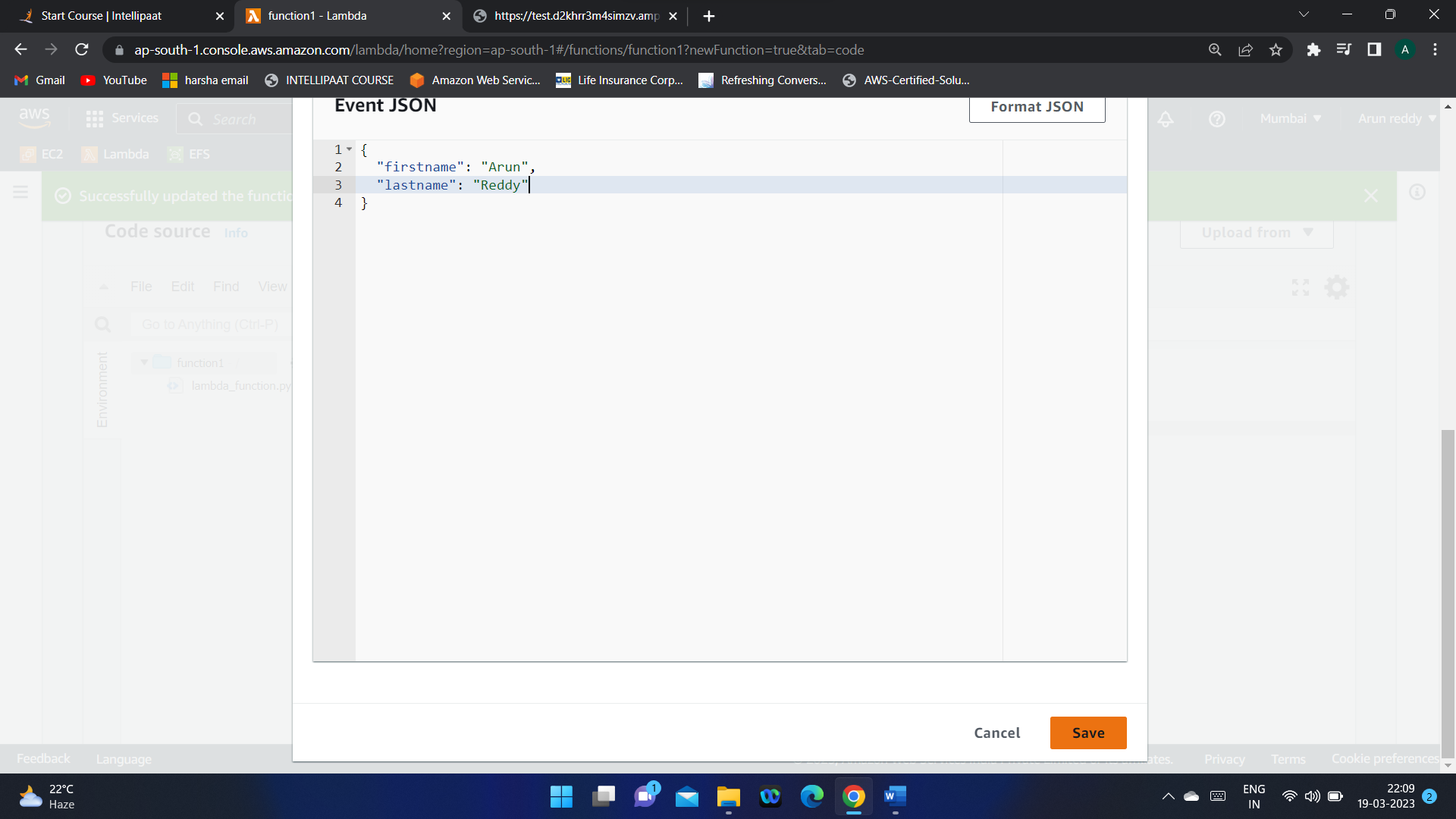
**In AWS console to test your function**

****

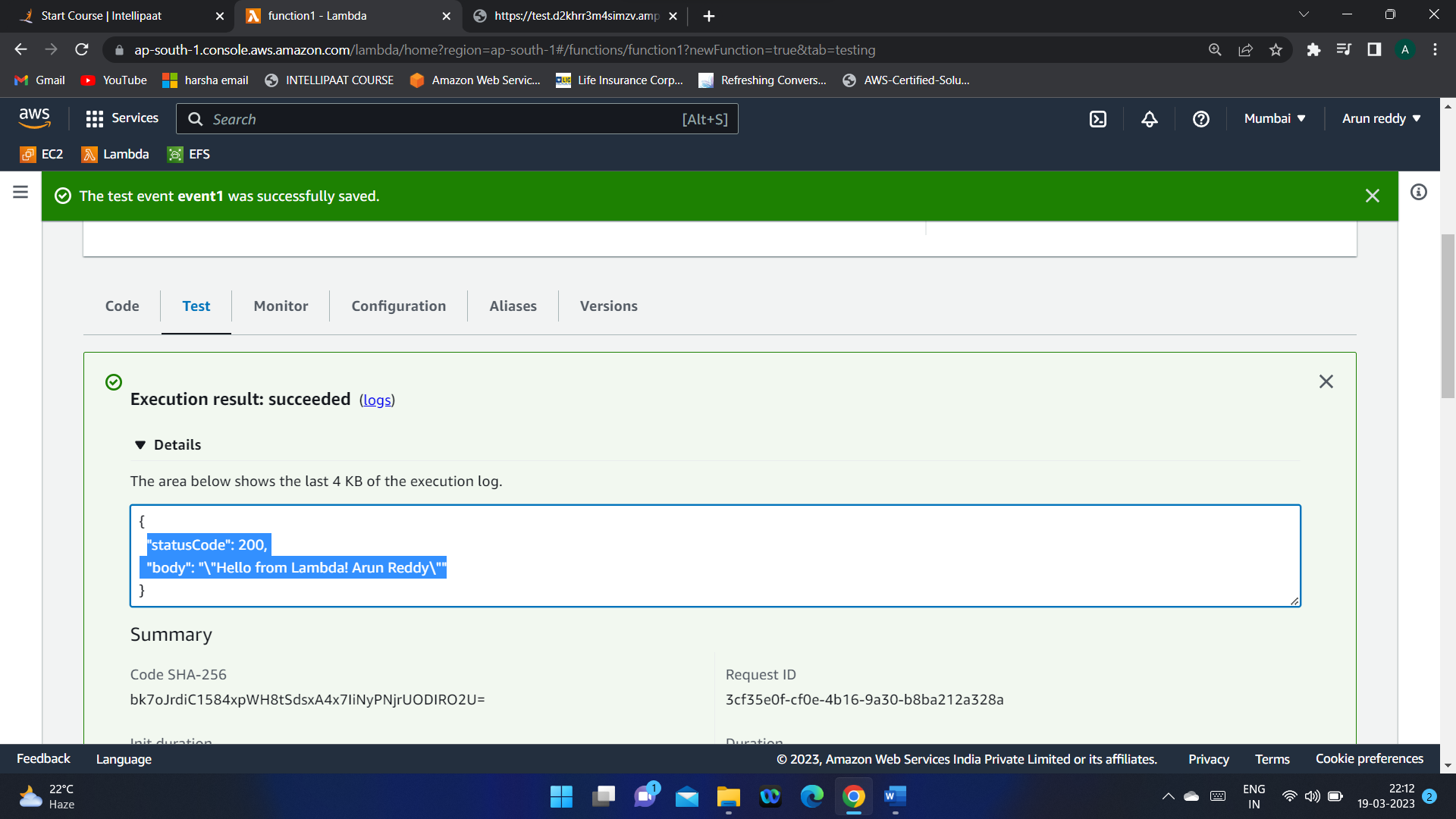
**2.write the code and deploy it**

****

**Test the code by giving the event variables in json file :**

****

**You can see the test results :**

****

**Now we have a working Lambda function**

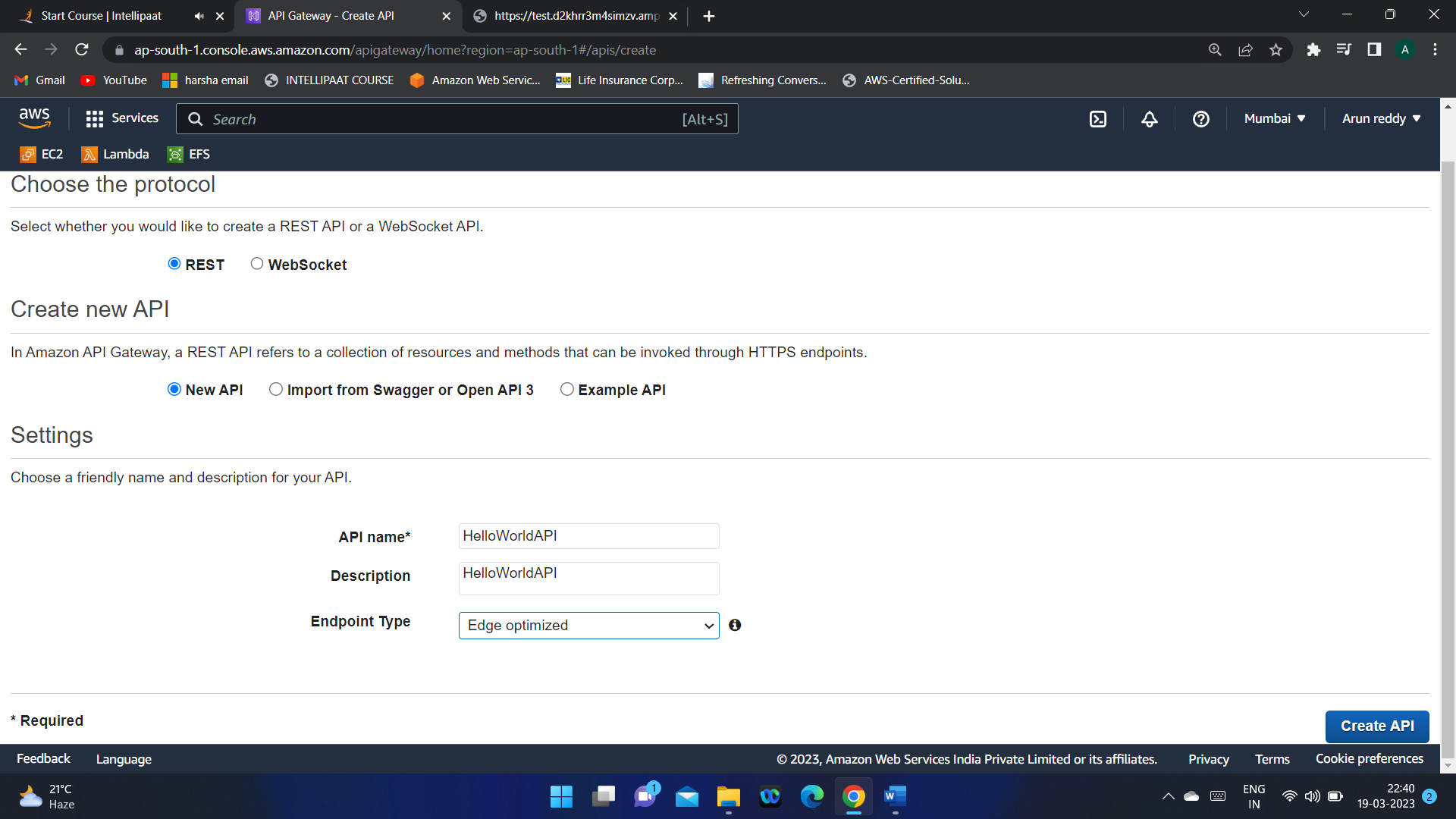
**III)API Gateway gives connection b/w the lambda and the app**

**API is like a waiter in hotel, acts like a mediator between customer and chef.**

**In this module,**

* **We will create an API**
* **Define HTTP methods on your API**
* **Trigger Lambda function from an api**
* **Enable Cross region resource sharing on an api, so you can consume resources from a different origin(domain)**
* **Test an API created with API GATEWAY from AWS management console.**

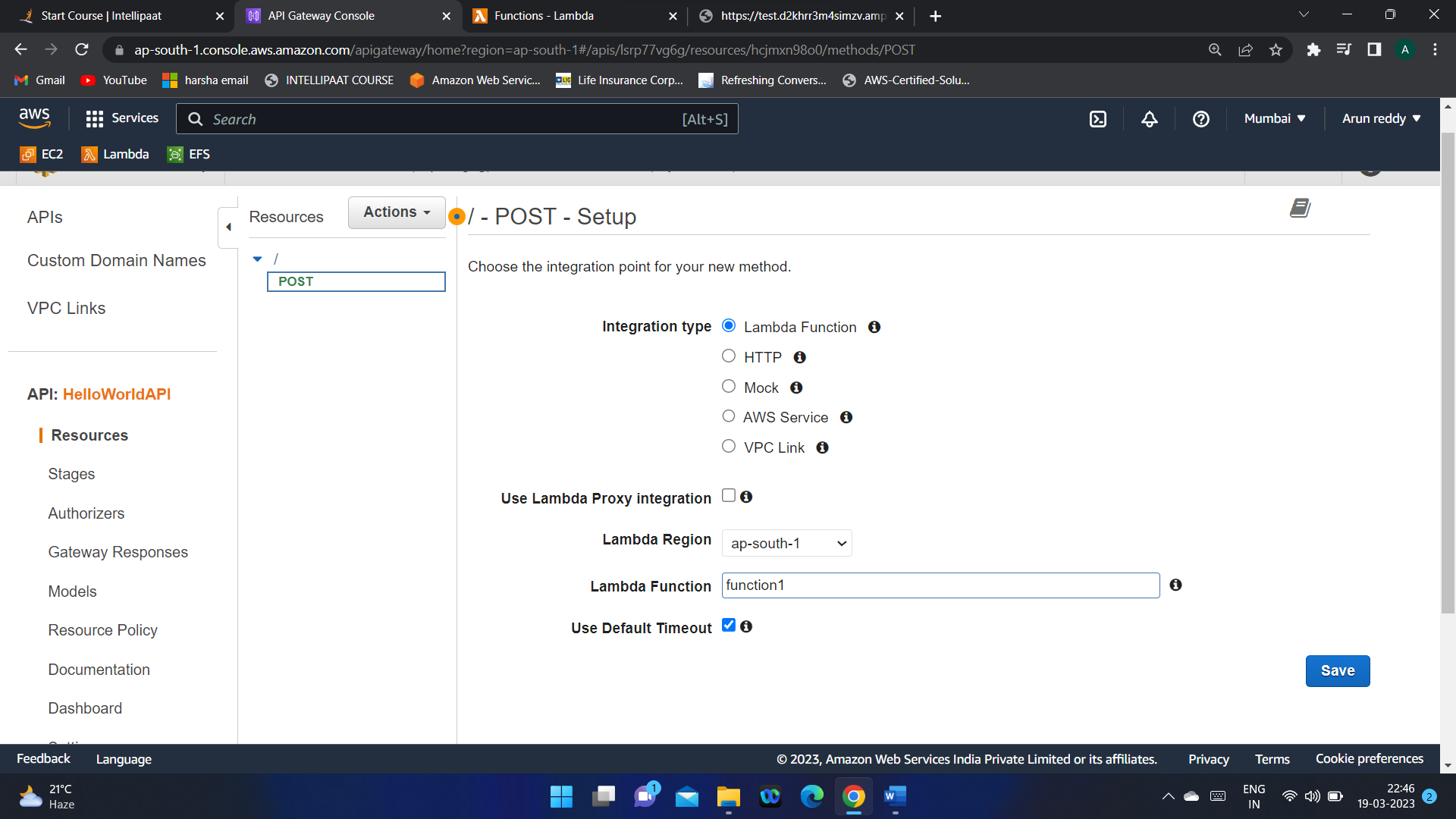
**1.Build a Rest API**

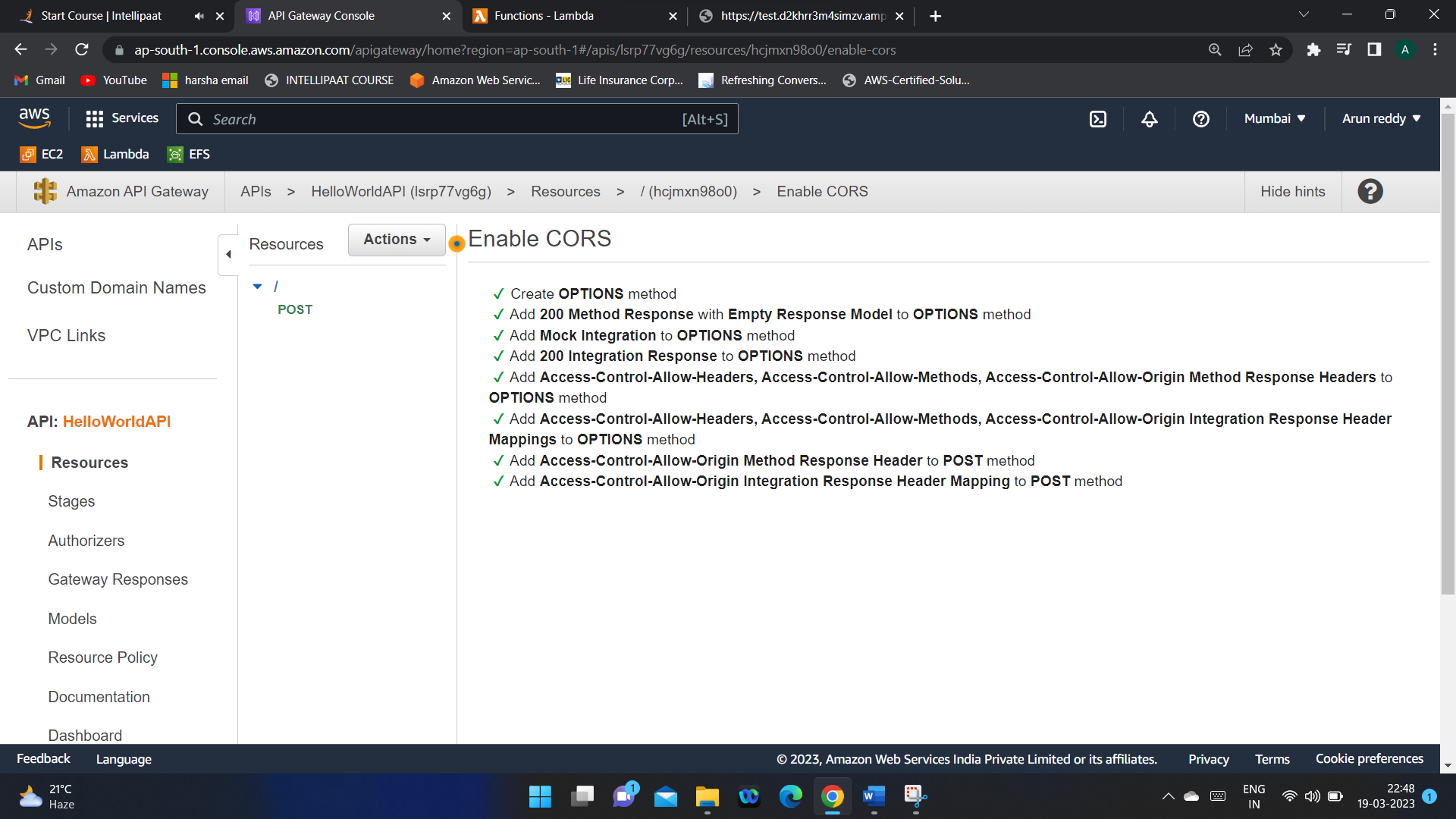
****

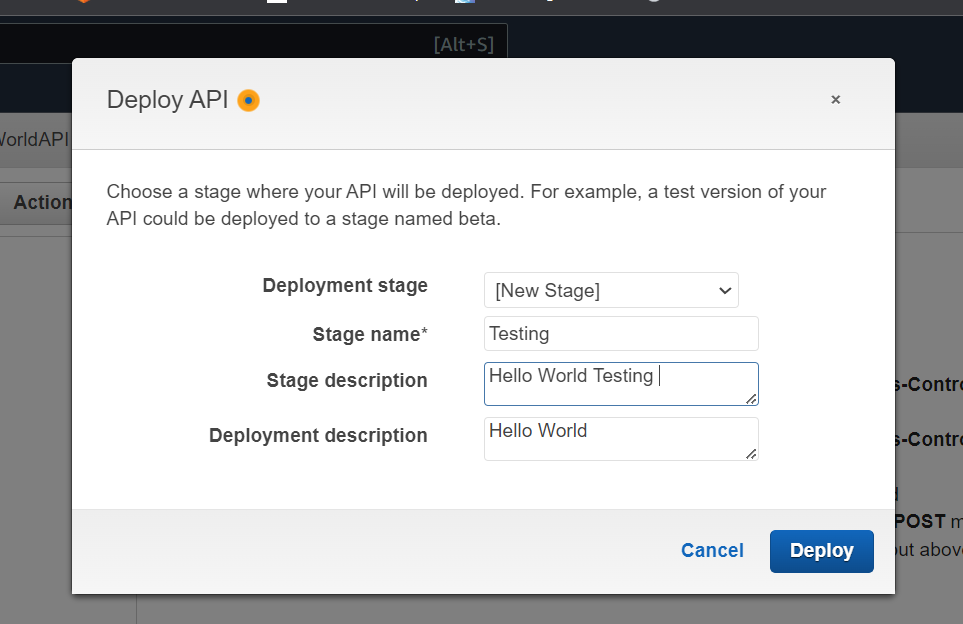
**2.In the actions create a method (POST)**

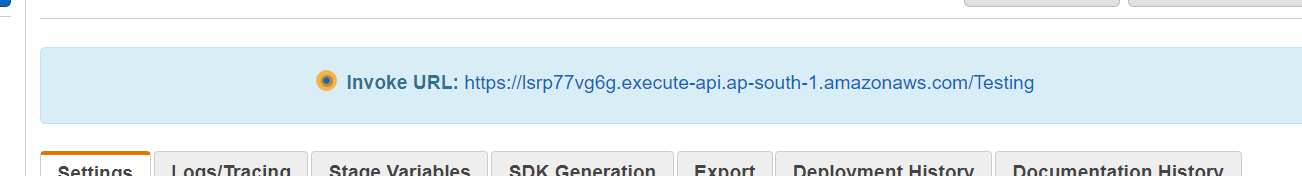
**Post ->as we want to post something into Lambda function**

**3.Enable CORS in Actions**

****

****

* So Now we have created a Resource and a Method
* IF we enable CORS, then our Amplifier would interact with API gateway and API Gateway would trigger Lambda
* Now we can Deploy our API (in Actions dropdown)

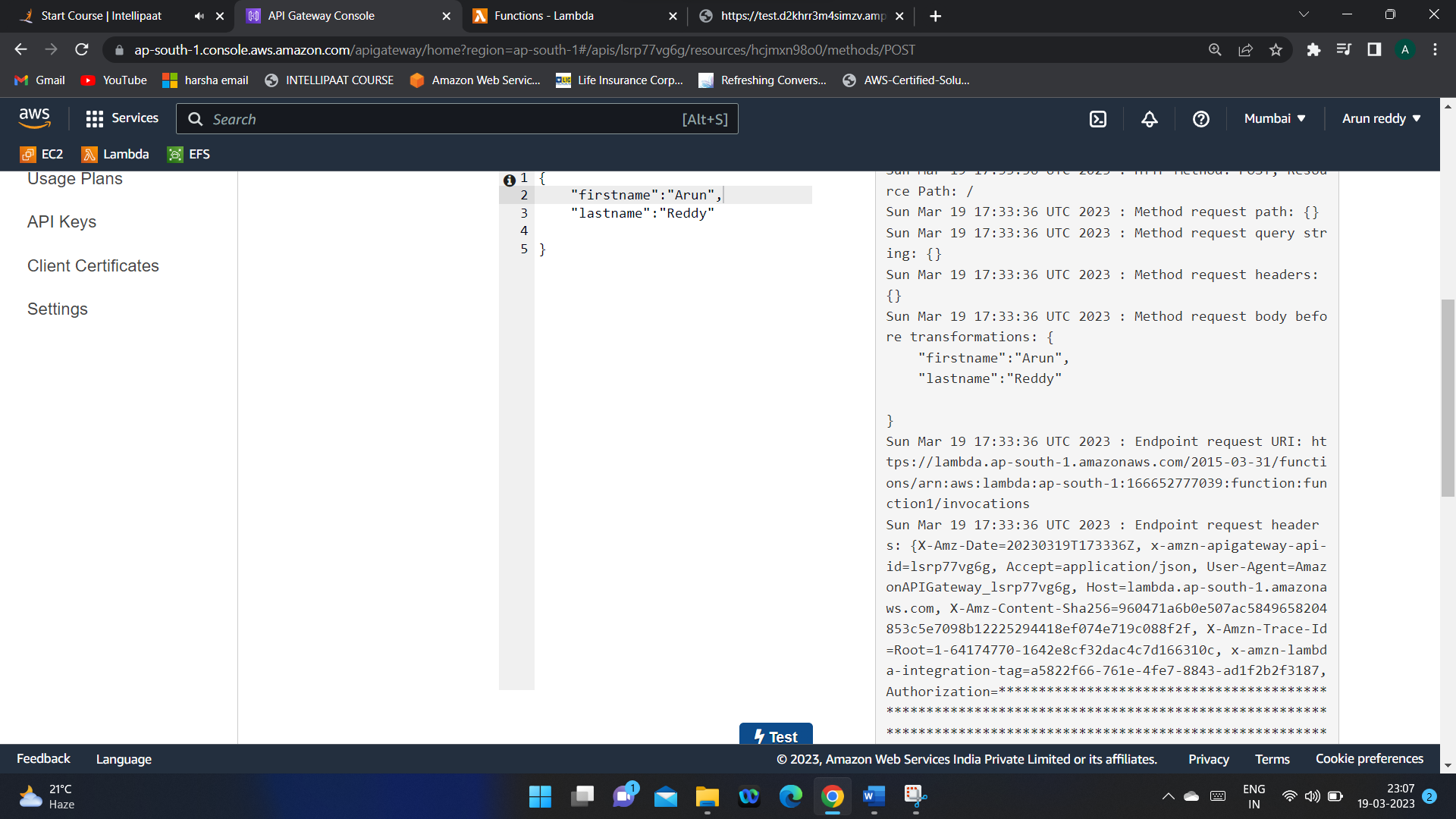


After Deploying we get an INVOKE URL

<https://lsrp77vg6g.execute-api.ap-south-1.amazonaws.com/Testing>

4.Go to resources and select the method we created and do “**TEST**”

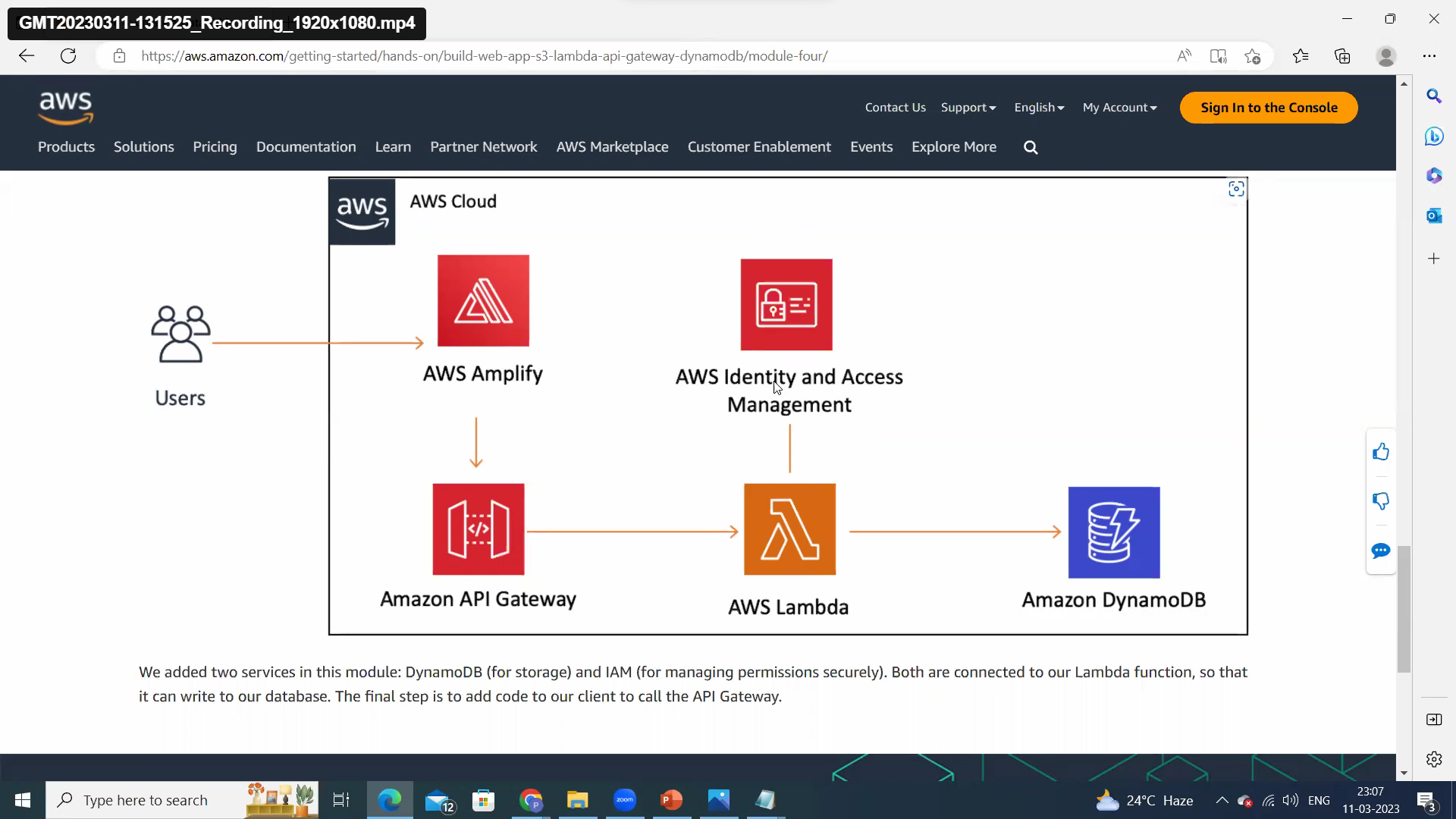
The return code should be 200 for successful testing



Now We have created API gateway and connected it to Lambda function

**IV) 1.Create Dynamo DB Table**

**2. save the ARN of the table :** arn:aws:dynamodb:ap-south-1:166652777039:table/helloworld



**3.create an IAM policy that would give lambda access to write data in Dynamo DB**

* + **Instead of creating a new role and giving policy to it and then adding it to Lambda function . We can directly modify the role permission of Lambda function to access the dynamo DB (in lambda function-> Permission ->Execution role)**
  + **Create inline policy for the same function**

**4.After creating IAM policy, you can able to put the data from LAMBDA to Dynamo DB**

**5.once you test the below code , it puts the name in Dynamo DB**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**6.DYNAMO DB results:**

**It returned the name that was tested in Lambda.**

**A screenshot of a computer

Description automatically generated**

**V) The final step is to add code to our client to call the API Gateway.**