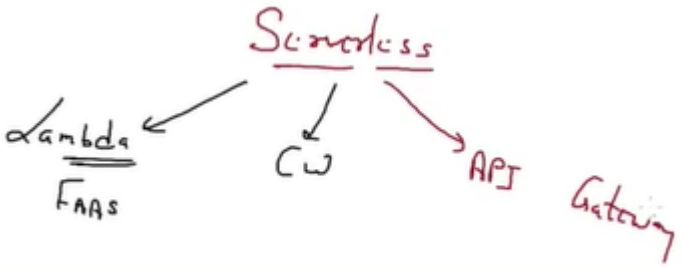
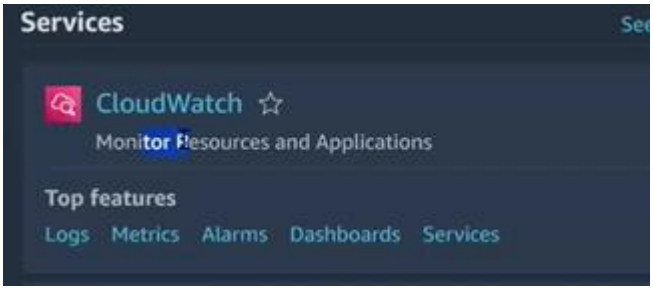
AWS Training Session No.04

Summary 06-03-2024

- In the last class we discussed **lambda** and **cloud watch** service
- As we know lambda is a serverless service means the server is there but all the server is managed by AWS cloud
- The cloud watch service is used for monitoring
- API gateway also is a serverless service

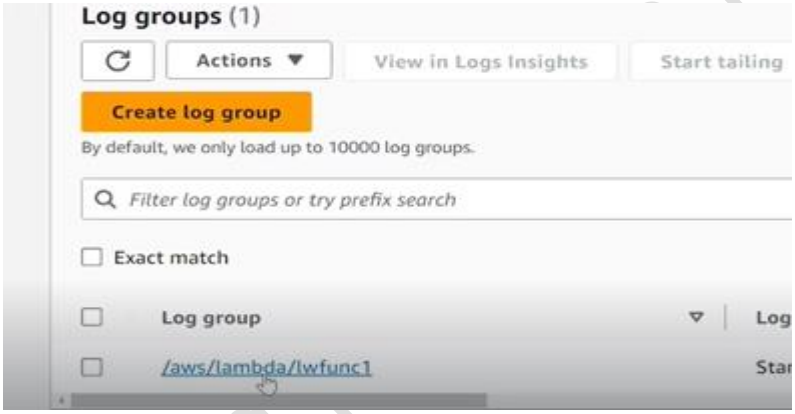


- As we know in the lambda service whenever you create a function then we write the code after we need to test the function
- When you test the function then they return the function output
- If the developer changes the code then we need to again test the code means whenever your code gets changed then always we need to test
- After running the code they create a log behind the scenes and this log is what you see in the Cloudwatch service as we know Cloudwatch is used for monitoring
- Whenever your code fails then go and see the Cloudwatch after that we see the error

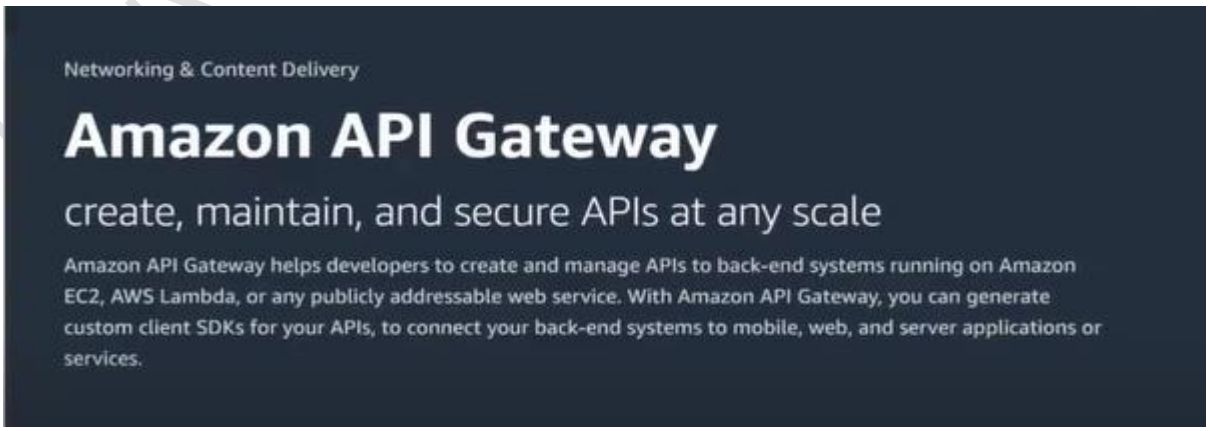


- Cloudwatch has Sub services

- In the Subservices we have a **logs** service this service gives when your function starts or ends they give how many print functions you have in the function and the time duration of the code run
- This service stores all the Information the log groups
- If you want to log then go to the log groups and create one log group when you run the function



- This information is stored permanently
- **Use of API Gateway**: When you have multiple functions or apps or microservices, you want the customer to reach any of them using just one URL. You can set up an API Gateway that will route the user to the desired location based on the /path they entered.



- API gateway is helping the user reach the final destination. It provides an interface, a kind of door to the final application. Hence, it is called **API**,

Application Programming Interface. API gateway server can be set up and launched on the cloud by using AWS API Gateway.

- **The google.com example**:

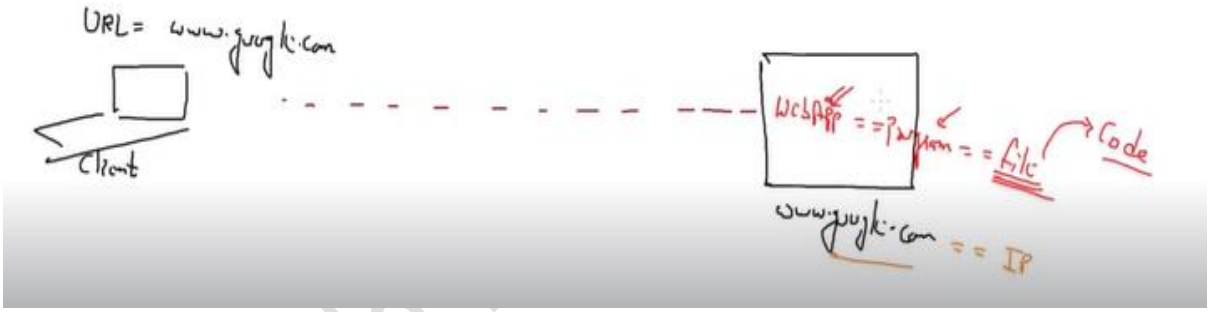
  ➢ Whenever you type google.com, you hit the API Gateway running in Google's server.

  ➢ www.google.com is the IP address/hostname/domain name of the computer onto which the API Gateway is running.

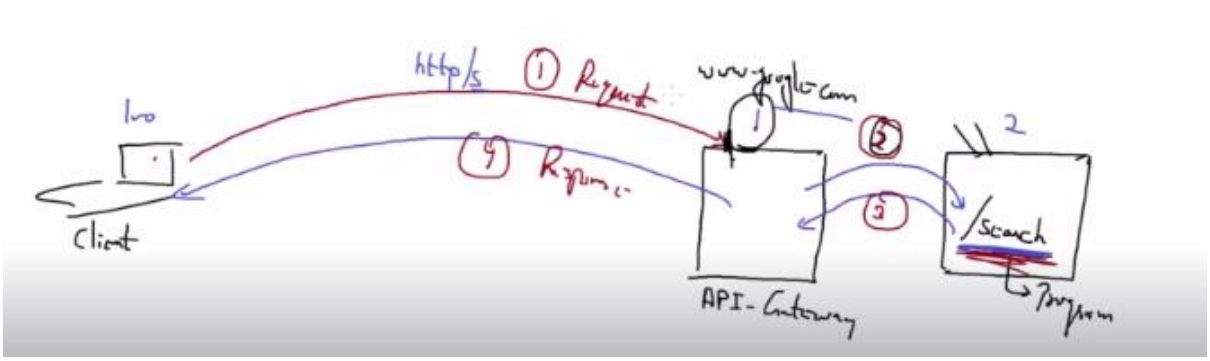
  ➢ According to the /path, you'll be sent to either /search or /mail in this example.

  https://www.google.com/search

  ➢ The default rule always takes the user to /search, if /path is not specified. www.gmail.com is converted to www.google.com/mail in background.

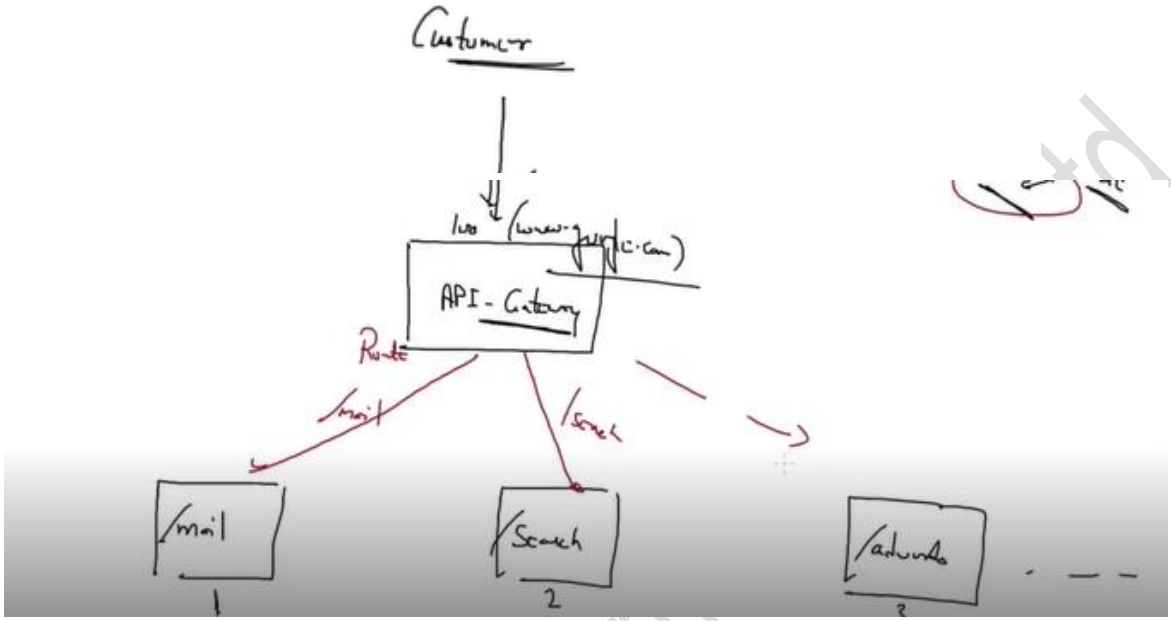


- Whenever you write the code we create a file and we write the code in any language
- For example whenever you hit Google, com then behind the site client hits API gateway after that we give the interface to you and this gateway layer is hidden with the help of protocol

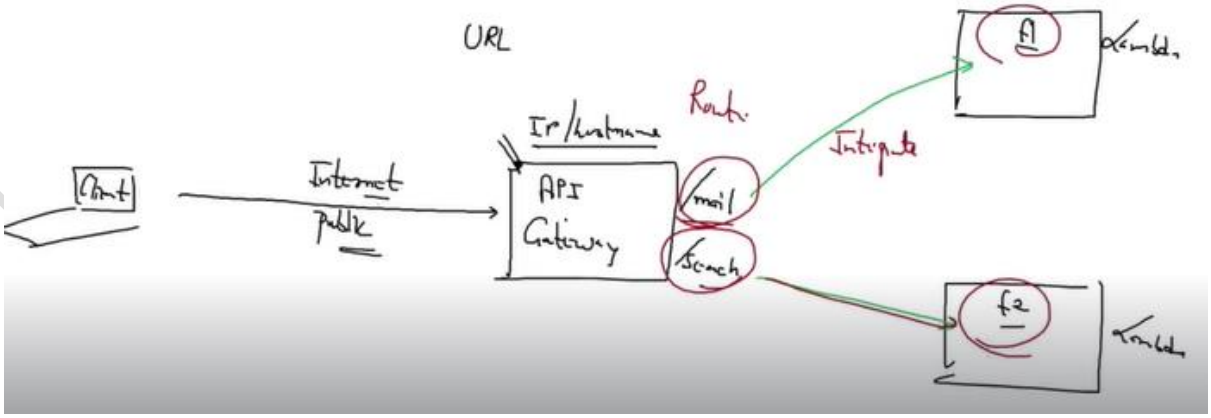


- Means the client gives the request to the gateway then the gateway gives the response to the client

- Lambda service we use as a **backend** server means the **endpoint** also known as **route**
- For example we have three servers: mail, search, and admin
- If the customer wants to connect with the mail server then the customer goes to the gateway after that gateway connects to the mail server



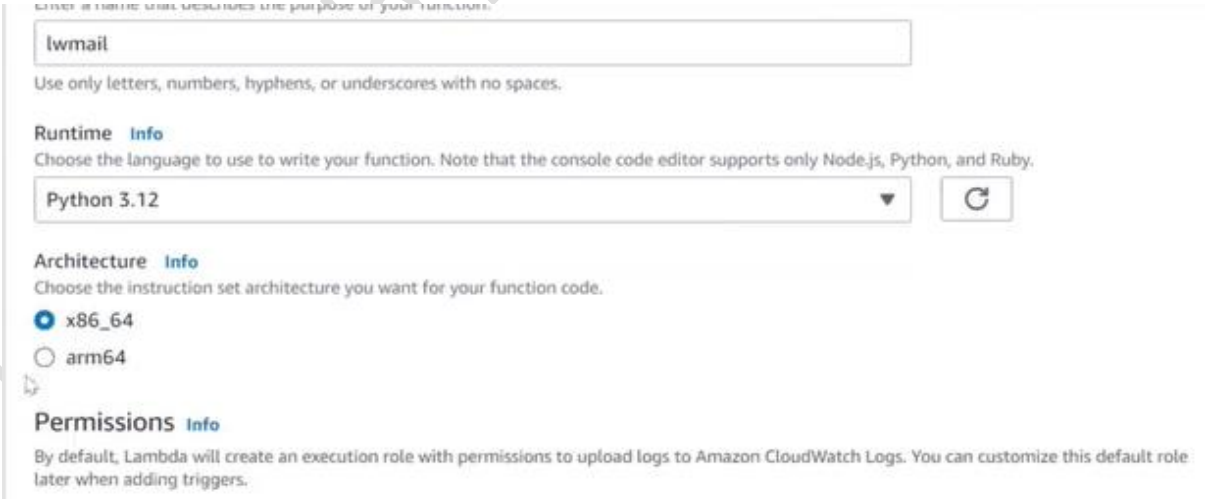
- One use of a gateway use is they create a path
- **API Gateway** gives a lot of things to you like a load balancer, security, subscription, Etc
- API Gateway manages the Traffic they give Autoscaling for that API Gateway uses a **reverse proxy**
  For example:- The URL syntax is **https://IP/route**
- To practical we create one demo



- To integrate lambda with API Gateway
- For that first go to the AWS cloud then search lambda service

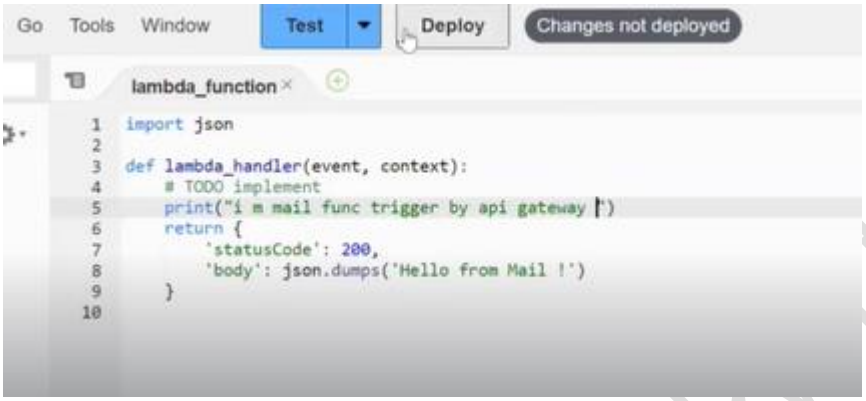
- Then create a two lambda function
- To create a lambda function first click on the create a function

- After that Give the Function name and set Runtine to Python 3.0



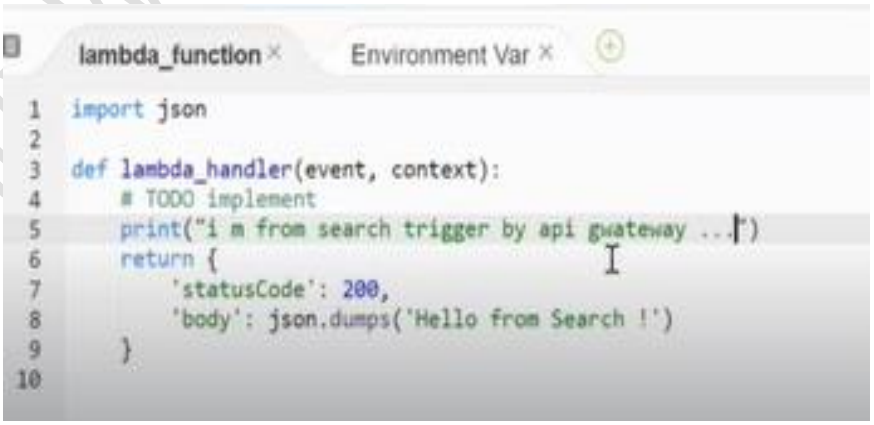
- After that click on the Create function

- After that write the code in the function and to run code we have two ways to run the code first is manually by going and clicking the test button and the second is the API Gateway triggers the lambda function automatically when the client is coming



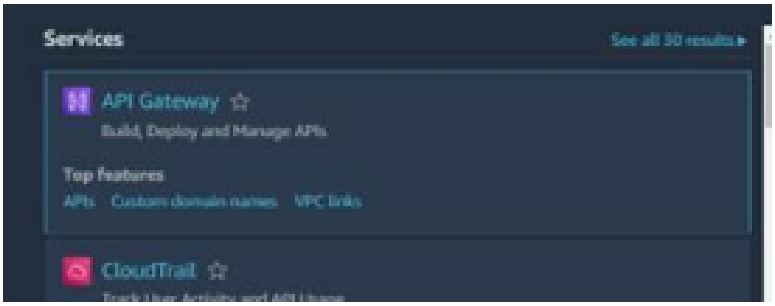
- After that create a second function

- Then write the code in the function

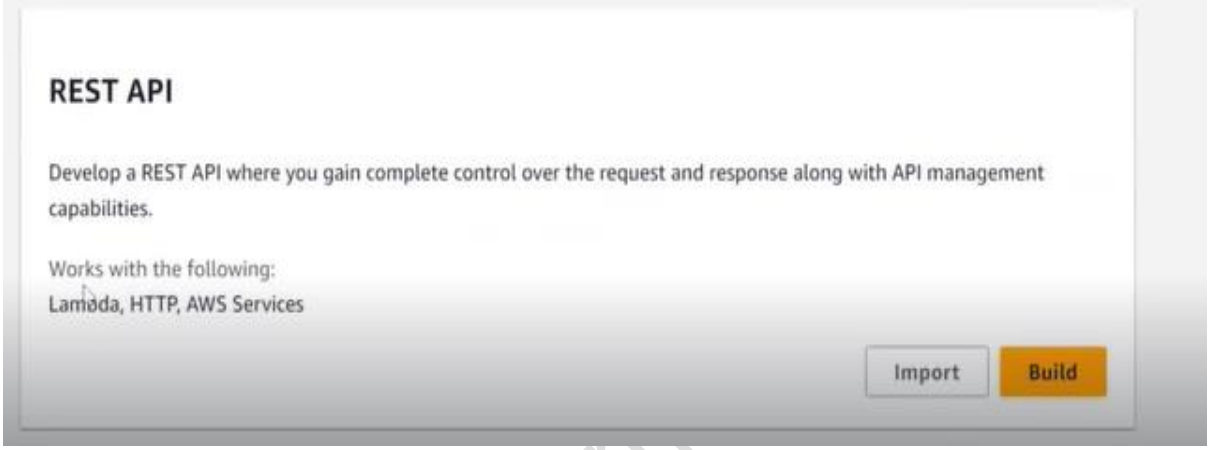


- After that deploy the code first time
- These functions will be running your private AWS account. To give outside people access to it, we will set up API Gateway.

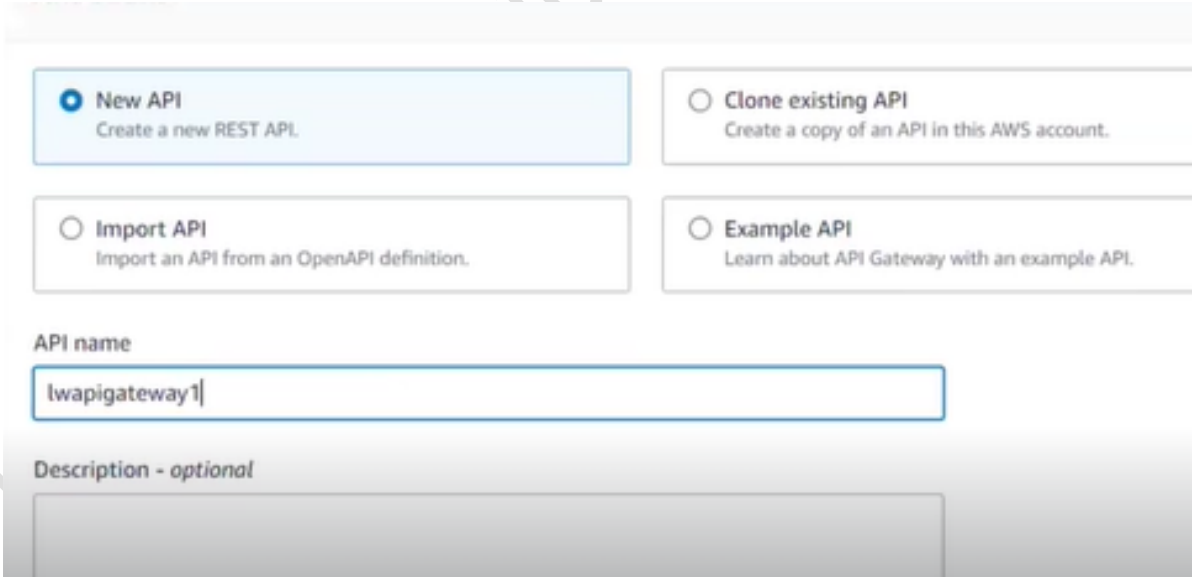
- Search API Gateway



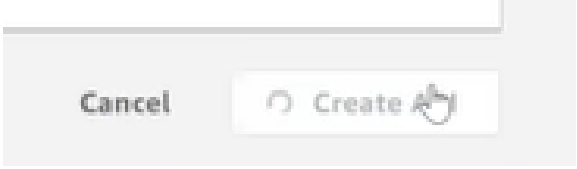
- To use the API gate Choose an API type and use the REST API



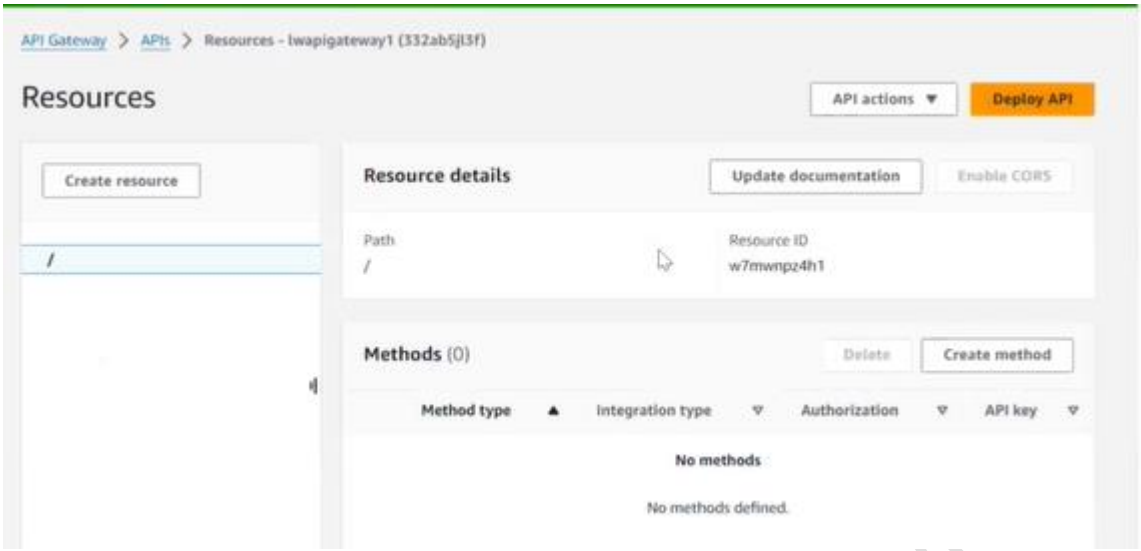
- After that click on the Build
- Then give the API Gateway name



- After that click on the Create API

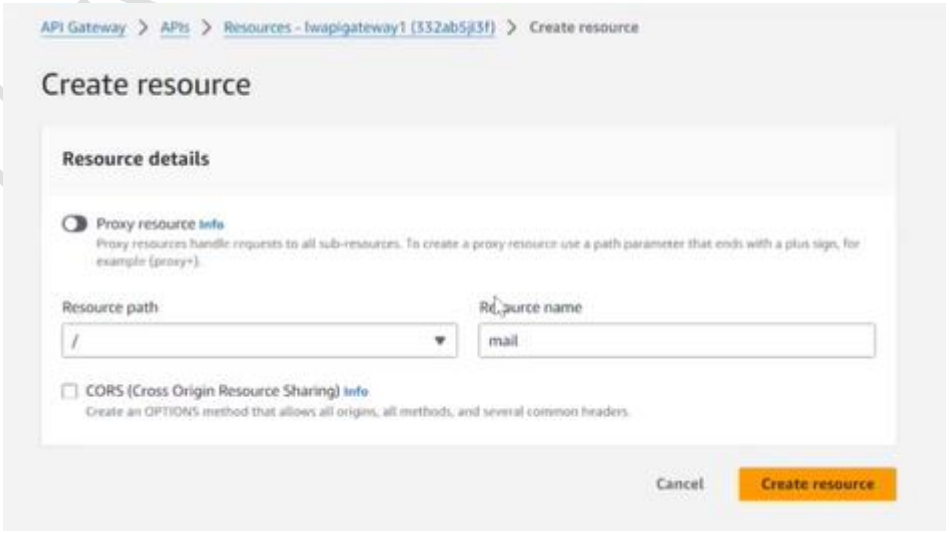


- Now we successfully create API

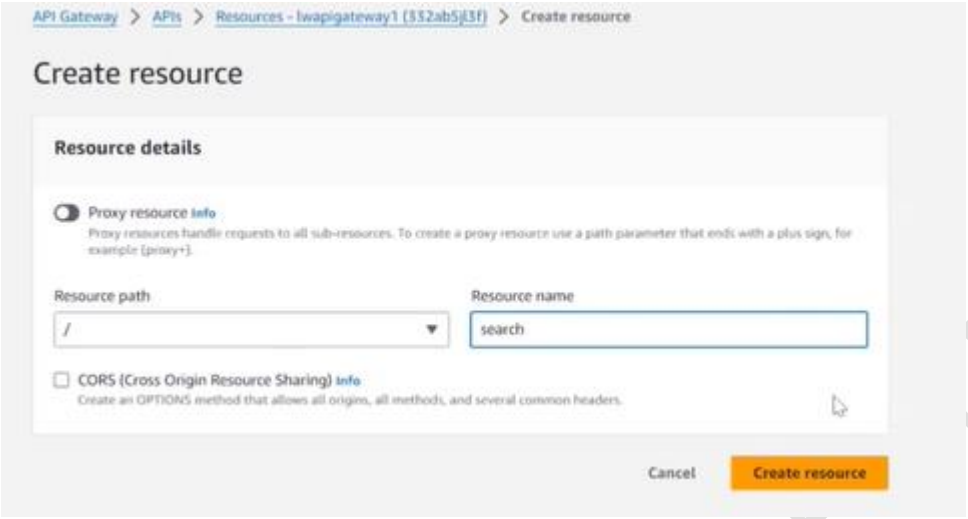


- API will manage for you the hardware, networking, route, and public IP address. It can also manage the load by scaling the servers in times of incoming or sudden traffic. It is highly scalable with good performance.
- EC2 service is an on-demand service and charges you for the number of computers you launch even when there is no traffic on your web app.
- API Gateway charges you per request. Hence, saves cost as well
- After that we need to create a resource click on the created resource

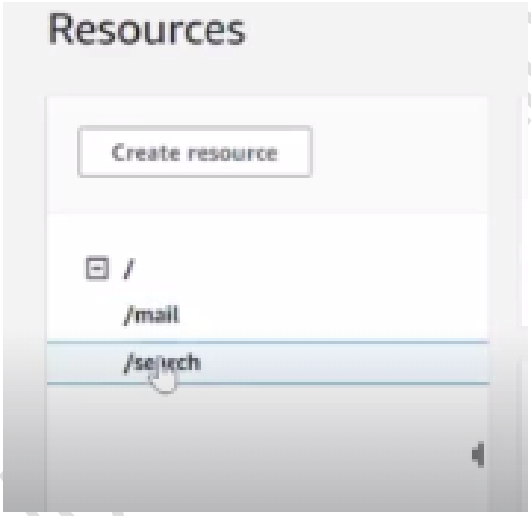
- After that give the resource name then click on the Create resource



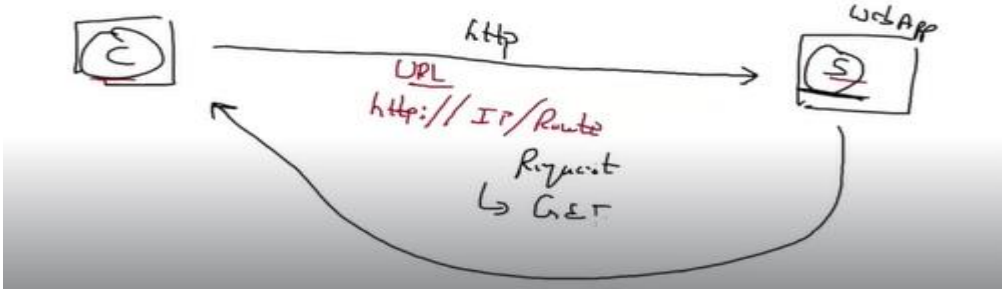
- We create mail recourse successfully
- Now we want to create one more resource for the search server



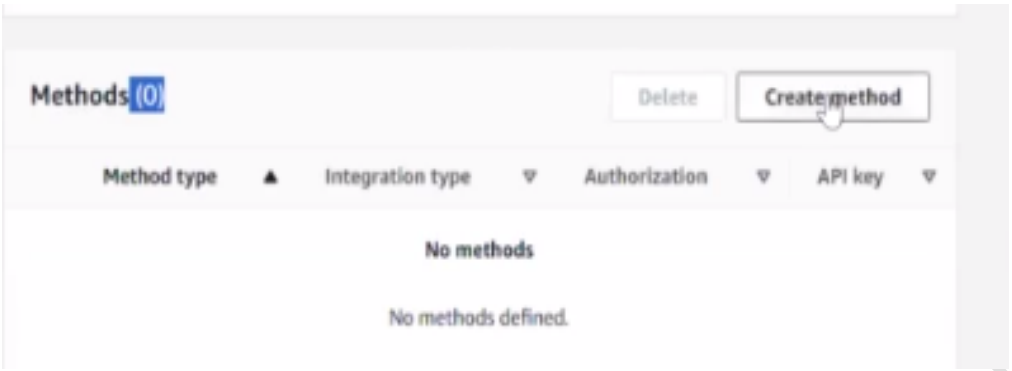
- As we can see we create two resources



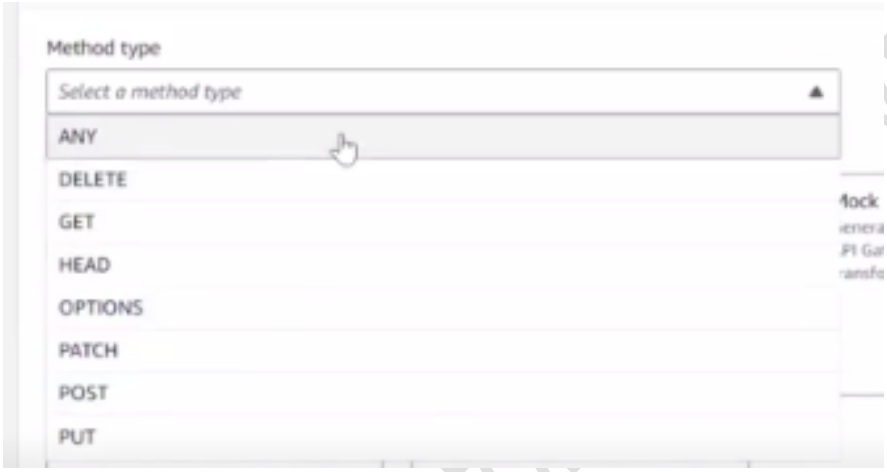
- When visiting a website, a user sends the following:
  Protocol: http/https
  Url Route: /path
  Verb: GET/ POST / …
- The Gateway will respond to a request only when the route and the verb match that of an existing Lambda function.



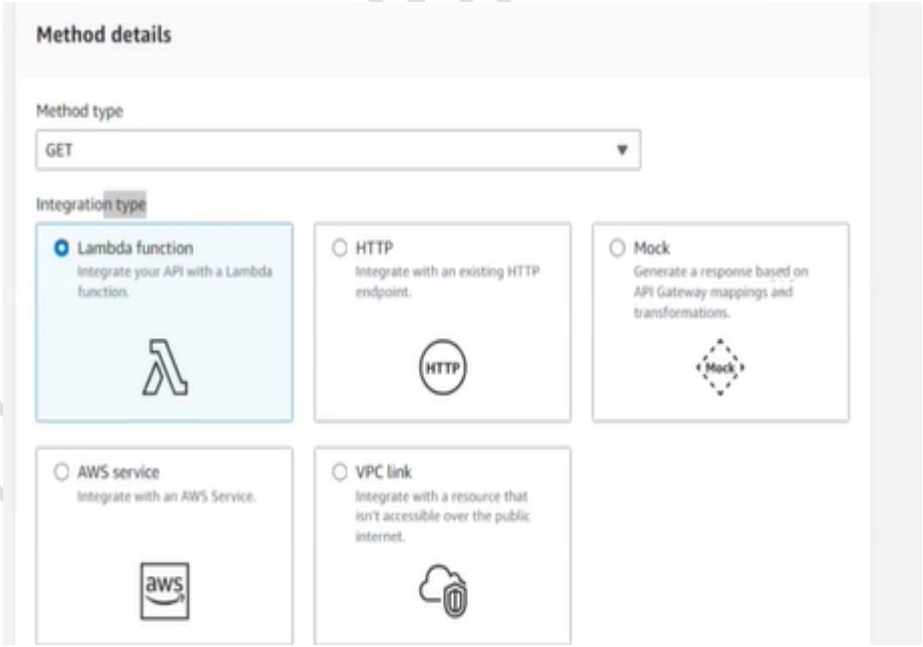
- For that we need to create a method for clicking on the create method



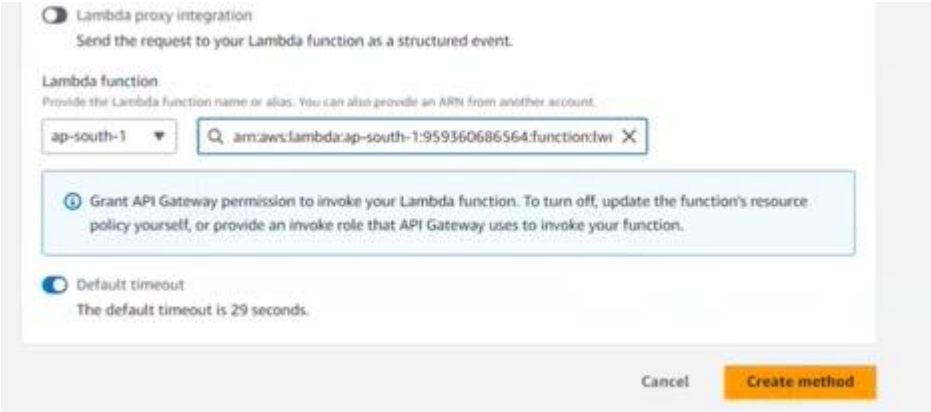
- After that choose the method we have a lot of methods



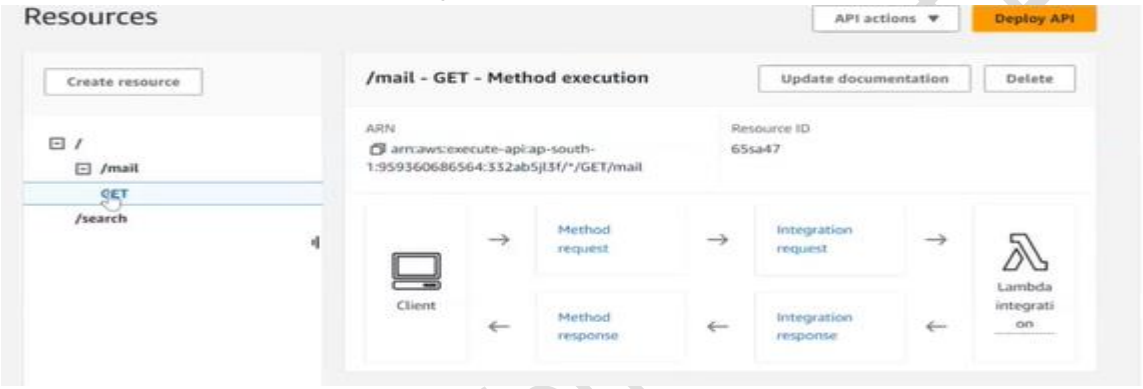
- We use the Get method



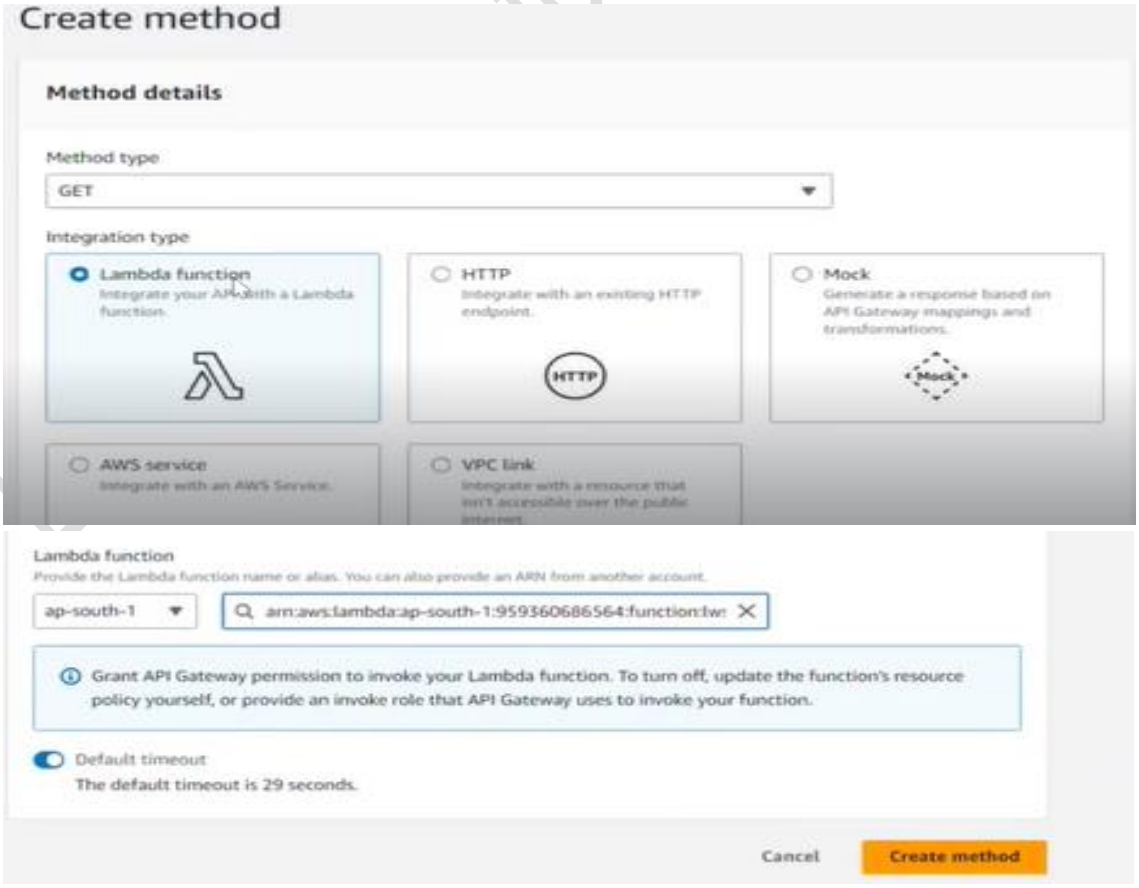
- After that selecting the region is very important because if your gateway service and lambda service are present in a different region then it not working
- After that select the lambda function



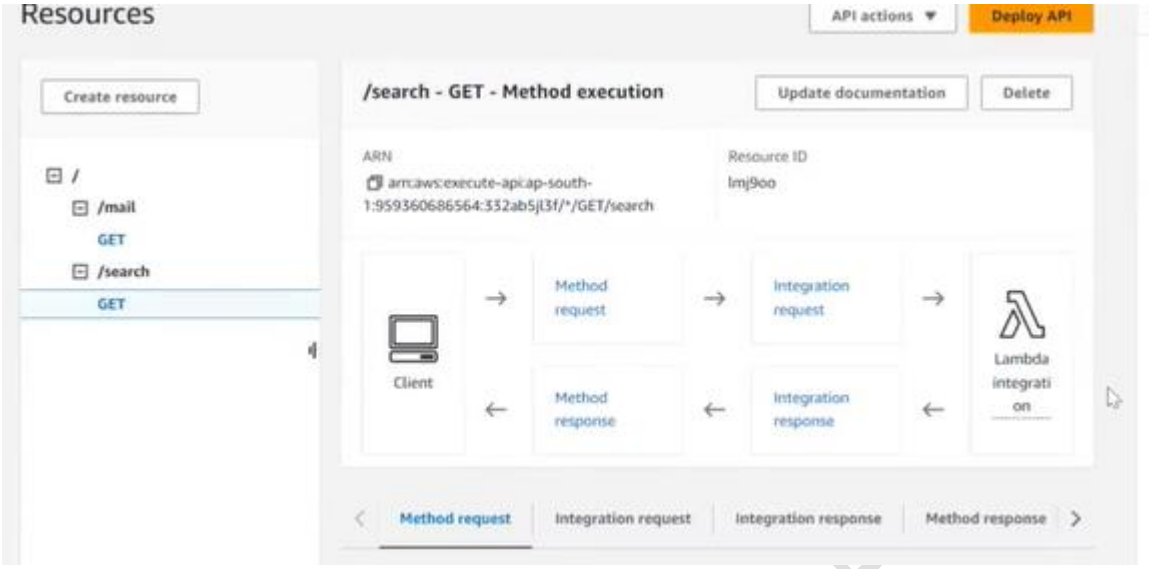
- After that click on the Create method
- As we can see we successfully create a method



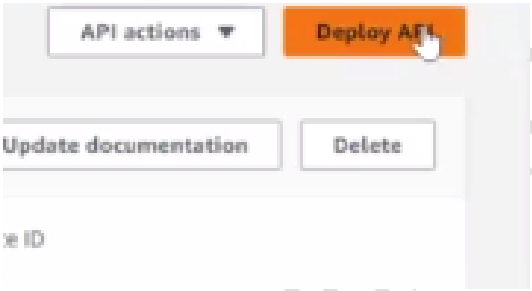
- Then same method we create for the search server



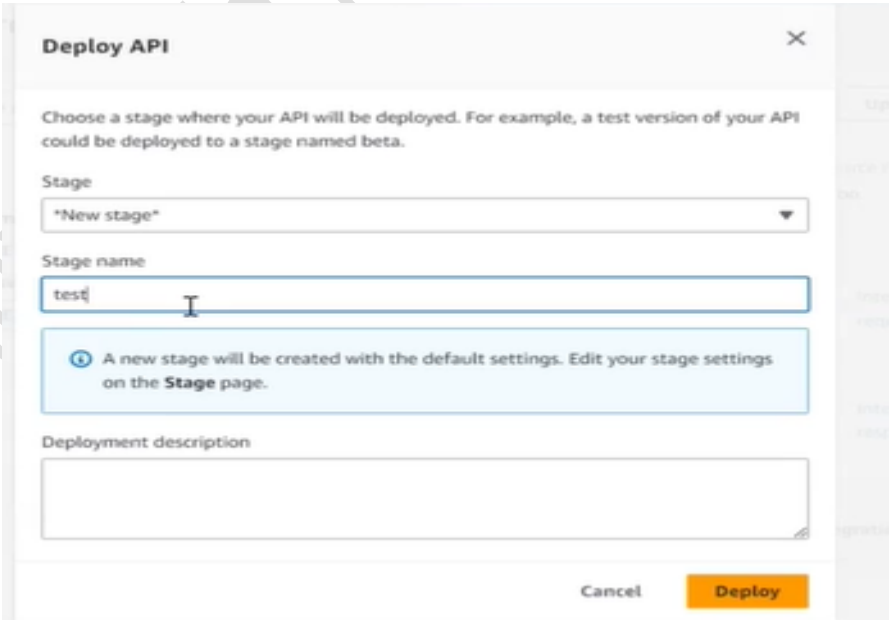
- As we can see we create two method



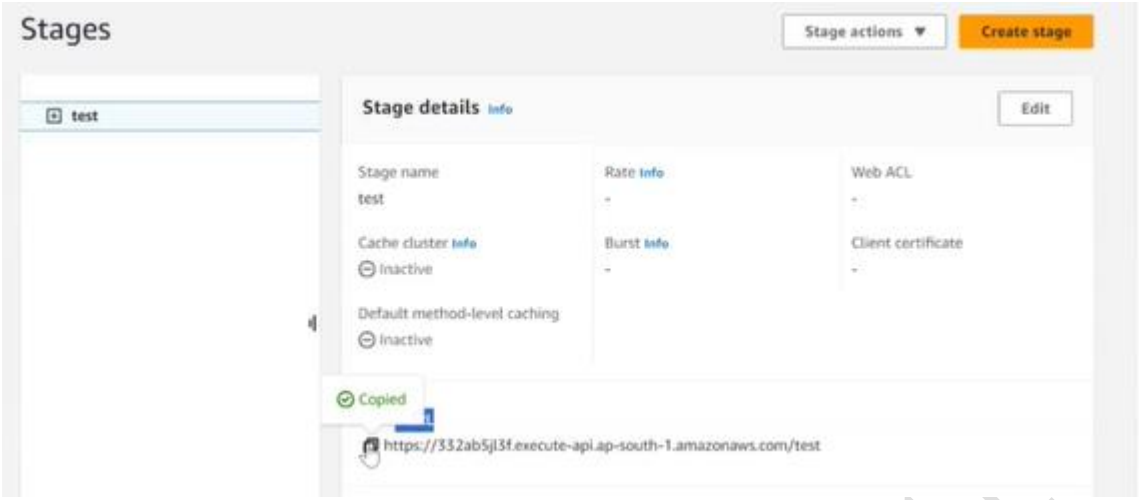
- After that deploy the API
- For that click on the deploy API



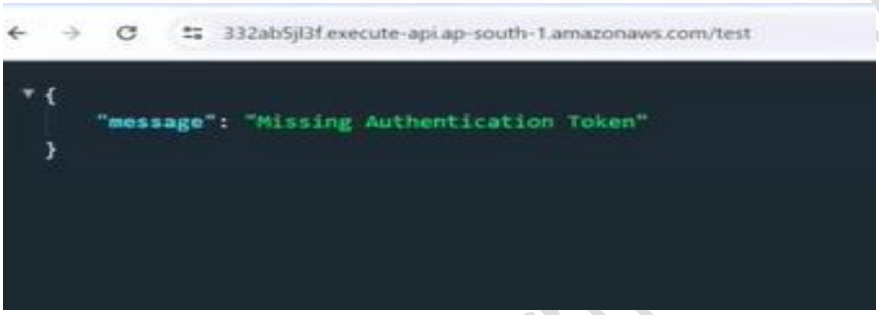
- After that give the stage name



- After that click on the deploy
- As we can see we create a stage in that stage we have one link copy this link and paste it into the browser

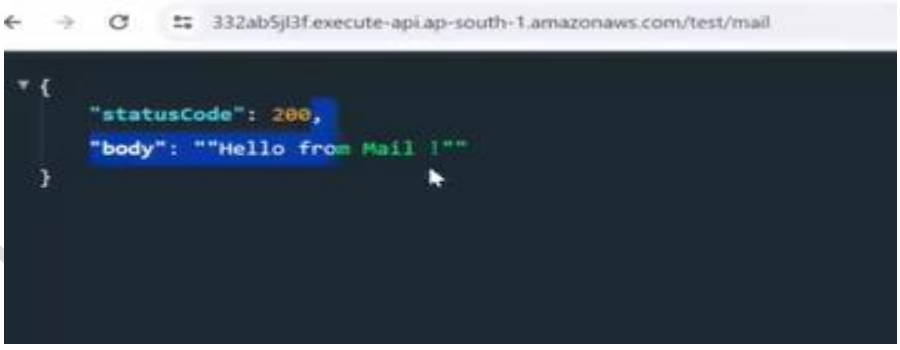


- Now we can see the test



- After that change at the end of the URL

- This URl is a public URL any connected with this URL
- After change you hit enter then give the output of the mail lambda function



- Same for the search server only change the ULR

- If anyone changes the code then only refresh the page they give output to you
- In the search function