Mayuri Yevande
D15B
70

EXPERIMENT – 11

AIM: To understand AWS Lambda, its workflow, various functions and create your first lambda function using python / Java / Node.js

THEORY:

⇒ AWS Lambda :- It is a serverless computing service provided by Amazon web services. Users of AWS Lambda create function, self contained applications written in one of supported languages and runtimes and upload them to AWS Lambda which executes those functions in an efficient and flexible manner.
AWS Lambda is a fully managed service that takes care of all infrastructure.

⇒ Features of AWS Lambda:-

- AWS Lambda easily scales the infrastructure without any additional configuration.
- It offers multiple options like AWS S3, Cloud watch, Kinesis, Code Commit and many more to trigger an event
- You dont need to invest upfront. You pay only for memory used by lambda function and minimal cost on number of requests hence cost efficient.

- AWS Lambda & Serverless Framework uses AWS IAM to define all the roles and security policies.

⇒ Packaging functions:- Lambda functions need to be packaged and sent to AWS. To help us with this process we use SST. (Serverless Stack Framework)

⇒ Execution model:- The container that runs our function is managed completely by AWS. It is brought up when an event takes place and is turned off if it is not being used. This means if we are experiencing a usage spike, the cloud provider simply creates multiple instances of containers with our function to serve those requests:-

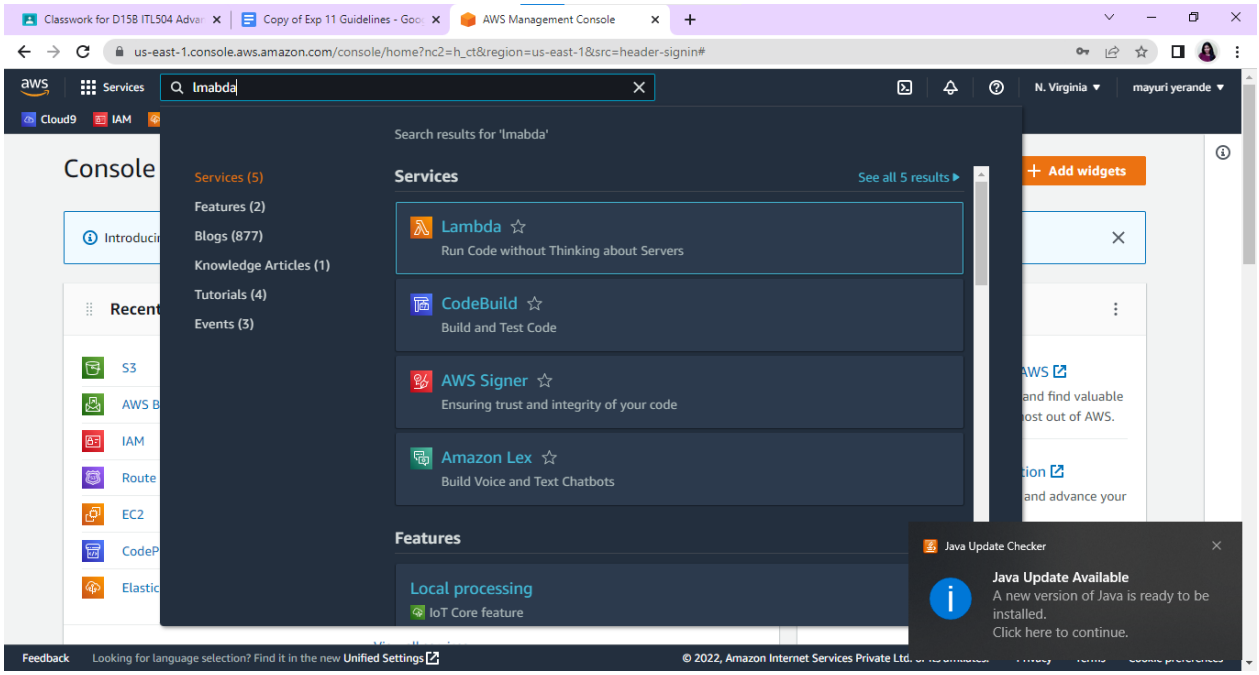
⇒ Due to Lambda's architecture, it can deliver great benefits over traditional cloud computing setups for applications where:-

- Individual tasks run for a short time
- Each task is generally self-contained
- There is a large difference between lowest and highest levels of workload of application.

⇒ Lambda functions are optimized for event based data processing. It is easy to integrate AWS Lambda with data sources.

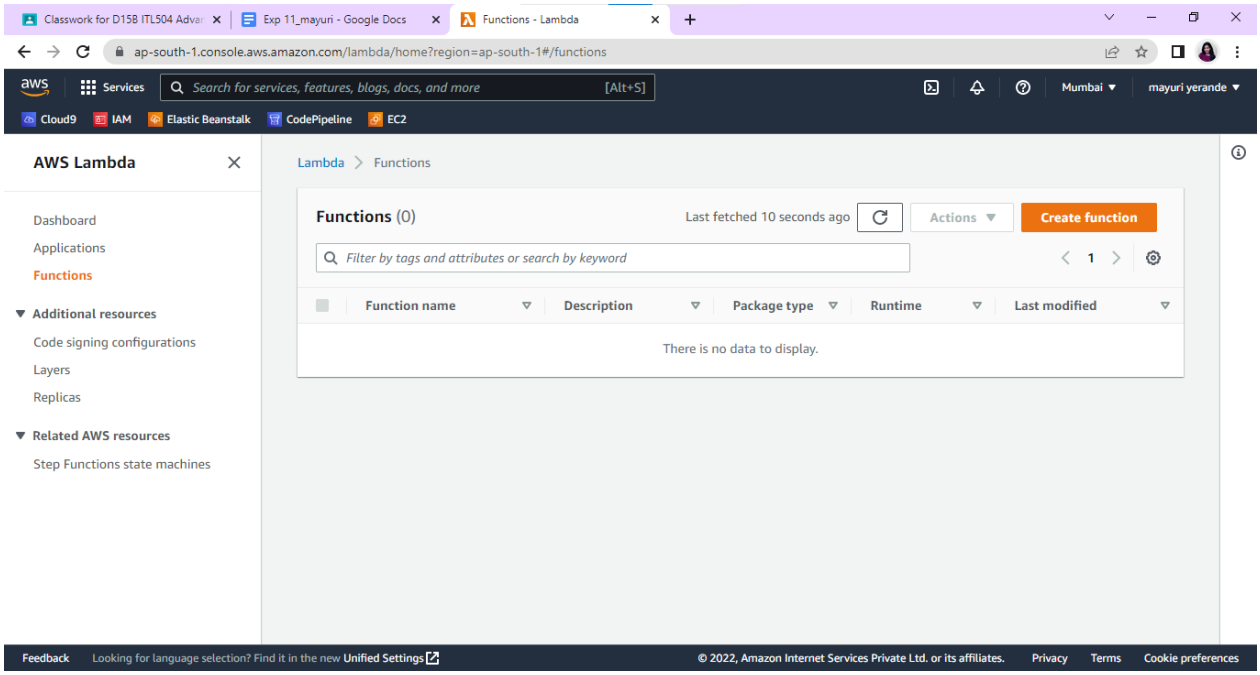
Steps to create an AWS Lambda function

Step 1: Open up the Lambda Console



- click on the Create button.

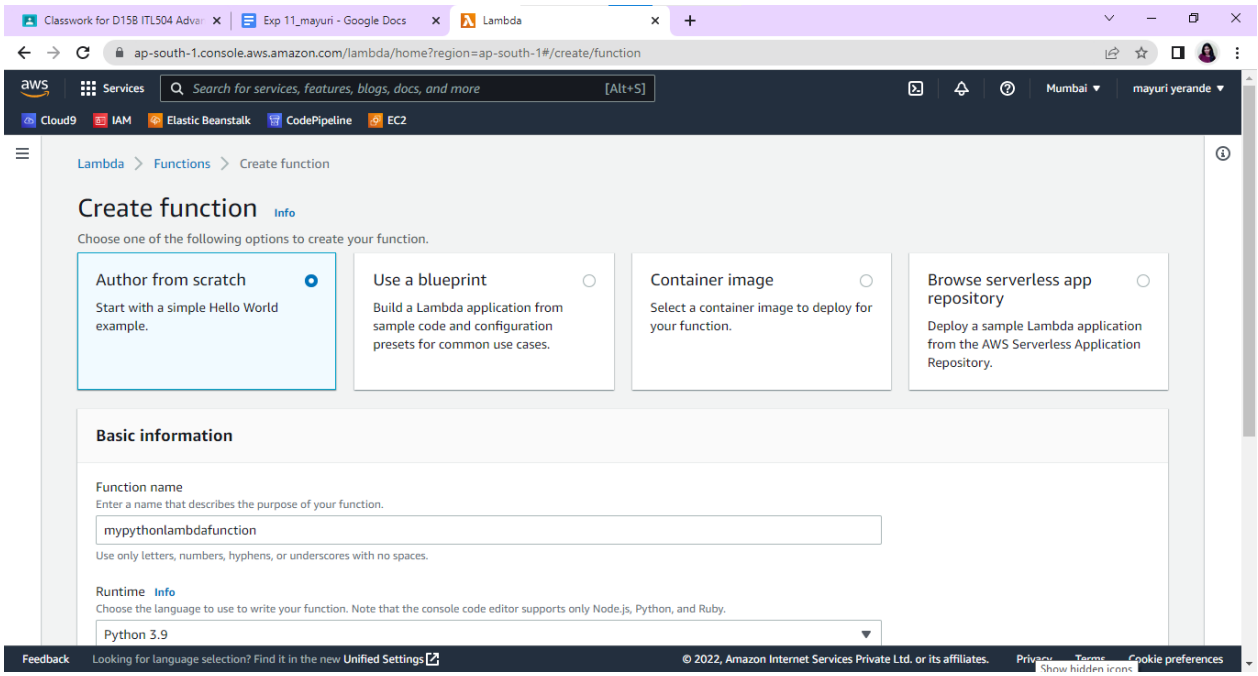
  Be mindful of where you create your functions since Lambda is region-dependent.

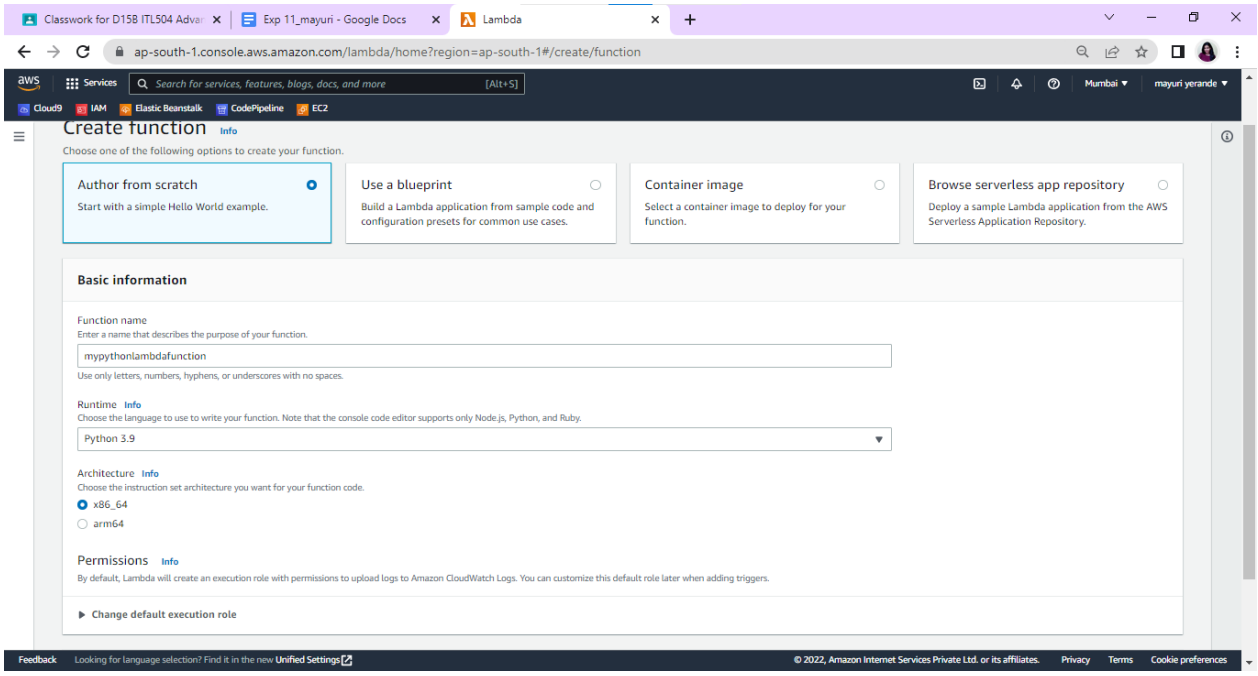


Step 2. Choose to create a function from scratch or use a blueprint, i.e templates defined by AWS for you with all configuration presets required for the most common use cases.

Then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones.

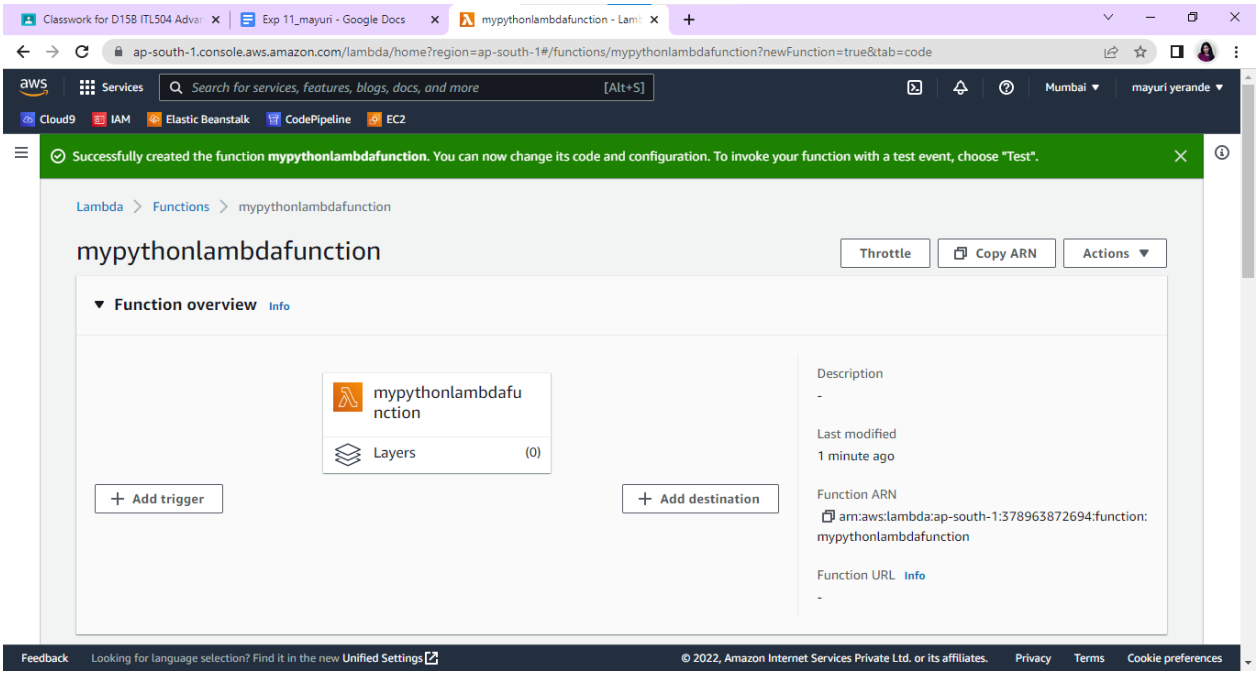
After that, choose to create a new role with basic Lambda permissions if you don't have an existing one.

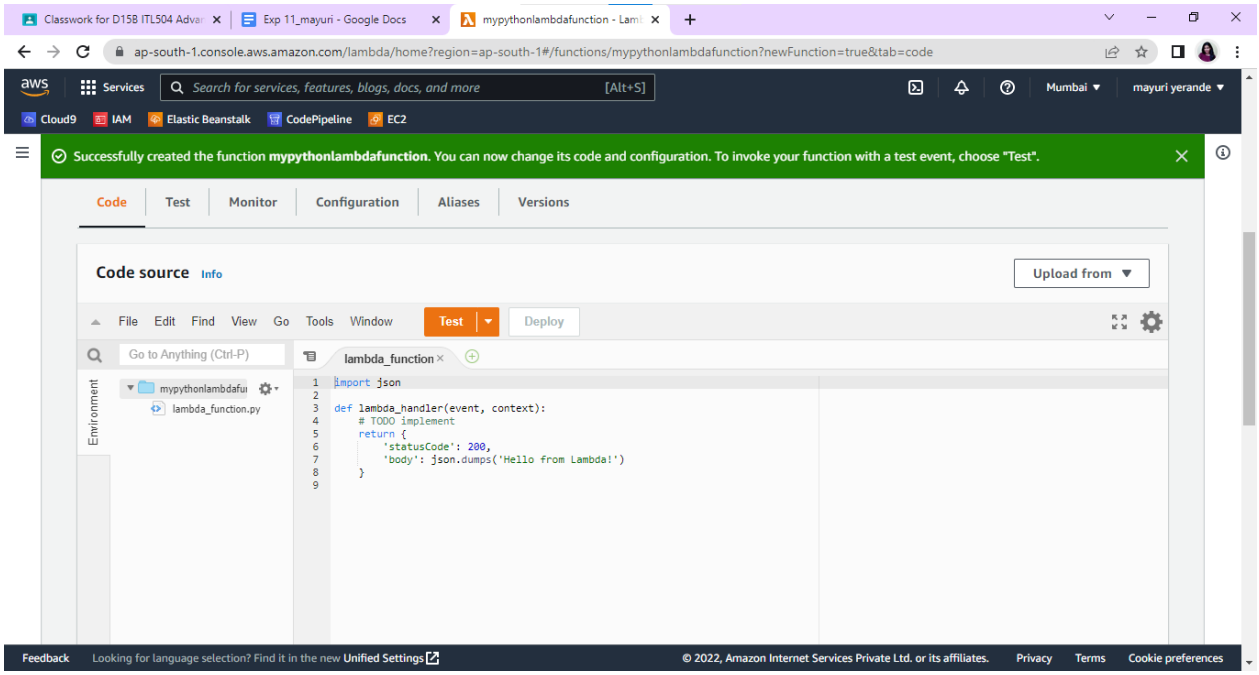




Click on the *Create* button.

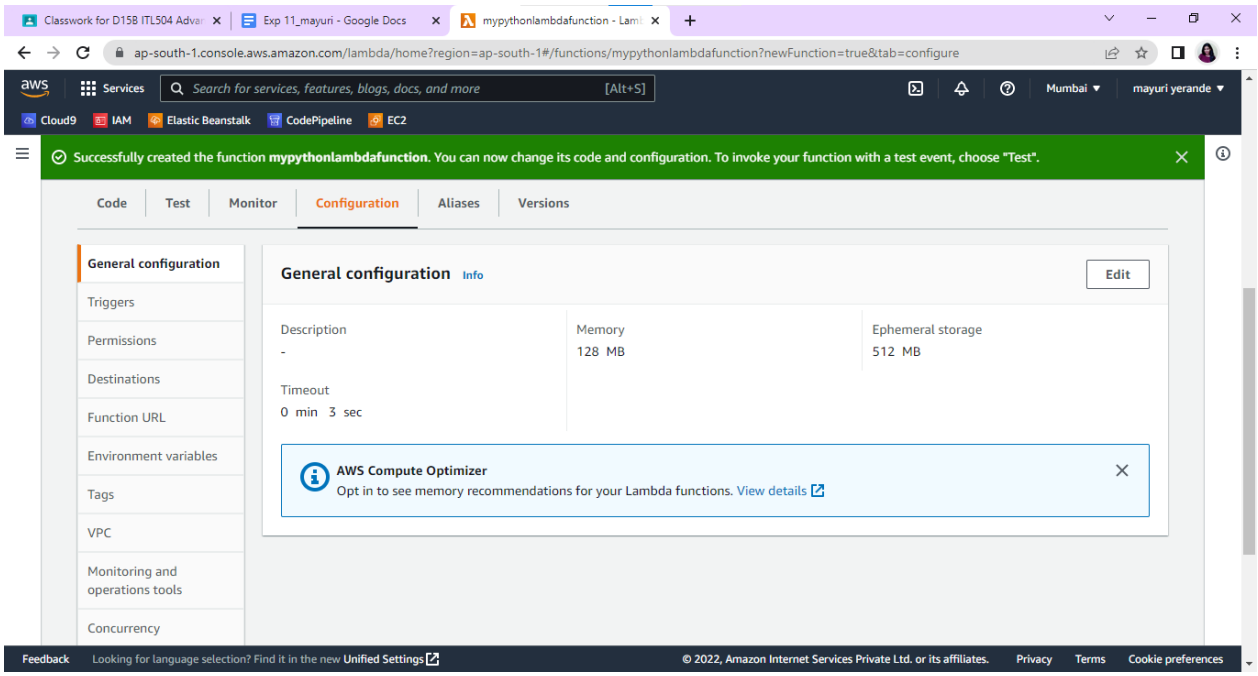
Step 3. This process will take a while to finish and after that, you'll get a message that your function was successfully created.

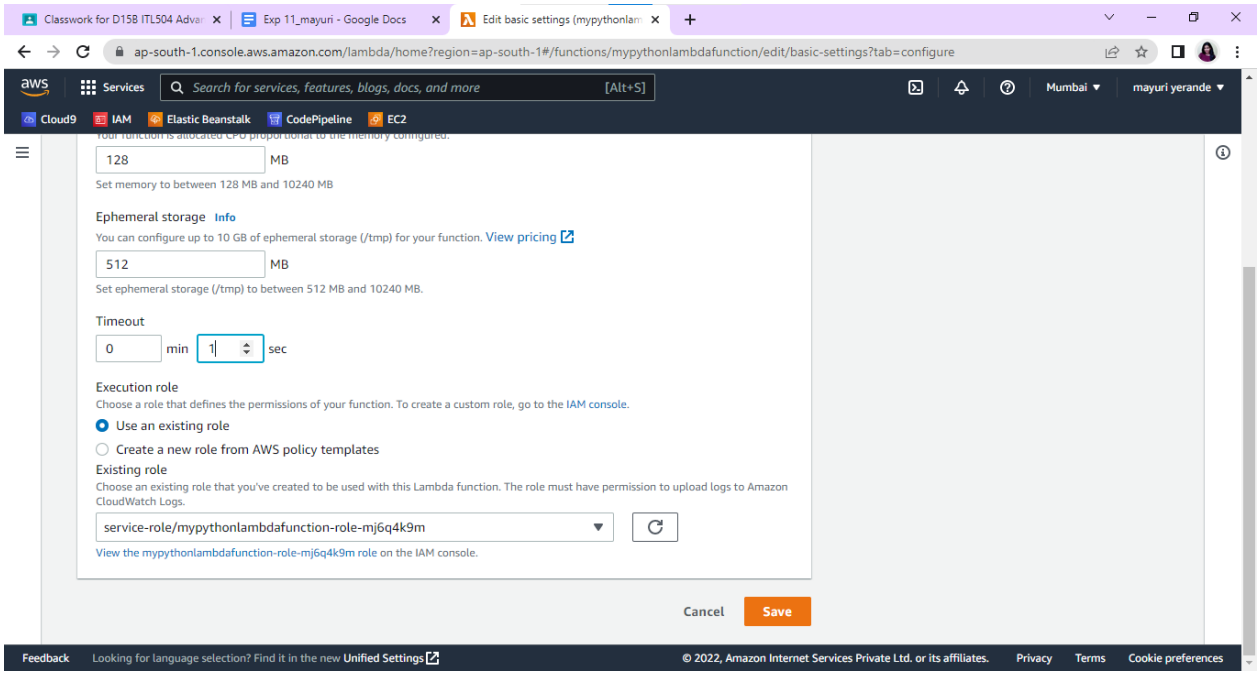




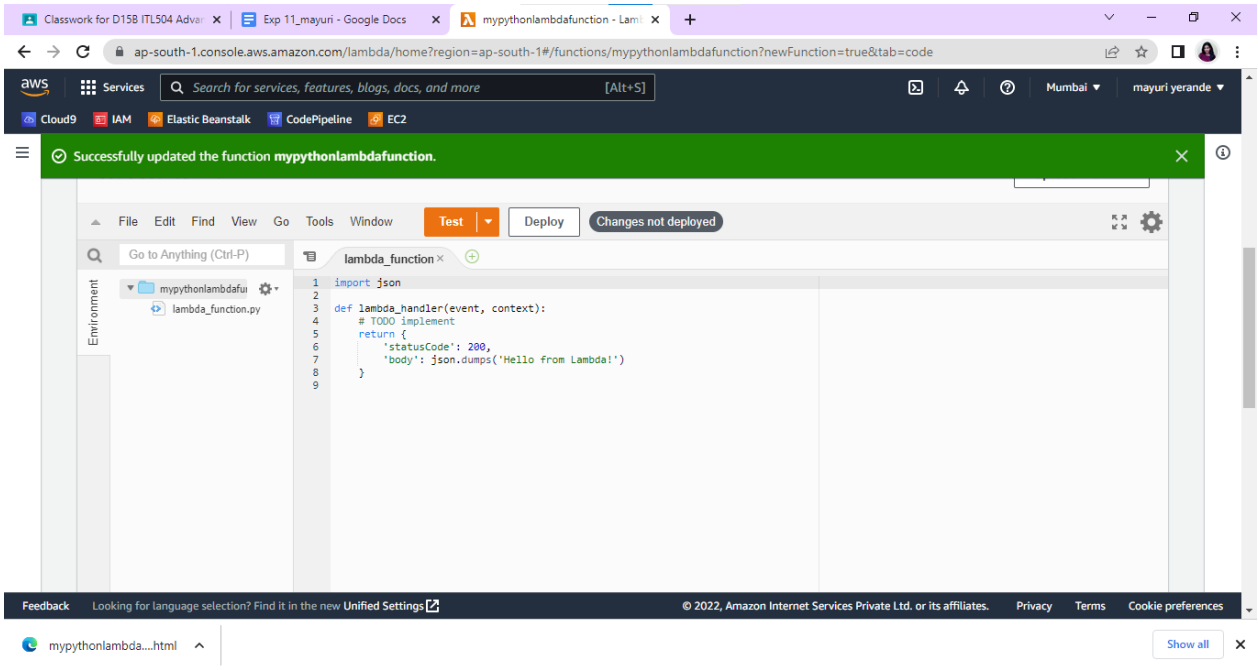
Step 4. To change the configuration, open up the Configuration tab and under General Configuration, choose Edit.

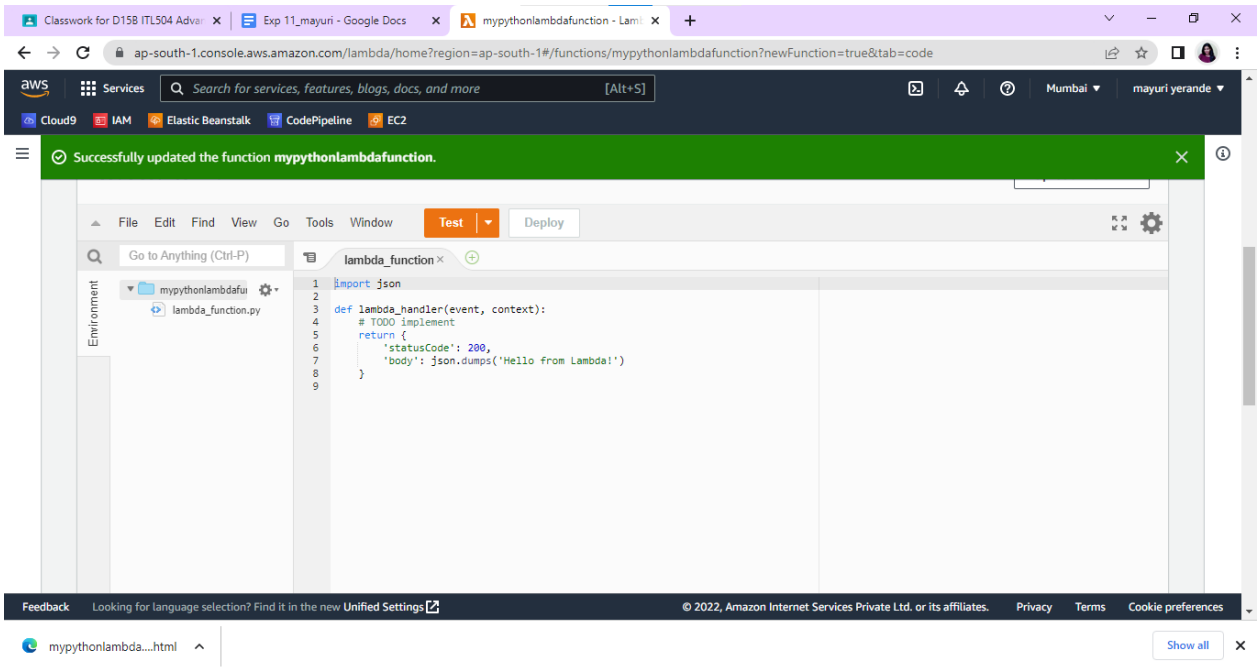
Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.



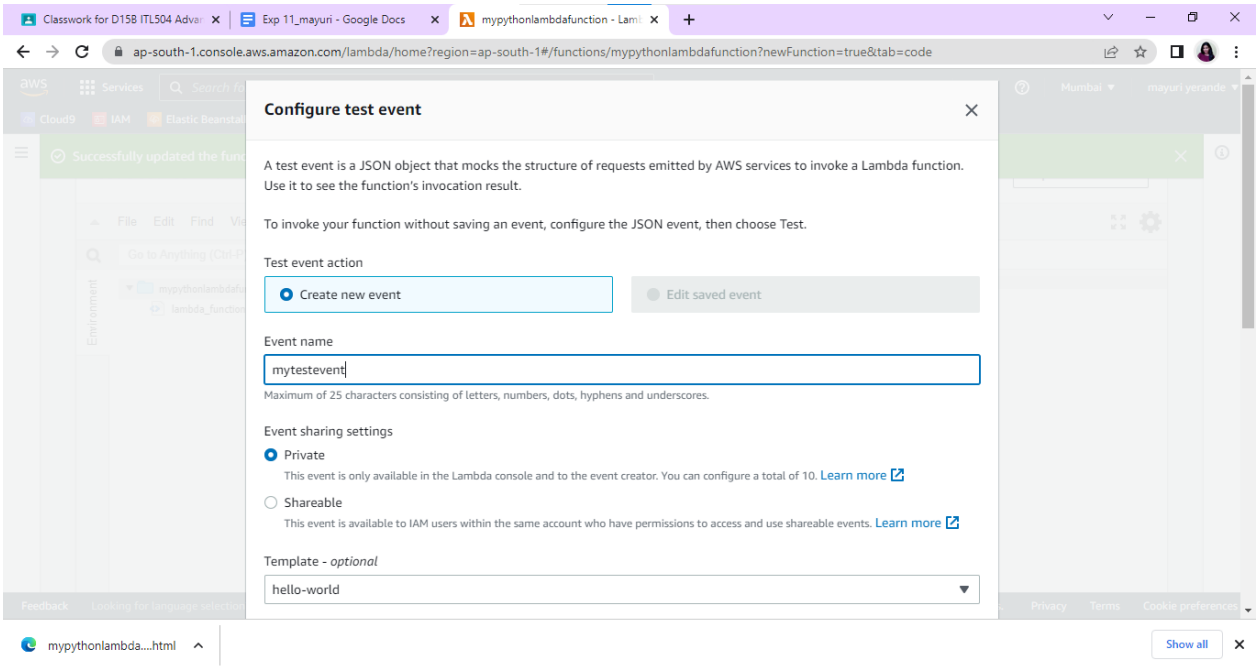


Step 5. You can make changes to your function inside the code editor. You can also upload a zip file of your function or upload one from an S3 bucket if needed.
Press Ctrl + S to save the file and click Deploy to deploy the changes.

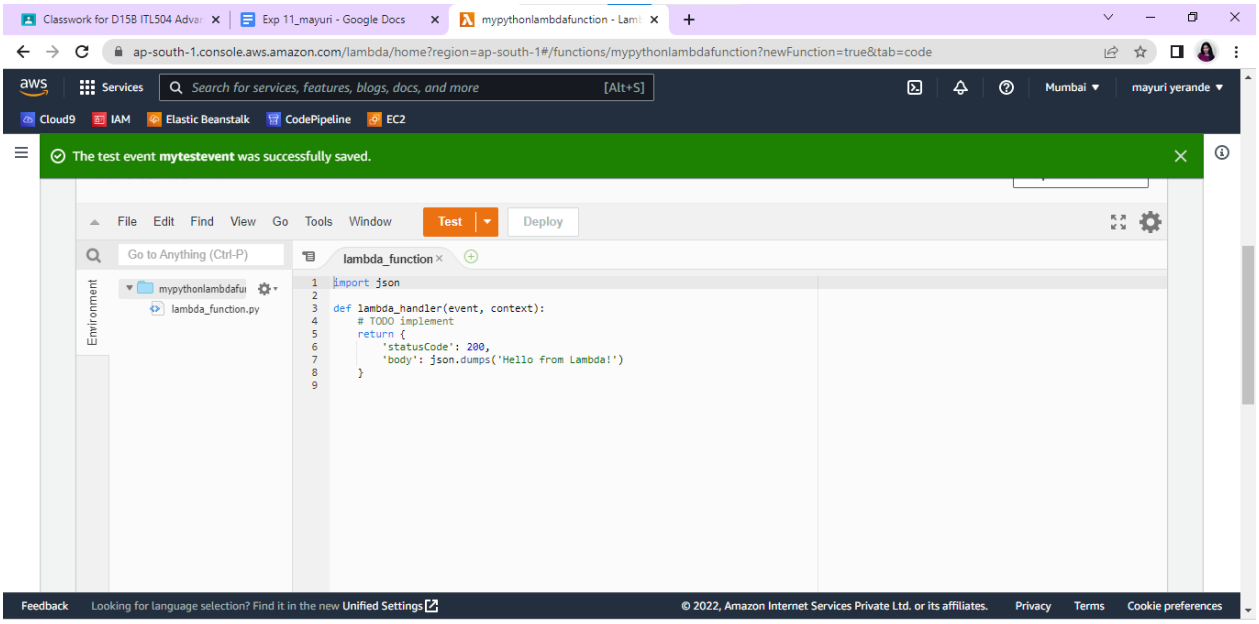


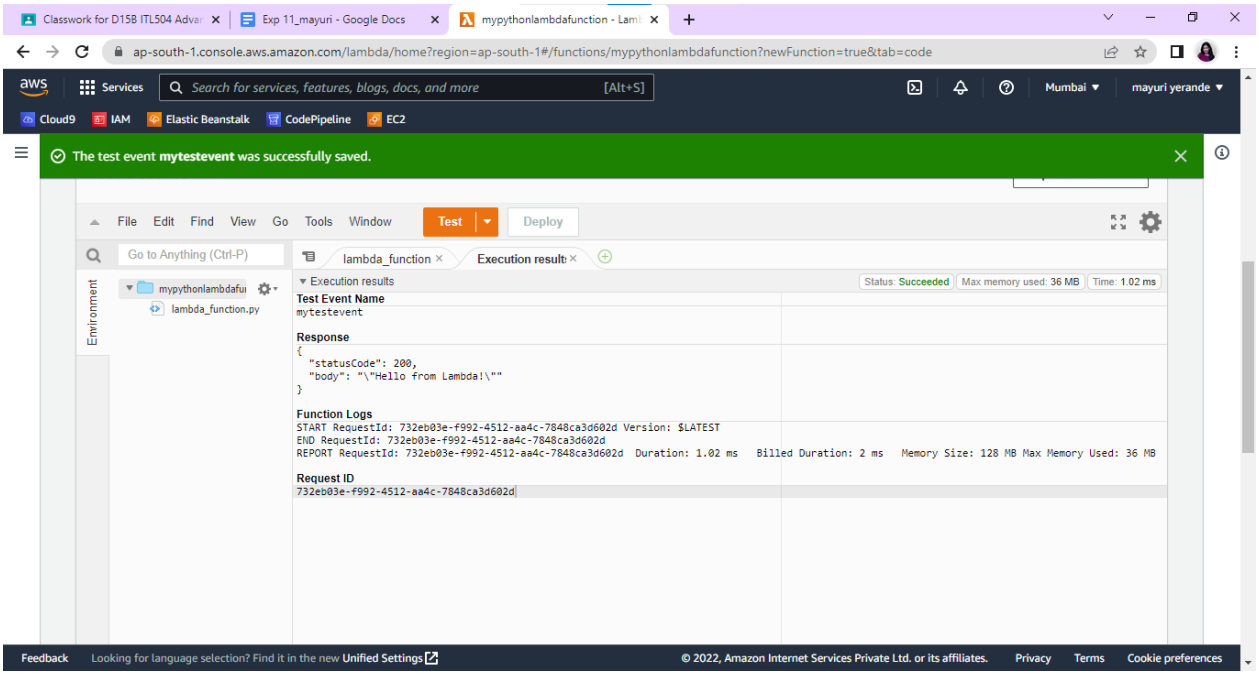


Step 6. Click on Test and you can change the configuration, like so. If you do not have anything in the request body, it is important to specify two curly braces as valid JSON, so make sure they are there.



Step 7. Now click on Test and you should be able to see the results.




Classwork for D15B ITL504 Advar
Exp 11_mayuri - Google Docs
mypythonlambdafunction - Lam
ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/functions/mypythonlambdafunction?newFunction=true&tab=code
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
mayuri yerande
Cloud9
IAM
Elastic Beanstalk
CodePipeline
EC2
The test event mytestevent was successfully saved.
File
Edit
Find
View
Go
Tools
Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Execution result
Environment
mypythonlambdafu
lambda_function.py
Execution results
Status: Succeeded
Max memory used: 36 MB
Time: 1.02 ms
Test Event Name
mytestevent
Response
{
"statusCode": 200,
"body": "\"Hello from Lambda!\""
}
Function Logs
START RequestId: 732eb03e-f992-4512-aa4c-7848ca3d602d Version: $LATEST
END RequestId: 732eb03e-f992-4512-aa4c-7848ca3d602d
REPORT RequestId: 732eb03e-f992-4512-aa4c-7848ca3d602d Duration: 1.02 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 36 MB
Request ID
732eb03e-f992-4512-aa4c-7848ca3d602d
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

Conclusion:-

In this experiment, we learned about AWS Lambda, and using it to create, deploy and test serverless function in the cloud.