Name: Hrishikesh Kumbhar

Div: D15A

Roll no: 32

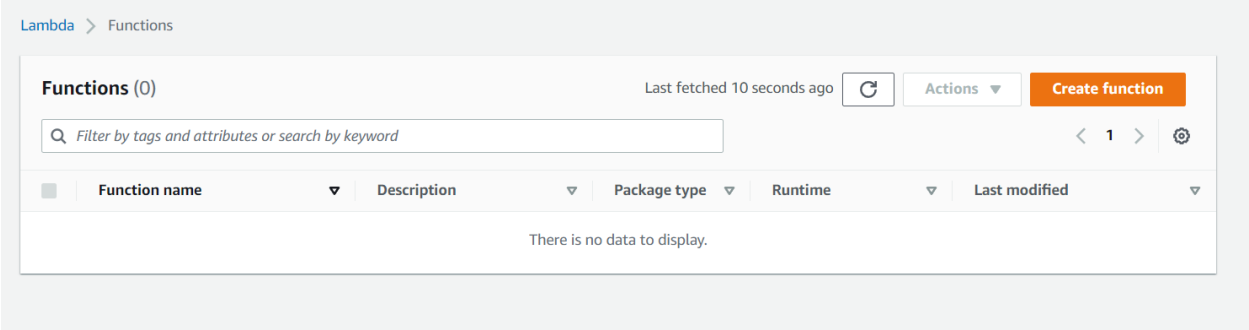
Sub: Advanced DevOps

Experiment No: 11

Aim: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java /Nodejs.

Steps:

Step 1. Open up the Lambda Console and click on the Create button.

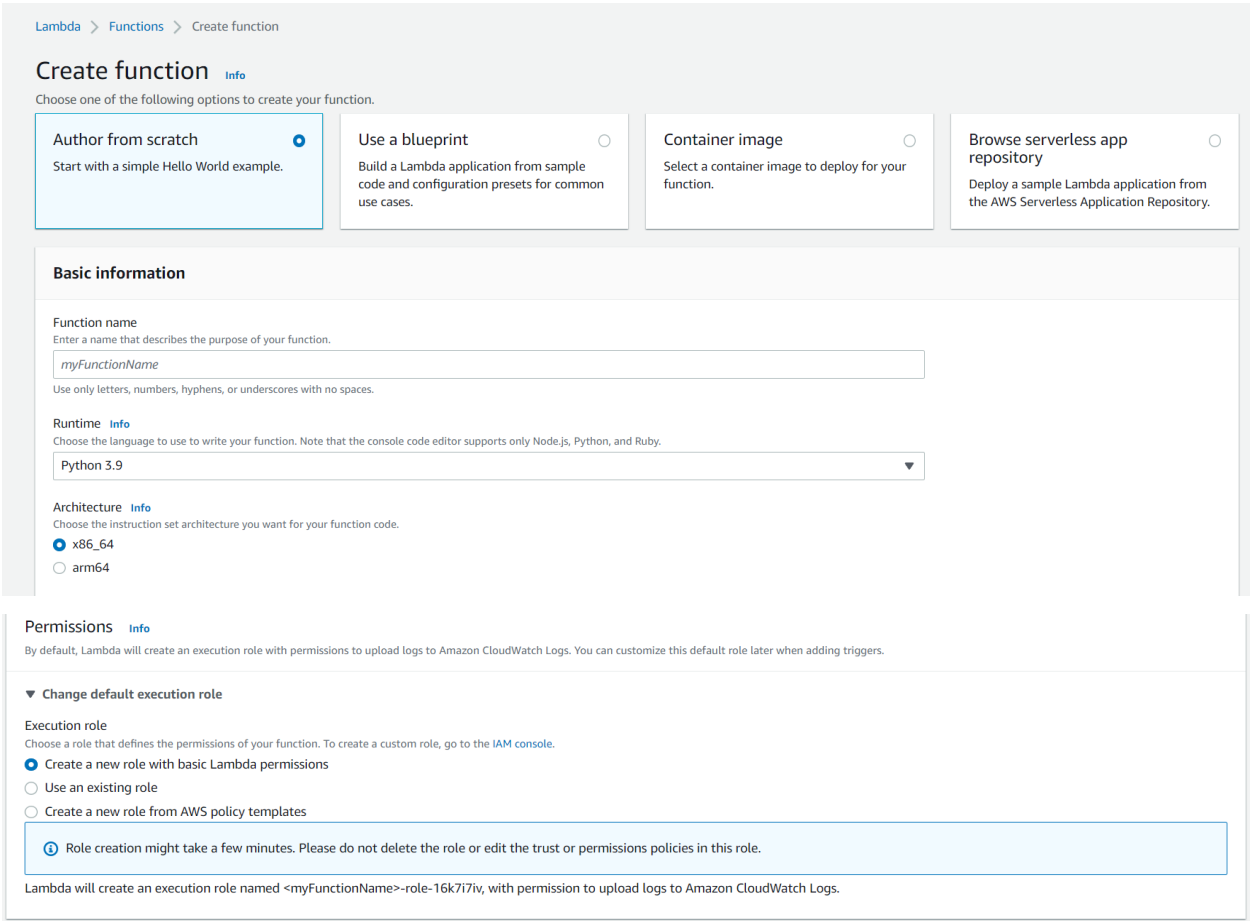


Be mindful of where you create your functions since Lambda is region-dependent.

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.

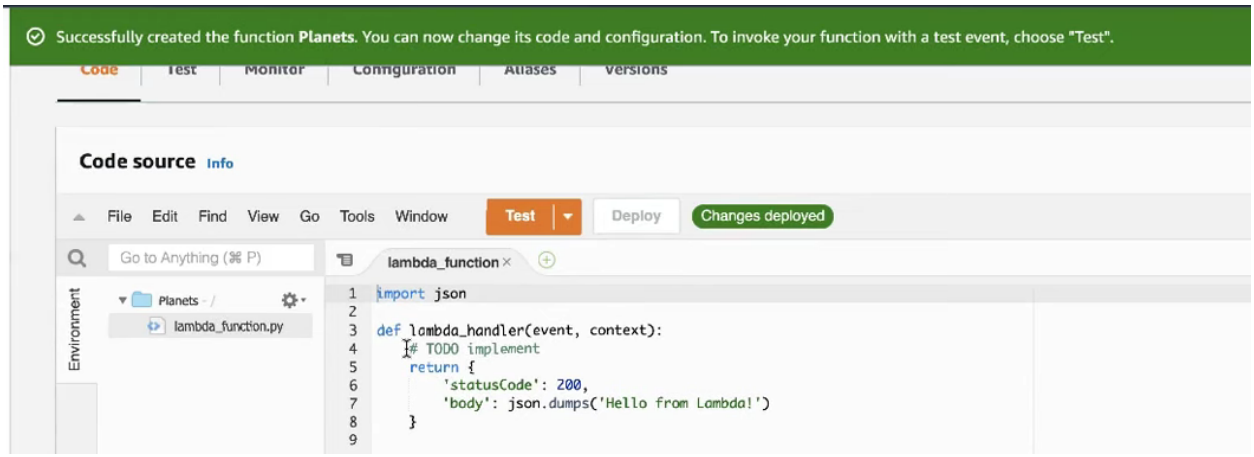
Then, choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones.

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

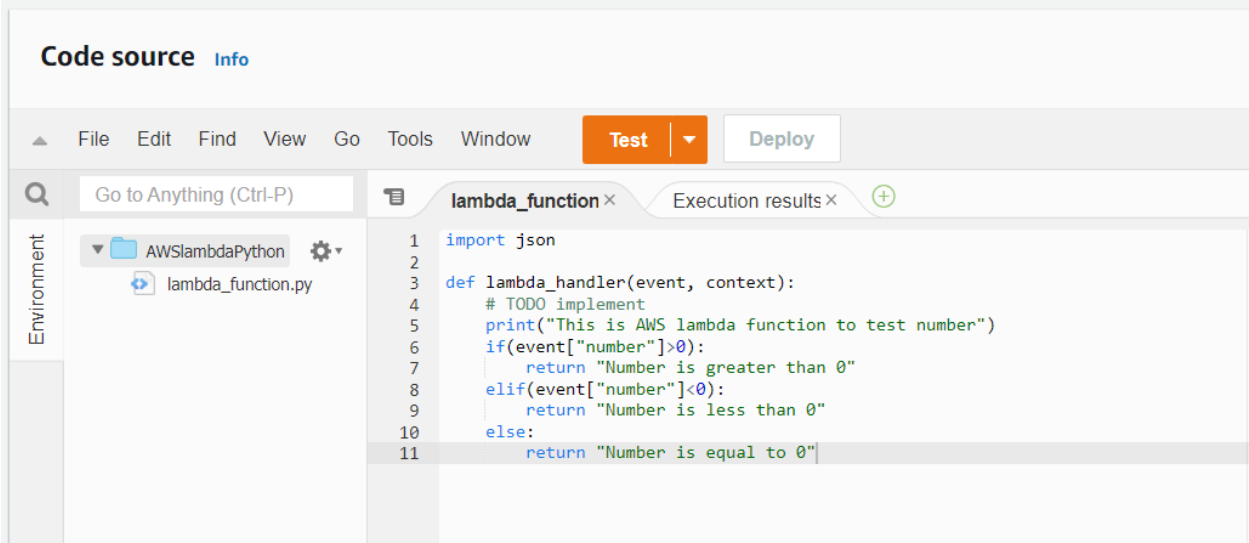


Click on the Create button.

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



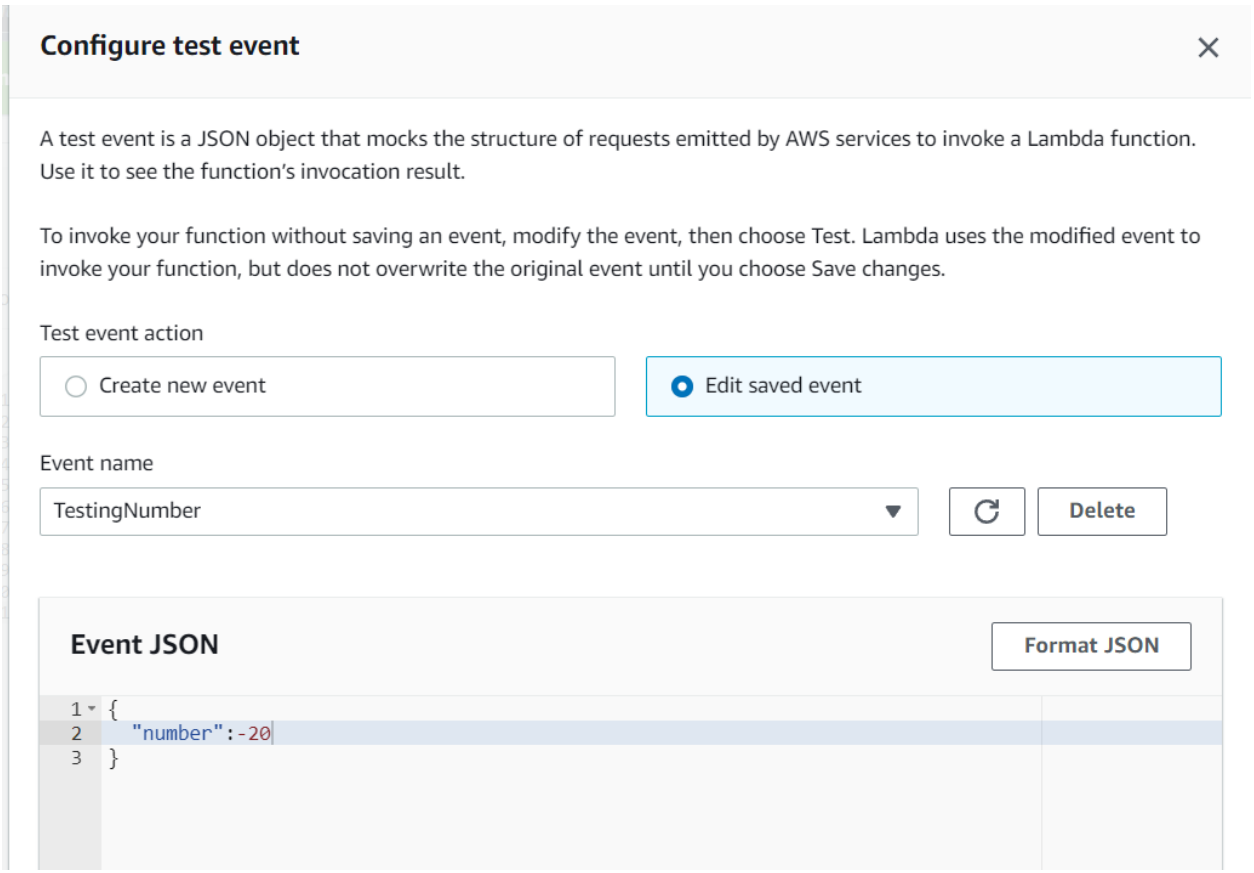
You can make changes to your function inside the code editor. You can also upload a zip file of your function or upload one from an S3 bucket if needed. Press Ctrl + S to save the file and click Deploy to deploy the changes.



Lamda_function.py

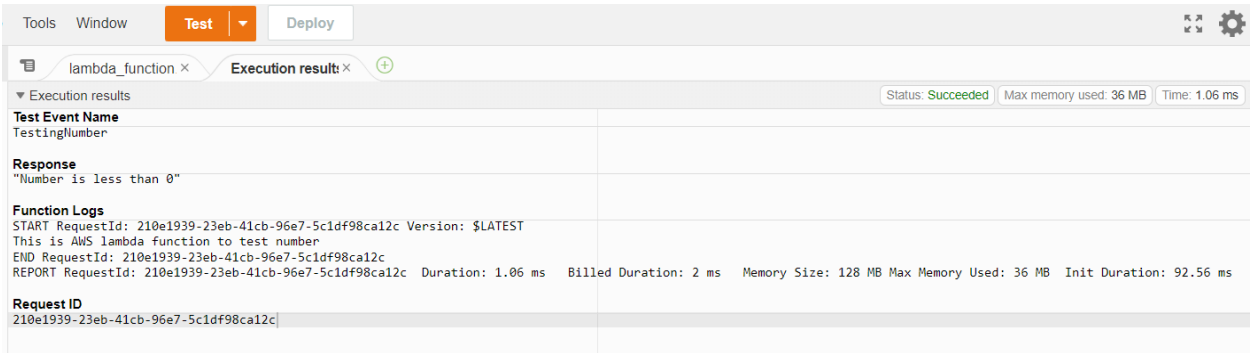
```
import json

def lambda_handler(event, context):
    # TODO implement
    print("This is AWS lambda function to test number")
    if(event["number"]>0):
        return "Number is greater than 0"
    elif(event["number"]<0):
        return "Number is greater than 0"
    else:
        return "Number is equal to 0"
```

Step 4: Configure test cases by clicking on test:



You can set event parameters in Event JSON. Here we have created an event TestingNumber to check whether the number is positive or negative or zero.
By setting key "number": <number> we can test the function for different inputs.
"number" key is imported via event["number"] in the lambda_function.py file.

Step 5: Now click on Test and you should be able to see the results.



Test event action

Create new event | Edit saved event

Event name

TestingNumber | Delete

Event JSON | Format JSON

```
1 {
2   "number":45
3 }
```

Tools | Window | Test | Deploy

lambda_function × | Execution result: ×

Execution results | Status: Succeeded | Max memory used: 37 MB | Time: 0.86 ms

Test Event Name

TestingNumber

Response

"Number is greater than 0"

Function Logs

```
START RequestId: 864ba921-047a-4c50-846f-cae4c0a6acd5 Version: $LATEST
This is AWS lambda function to test number
END RequestId: 864ba921-047a-4c50-846f-cae4c0a6acd5
REPORT RequestId: 864ba921-047a-4c50-846f-cae4c0a6acd5  Duration: 0.86 ms   Billed Duration: 1 ms   Memory Size: 128 MB Max Memory Used: 37 MB
```

Request ID

864ba921-047a-4c50-846f-cae4c0a6acd5

Test event action

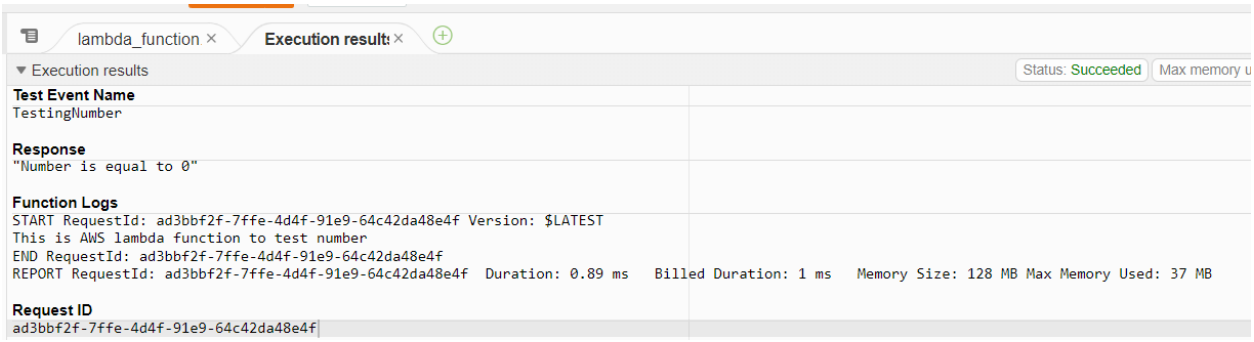
Create new event | Edit saved event

Event name

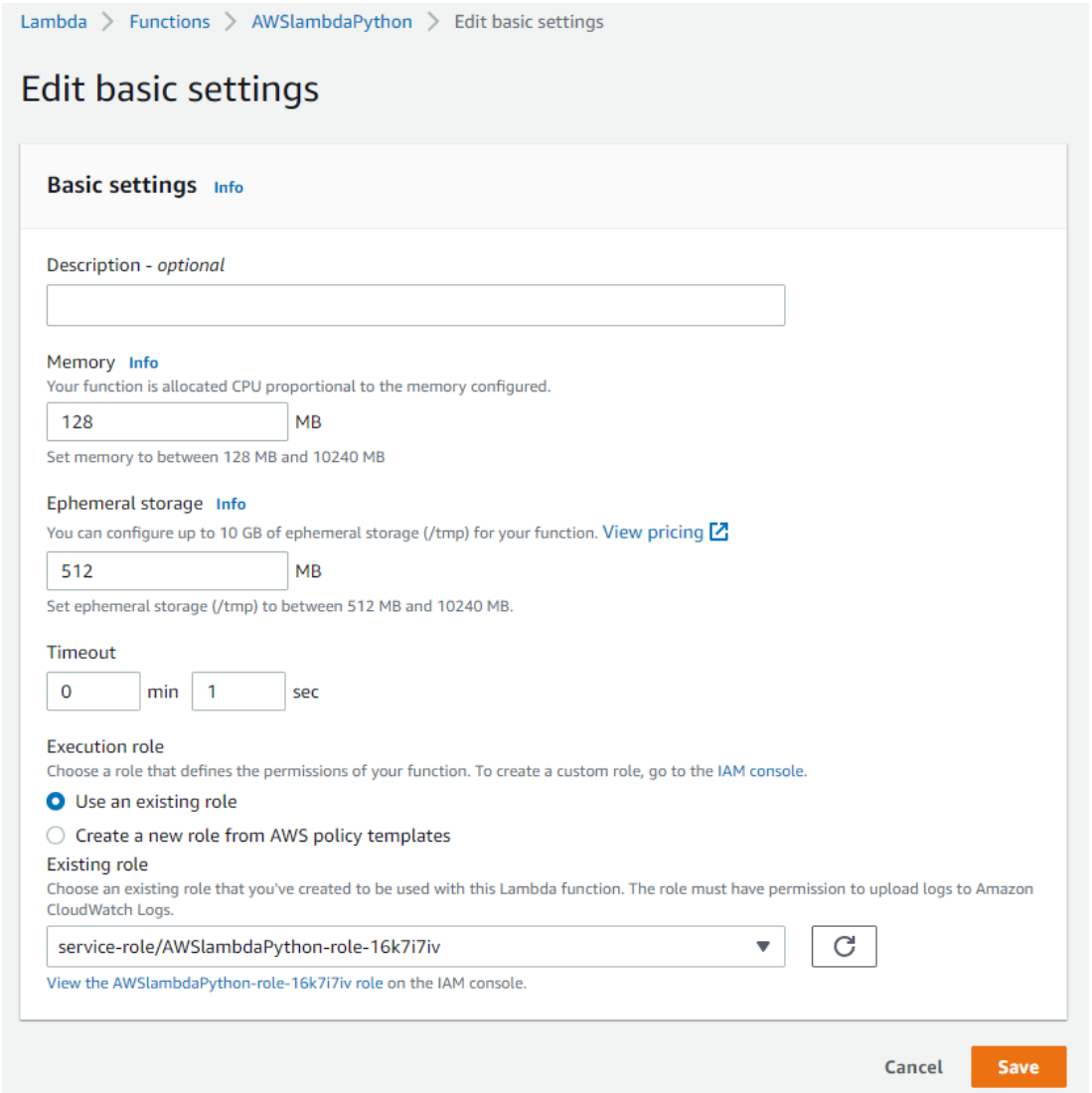
TestingNumber | Delete

Event JSON | Format JSON

```
1 {
2   "number":0
3 }
```



Step 6: 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.



Conclusion:

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