Name - Mikil Lalwani   D15B/37

Advance DevOps Lab

Experiment 11

Aim -

To understand AWS lambda, its workflow, various functions and create your first lambda function using Python/ Java/ Node JS.
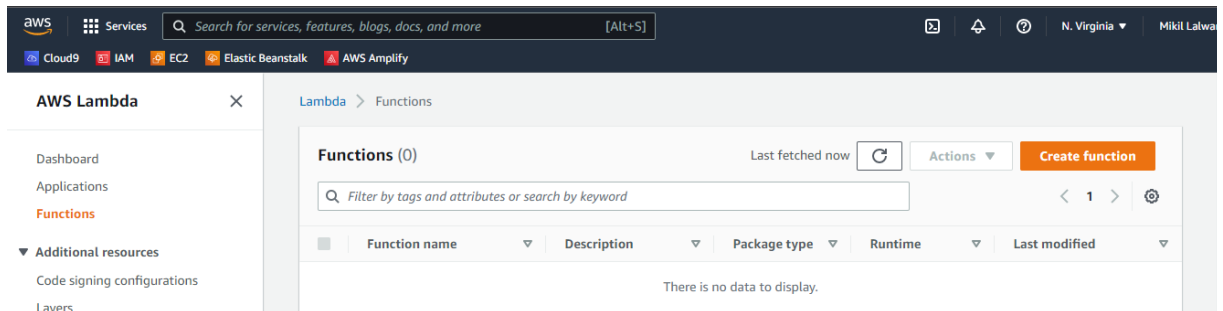
Theory -

AWS lambda is a serverless computing service provided by AWS. Users of lambda creates functions, self contained applications written in one of the supported languages in an efficent and flexible manner. The lambda function can perform any kind of computing task, from serving web pages and processing streams of data to calling API's and integrating with other AWS services.

The concept of serverless computing refers to not needing to maintain your own servers to run those functions. Lambda takes care of all the infrastructure for you. But serverless doesn't mean there are no servers involved, it just means that the servers, Os and networking is already taken care of.
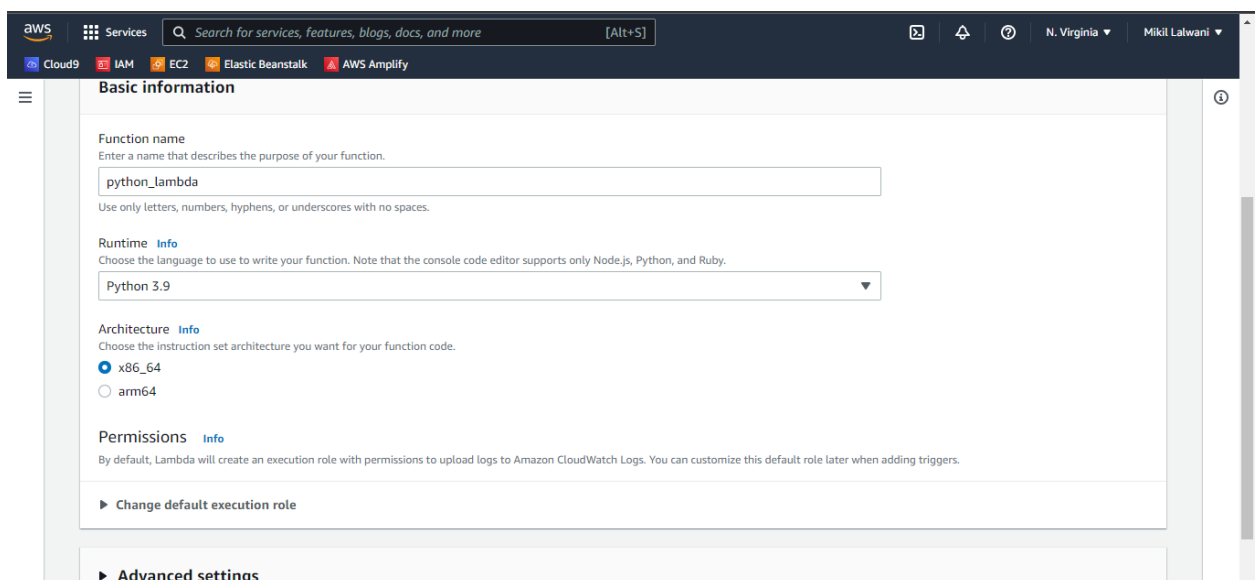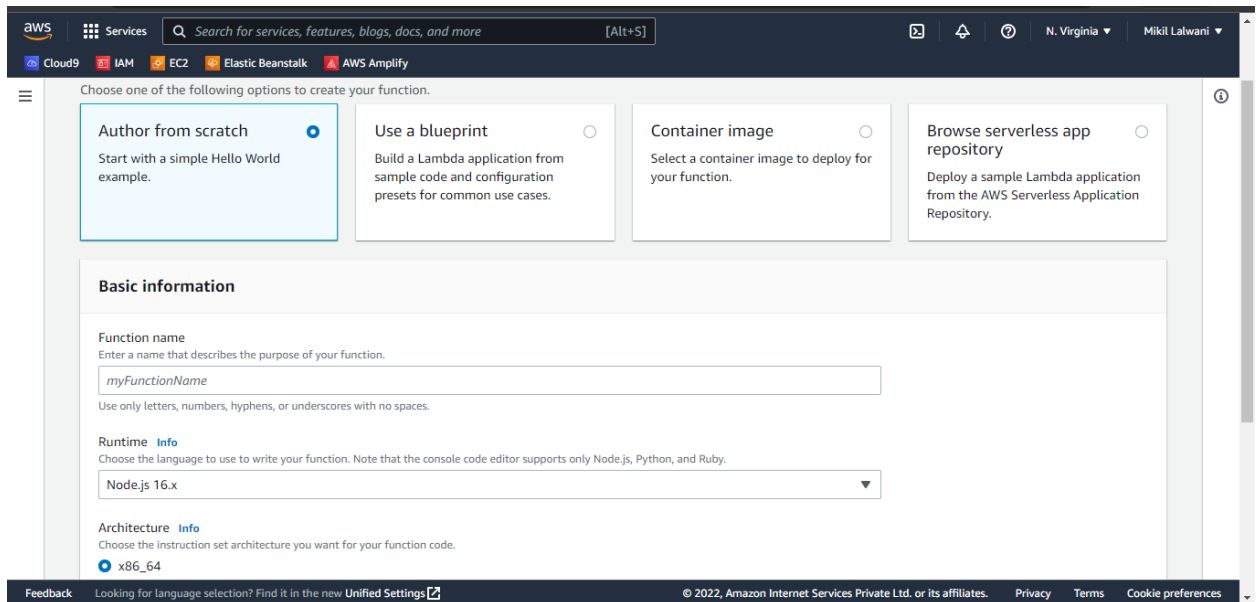
Steps-

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.

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.

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

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.

| Code | Test | Monitor | **Configuration** | Aliases | Versions |

General configuration

Triggers

Permissions

Destinations

Function URL

General configuration Info   [Edit]

Description   Memory   Ephemeral storage
-                    128 MB   512 MB

Timeout
0 min 3 sec

Basic settings Info

Description - *optional*

[   ]

Memory Info
Your function is allocated CPU proportional to the memory configured.

[128] MB

Set memory to between 128 MB and 10240 MB

Ephemeral storage Info
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing ↗

[512] MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.
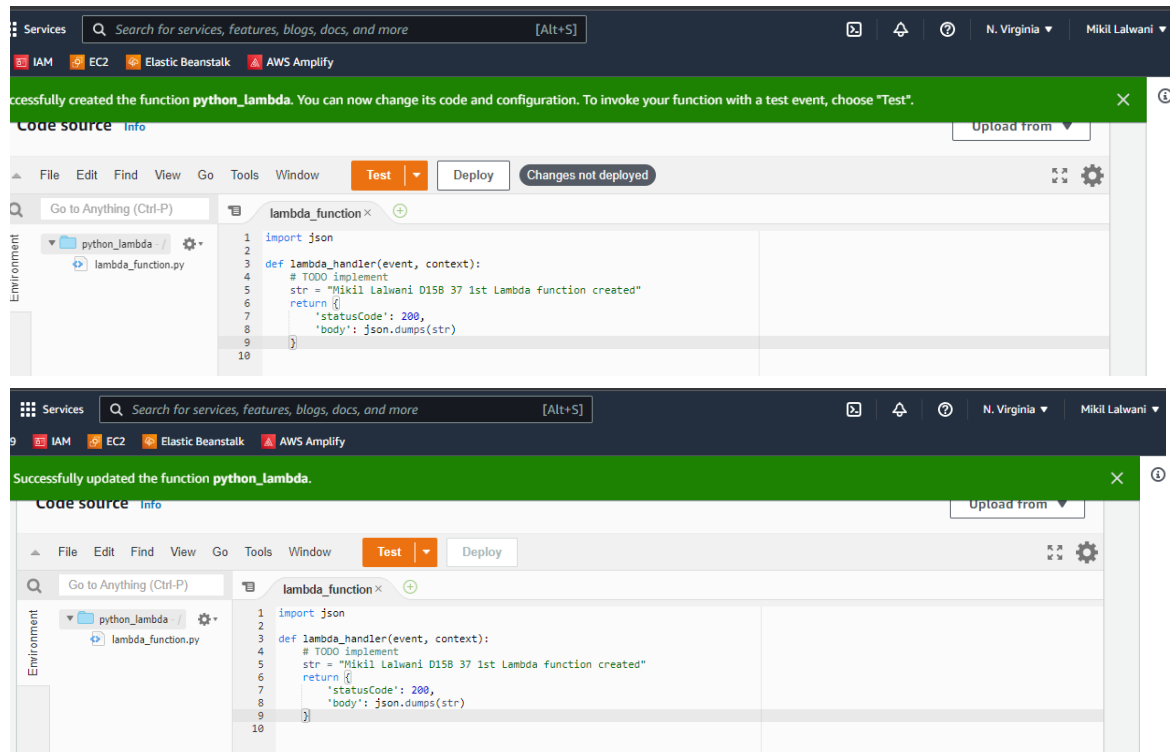
Timeout

[0] min [1] sec

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

◉ Use an existing role

○ Create a new role from AWS policy templates

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.

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.

Configure test event   ✕

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

● Create new event	○ Edit saved event

Event name

```
testevent
```

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

● Private
    This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more 🗗

○ Shareable
    This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more 🗗

Template - *optional*

```
hello-world                                                              ▼
```

Event JSON   Format JSON

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

Conclusion-

We learned to create, deploy and test serverless functions on AWS lambda.