PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)

Semester: V
Academic Year: 2024-25
Class / Branch: TE IT
Subject: Advanced Devops Lab (ADL)
Name of Instructor: Prof. Manjusha Kashilkar

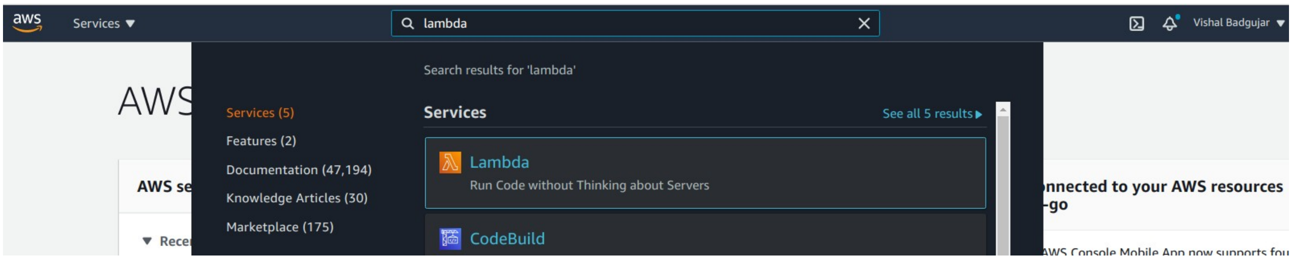
Name of Student: Chirag Jayesh Malde
Student ID: 22104186

EXPERIMENT NO. 11

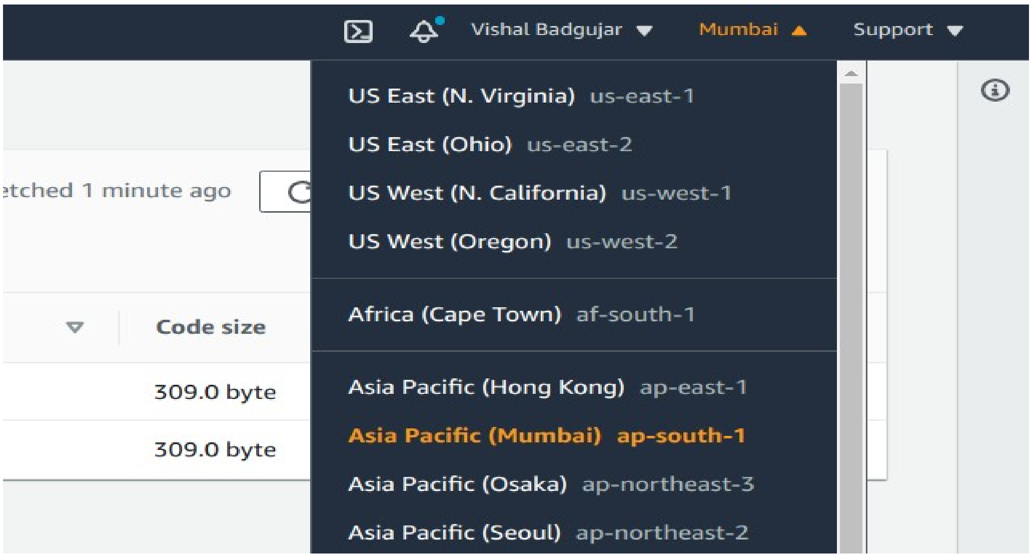
Aim: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

Steps: First Lambda functions using Python

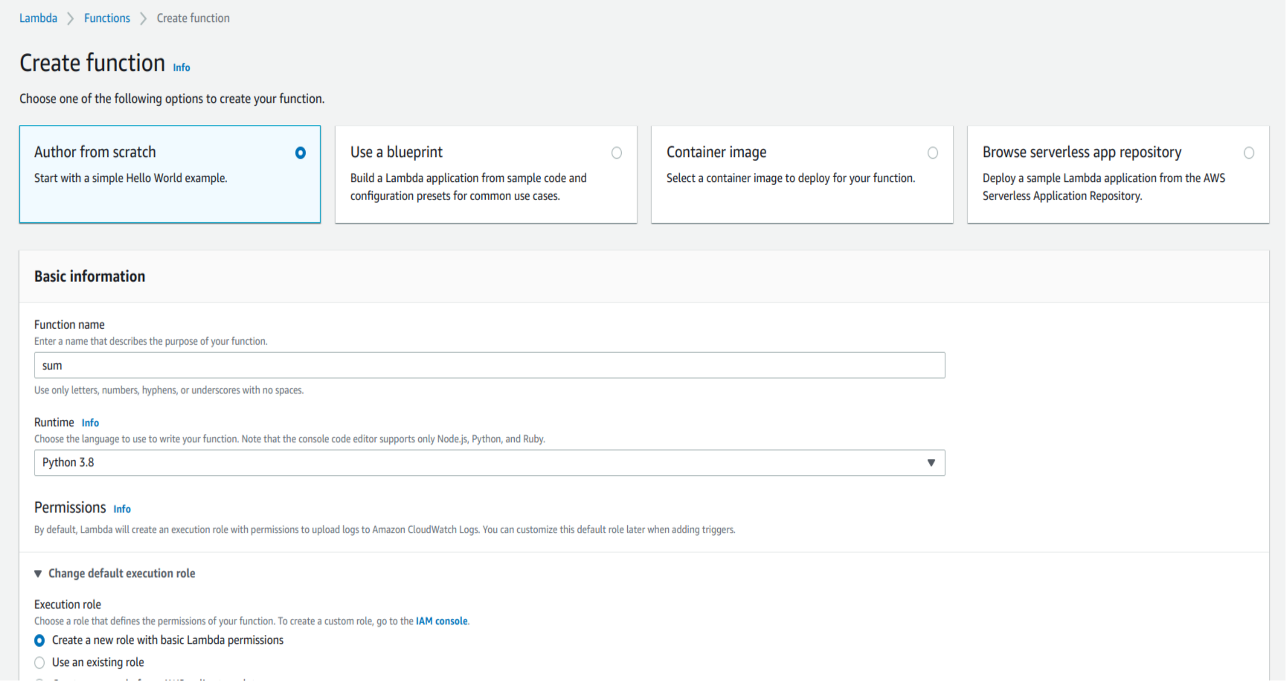
1. Open Aws Console and search for Lambda Service and open home screen of Lambda.



2. Choose region in which you need to create Lambda function as it is region specific.


Vishal Badgujar ▼
Mumbai ▲
Support ▼
etched 1 minute ago
Code size
309.0 byte
309.0 byte
US East (N. Virginia) us-east-1
US East (Ohio) us-east-2
US West (N. California) us-west-1
US West (Oregon) us-west-2
Africa (Cape Town) af-south-1
Asia Pacific (Hong Kong) ap-east-1
Asia Pacific (Mumbai) ap-south-1
Asia Pacific (Osaka) ap-northeast-3
Asia Pacific (Seoul) ap-northeast-2

3. Create sum as a Lambda Function in Python Language so select latest version of Python and choose role with basic Lambda Permission to allow cloudwatch for monitoring.



Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy for your function.

○ Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.
sum
Use only letters, numbers, hyphens, or underscores with no spaces.

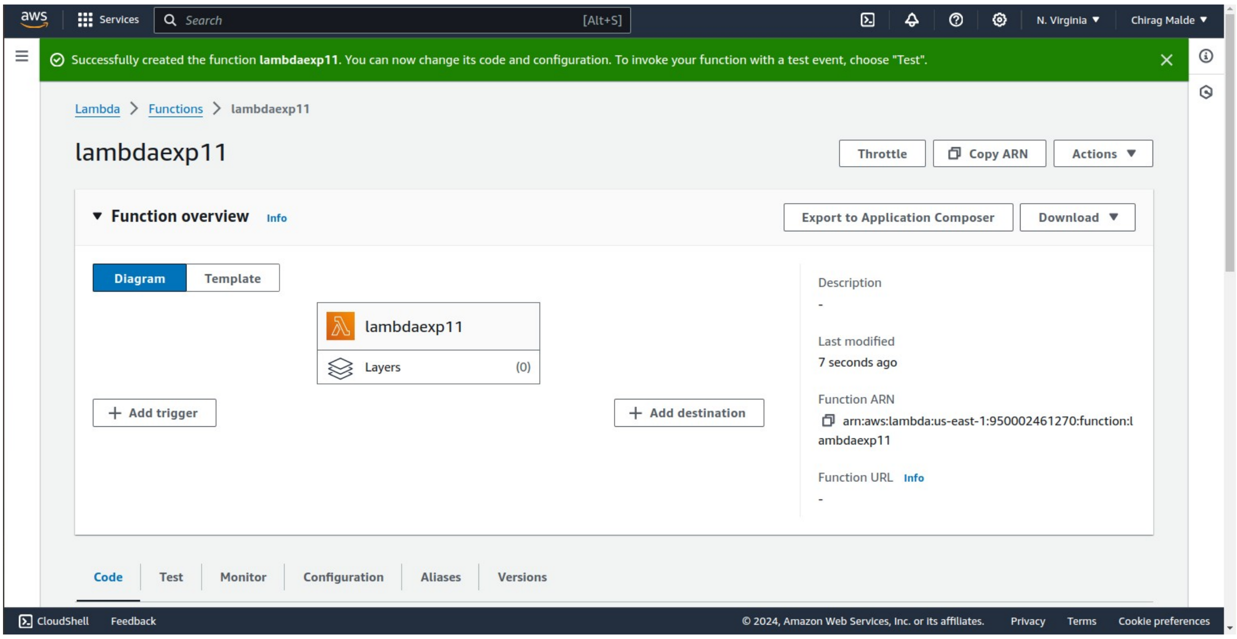
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.8 ▼

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
● Create a new role with basic Lambda permissions
○ Use an existing role

4. Lambda sum function is created successfully



Successfully created the function **lambdaexp11**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lambdaexp11

lambdaexp11

▼ Function overview Info

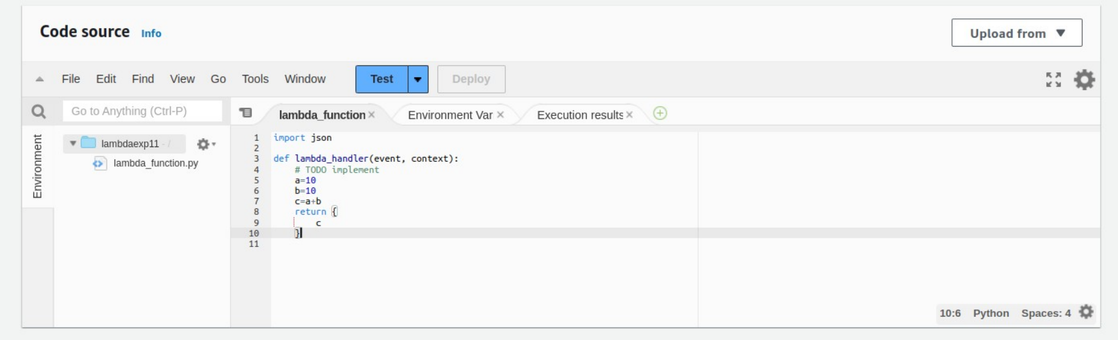
Description
-

Last modified
7 seconds ago

Function ARN
arn:aws:lambda:us-east-1:950002461270:function:lambdaexp11

Function URL Info
-

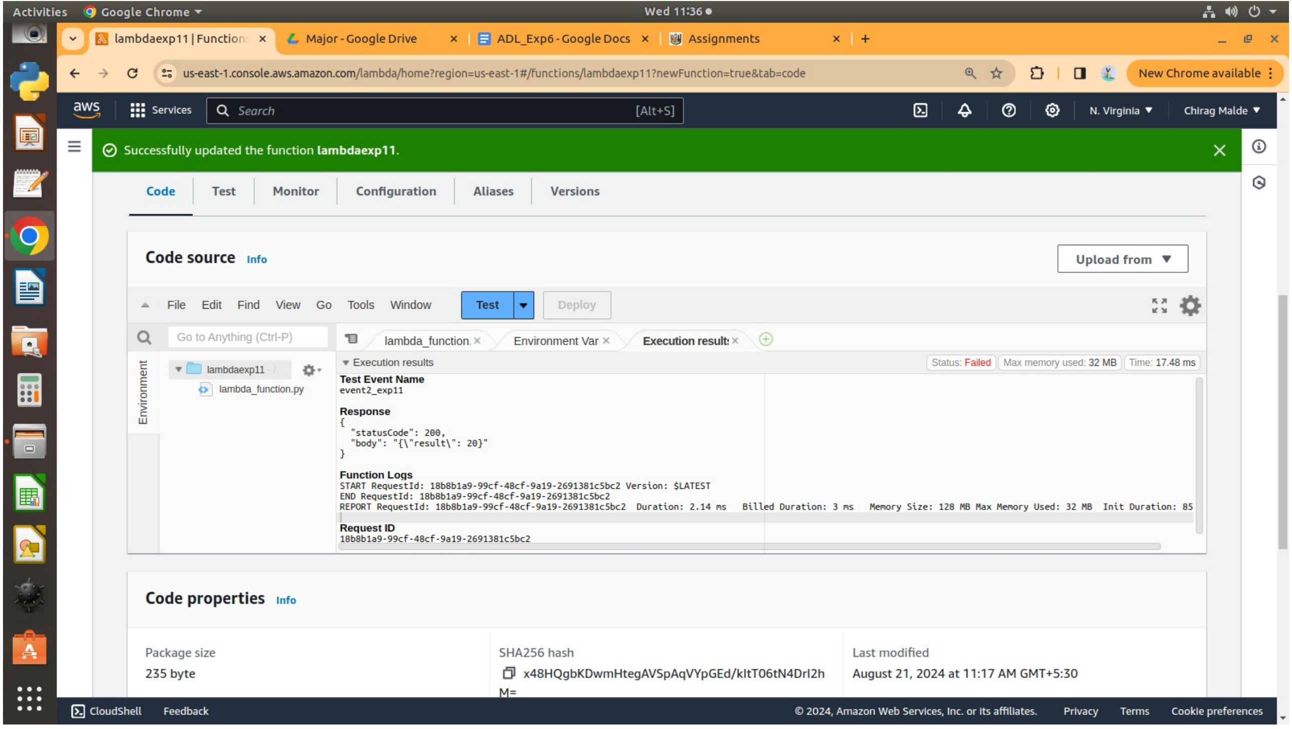
5. Write a sample python code for sum of two numbers:



```
import json

def lambda_handler(event, context):
    # TODO implement
    a=10
    b=10
    c=a+b
    return {
        c
    }
```

6. Configure Test Event in Json Format



Successfully updated the function **lambdaexp11**.

▾ Execution results

Status: Failed | Max memory used: 32 MB | Time: 17.48 ms

Test Event Name

```
event2_exp11
```

Response

```
{
  "statusCode": 200,
  "body": "{\"result\": 20}"
}
```

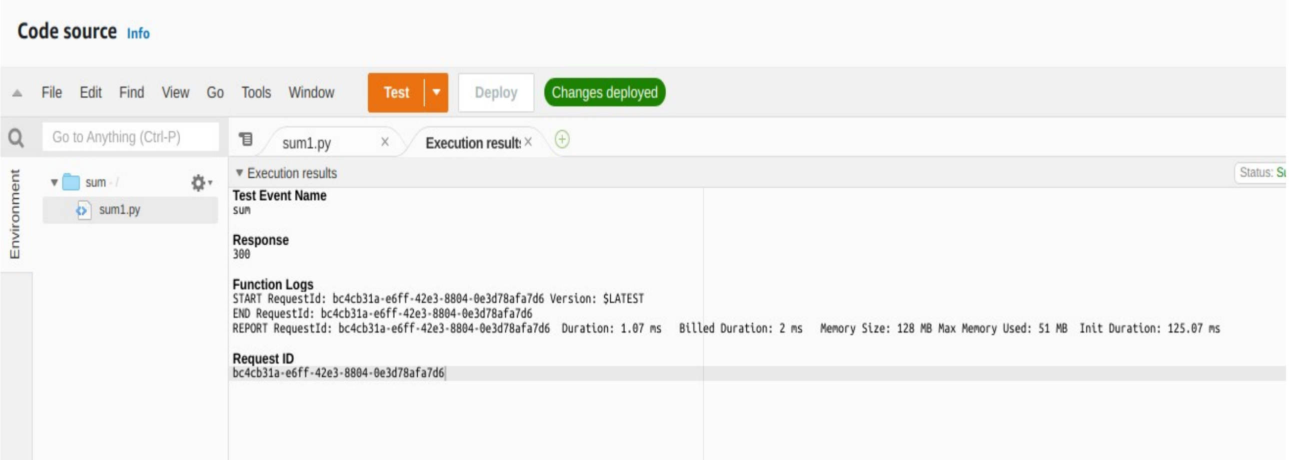
Function Logs

```
START RequestId: 18b8b1a9-99cf-48cf-9a19-2691381c5bc2 Version: $LATEST
END RequestId: 18b8b1a9-99cf-48cf-9a19-2691381c5bc2
REPORT RequestId: 18b8b1a9-99cf-48cf-9a19-2691381c5bc2  Duration: 2.14 ms   Billed Duration: 3 ms   Memory Size: 128 MB Max Memory Used: 32 MB  Init Duration: 85
```

Request ID

```
18b8b1a9-99cf-48cf-9a19-2691381c5bc2
```

Package size	SHA256 hash	Last modified
235 byte	x48HQgbKDwmHtegAVSpAqVYpGEd/kItT06tN4DrI2hM=	August 21, 2024 at 11:17 AM GMT+5:30



▾ Execution results

Test Event Name

```
sum
```

Response

```
300
```

Function Logs

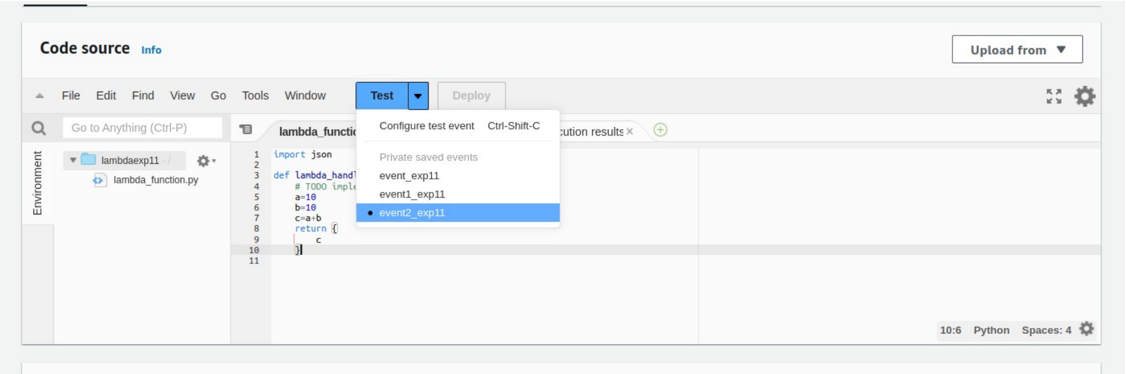
```
START RequestId: bc4cb31a-e6ff-42e3-8804-0e3d78afa7d6 Version: $LATEST
END RequestId: bc4cb31a-e6ff-42e3-8804-0e3d78afa7d6
REPORT RequestId: bc4cb31a-e6ff-42e3-8804-0e3d78afa7d6  Duration: 1.07 ms   Billed Duration: 2 ms   Memory Size: 128 MB Max Memory Used: 51 MB  Init Duration: 125.07 ms
```

Request ID

```
bc4cb31a-e6ff-42e3-8804-0e3d78afa7d6
```

Write a sample Second sample python Code:

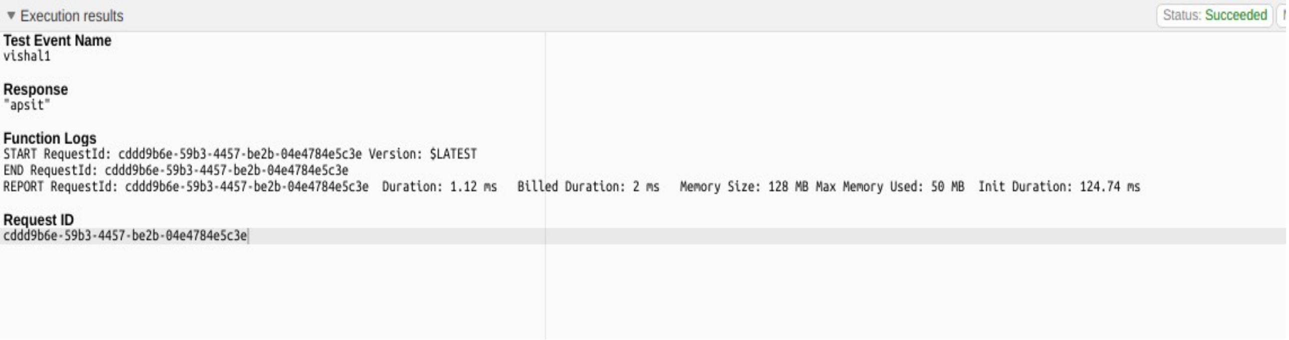
Configure Test Event



```
import json

def lambda_hand
    # TODO imple
    a=10
    b=10
    c=a+b
    return {
        c
    }
```

If condition met returns a value as apsit



```
Test Event Name
vishal1

Response
"apsit"

Function Logs
START RequestId: cddd9b6e-59b3-4457-be2b-04e4784e5c3e Version: $LATEST
END RequestId: cddd9b6e-59b3-4457-be2b-04e4784e5c3e
REPORT RequestId: cddd9b6e-59b3-4457-be2b-04e4784e5c3e  Duration: 1.12 ms   Billed Duration: 2 ms   Memory Size: 128 MB Max Memory Used: 50 MB  Init Duration: 124.74 ms

Request ID
cddd9b6e-59b3-4457-be2b-04e4784e5c3e
```

Conclusion: The experiment highlighted the simplicity of deploying AWS Lambda functions using Python, showcasing efficient serverless execution and integration with AWS services. This approach eliminates the need for server management while enabling flexible and responsive code execution.