| | |
|---|---|
| **Semester: V** | **Name of Student:** |
| **Academic Year: 2022-23** | **Student ID:** |
| **Class / Branch: TE IT** | |
| **Subject: Advanced Devops Lab (ADL)** | |
| **Name of Instructor: Prof. Manjusha Kashilkar** | |

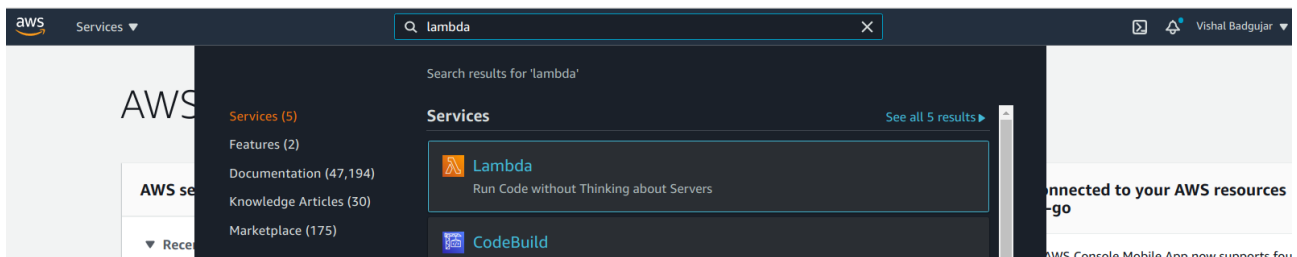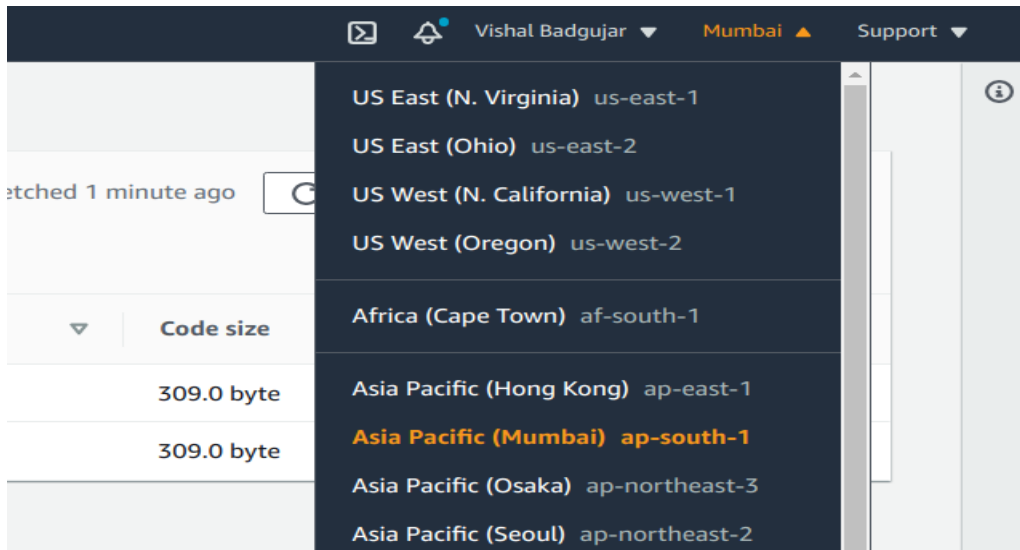## EXPERIMENT NO. 11

**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.**

**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.**



**4. Lambda sum function is created successfully**
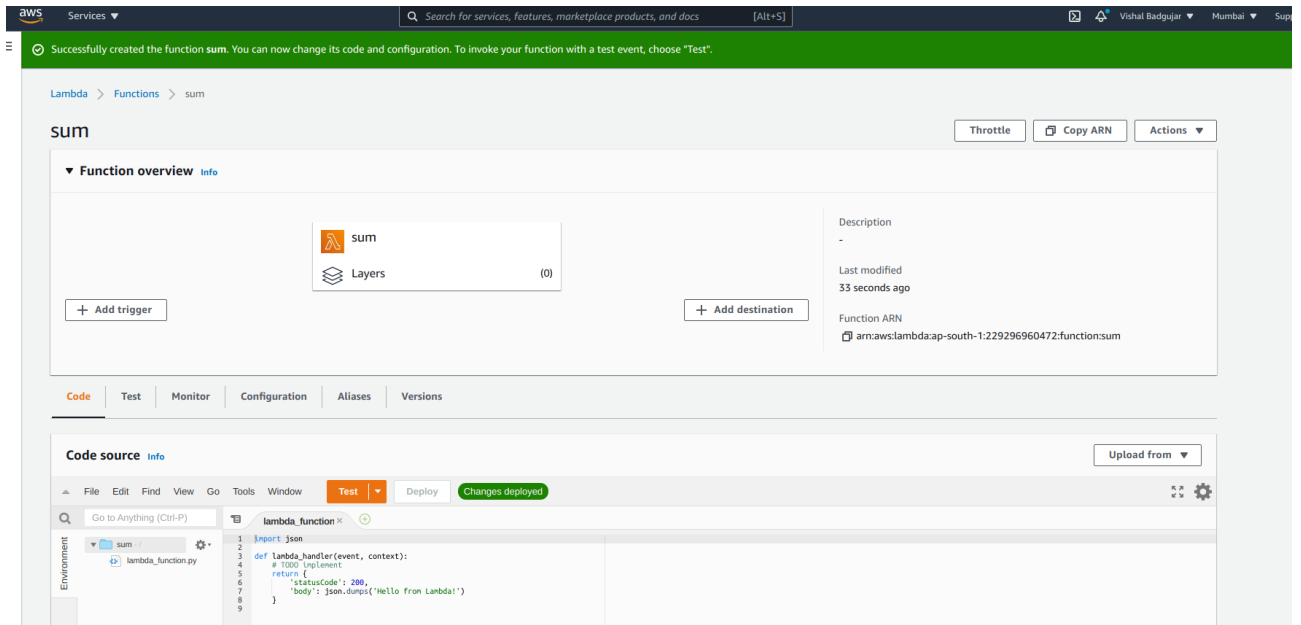
## 5. Write a sample python code for sum of two numbers:



```python
import json

def lambda_handler(event, context):
    first_number = 100
    second_number = 200
    sum = first_number + second_number
    return sum
```
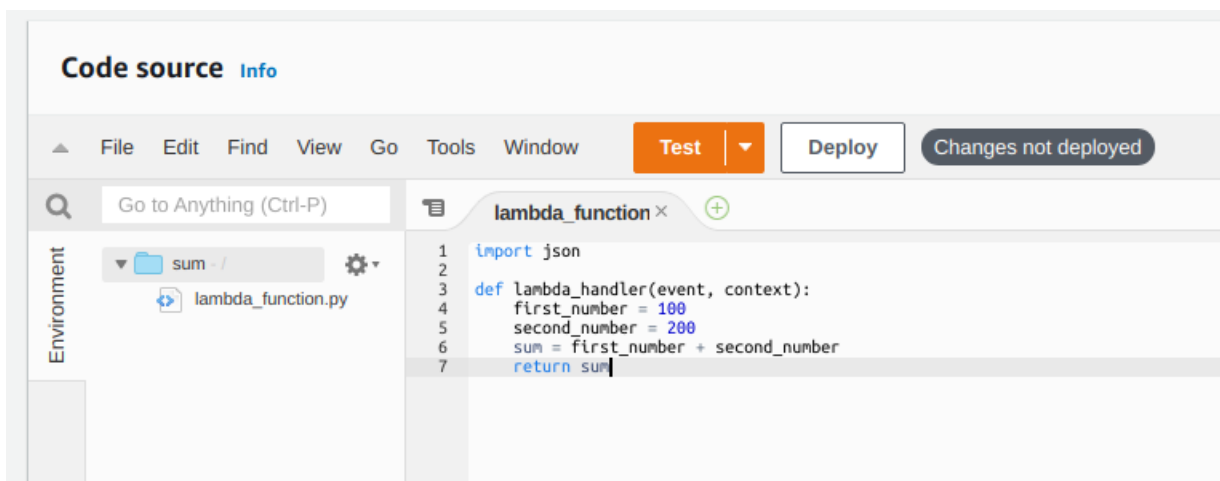
## 6. Configure Test Event in Json Format

## Configure test event                                                              ✕

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

🔵 Create new test event

⚪ Edit saved test events

Event template

```
hello-world                                                                    ▼
```

Event name

```
mytest1
```

```
1 ▾ {
2
3 }
```

---

## Code source  Info

| ▲ | File | Edit | Find | View | Go | Tools | Window | **Test** ▼ | Deploy | Changes deployed |

🔍 Go to Anything (Ctrl-P)    📄  sum1.py   ✕   Execution result: ✕   ⊕

▼ Execution results                                                                   Status: S...
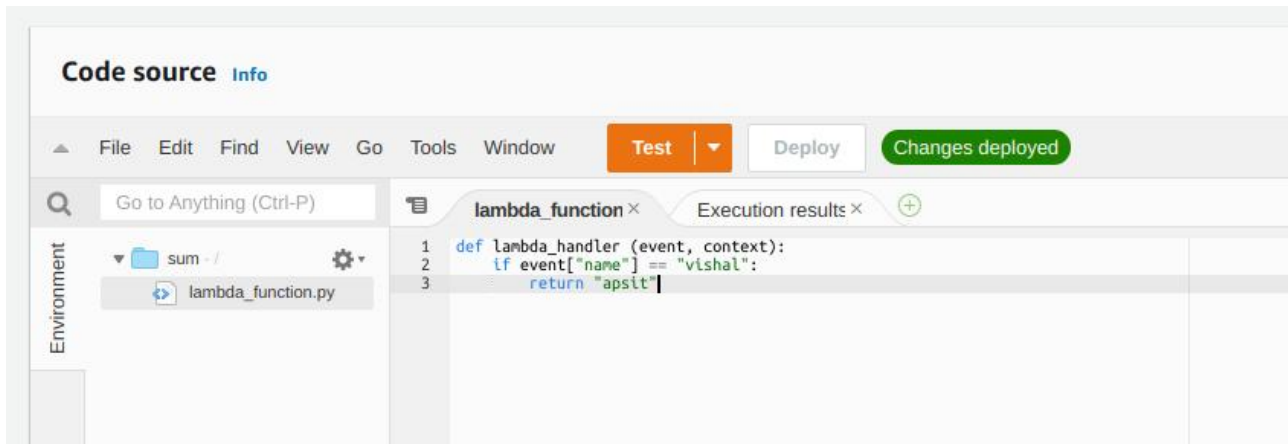
▼ 📁 sum ⚙▾

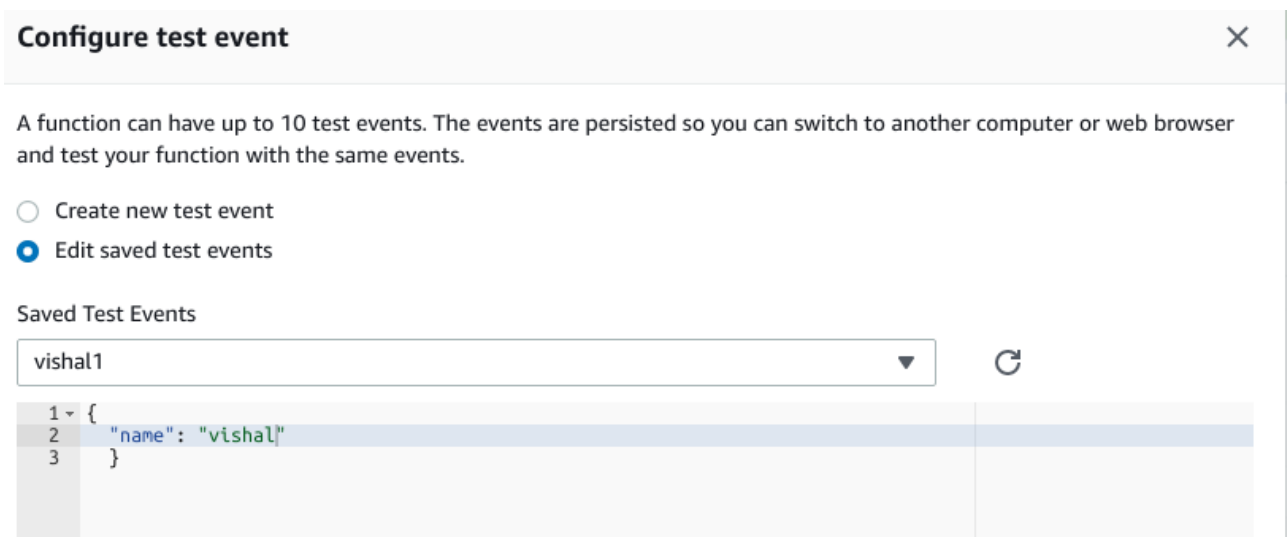  ◇ sum1.py

**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



**If condition met returns a value as apsit**

▼ Execution results

Status: **Succeeded**

**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: Write your own findings.**