



VASANTDADA PATIL PRATISHTHAN'S COLLEGE OF ENGINEERING AND VISUAL ARTS

ISO 9001:2015 Certified Institute

Department of Information Technology

NBA Accredited Course (Dated 01/07/2024 to 30/06/2027)

EXPERIMENT - 10

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

Theory:

AWS Lambda:

AWS Lambda is a serverless compute service provided by Amazon Web Services (AWS). It allows developers to run code in response to events without provisioning or managing servers. This event-driven, serverless Function as a Service (FaaS) enables rapid and cost-effective modern applications development.

Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, all you need to do is supply your code in one of the language runtimes that Lambda supports.

You organize your code into Lambda functions. The Lambda service runs your function only when needed and scales automatically. You only pay for the compute time that you consume—there is no charge when your code is not running.

Workflow:

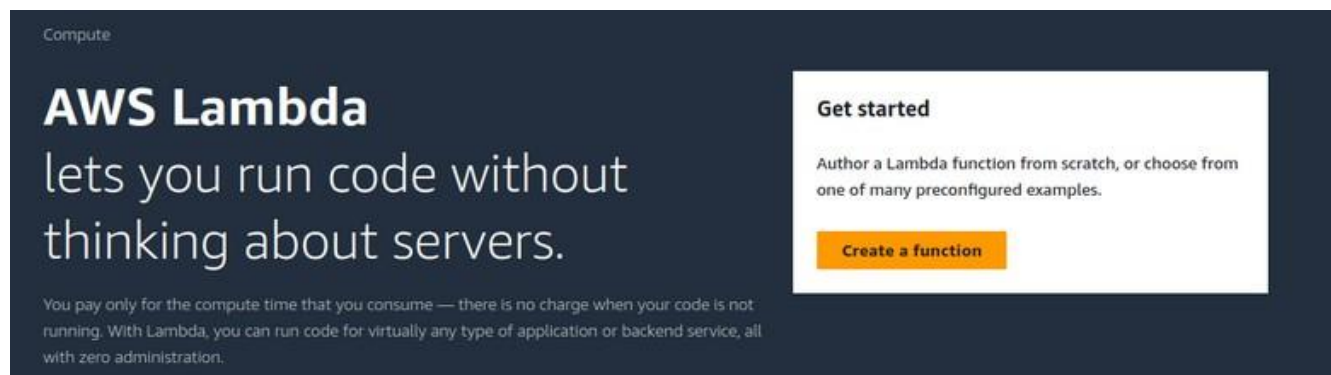
- **Function:** Your code that Lambda runs
- **Event Source:** AWS service or custom application that triggers your function
- **Trigger:** Association between an event source and a Lambda function
- **Execution Environment:** Secure and isolated runtime environment for your Lambda function

Some examples of various AWS Lambda functions:

1. **Image Resizing:** A Lambda function can be triggered by an Amazon S3 object upload and resize images to different dimensions, making them suitable for different use cases (e.g., thumbnails, banners).
2. **Real-time Data Processing:** Lambda functions can be used to process real-time data from IoT devices, such as sensor readings, and trigger actions like sending notifications or updating databases.
3. **API Gateway Integration:** Lambda functions can be used as the backend for API Gateway, handling API requests and responses, and integrating with other AWS services like DynamoDB or S3.
4. **Chatbot or Virtual Assistant:** A Lambda function can be triggered by voice or text input and respond with relevant information or actions, integrating with services like Amazon Lex or Amazon Comprehend.
5. **Email Processing:** Lambda functions can be used to process and analyze email attachments, such as extracting metadata or sending notifications to team members.
6. **Log Processing:** Lambda functions can be triggered by Amazon CloudWatch Logs and process log data, such as aggregating metrics, detecting anomalies, or sending alerts.
7. **Webhook Processing:** Lambda functions can be used to process webhooks from third-party services, such as payment gateways or social media platforms, and trigger actions in your application.

STEPS:

1. Sign in to the AWS Management Console, and search for Aws lambda service.



2. Click create a function

Enter the details as given below. → Create

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.

Basic information

Function name
Enter a name that describes the purpose of your function.

myFirstFunction

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 3.12

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64

☐ arm64

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

Lambda > Functions > myFirstFunction

myFirstFunction

Throttle Copy ARN Actions

▼ **Function overview** [Info](#)

Export to Application Composer Download

Diagram Template

myFirstFunction

Layers (0)

+ Add trigger + Add destination

Description -

Last modified 1 minute ago

Function ARN
arn:aws:lambda:us-east-1:283527553883:function:myFirstFunction

Function URL [Info](#)

Scroll down to code section,

Code Test Monitor Configuration Aliases Versions

Code source [Info](#)

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

myFirstFunction

lambda_function.py

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
```

Now change the default code with this:

```
import json
def lambda_handler(event, context):

    print("Received event: " + json.dumps(event, indent=2))
    message = 'Hello from Lambda!'
    return {
        'statusCode': 200,
        'body': json.dumps(message)
    }
```

Then, click on deploy, After it is deployed click on “Test”

Test event action

☒ Create new event ☐ Edit saved event

Event name

myTestEvent

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

myTestEvent

Event JSON

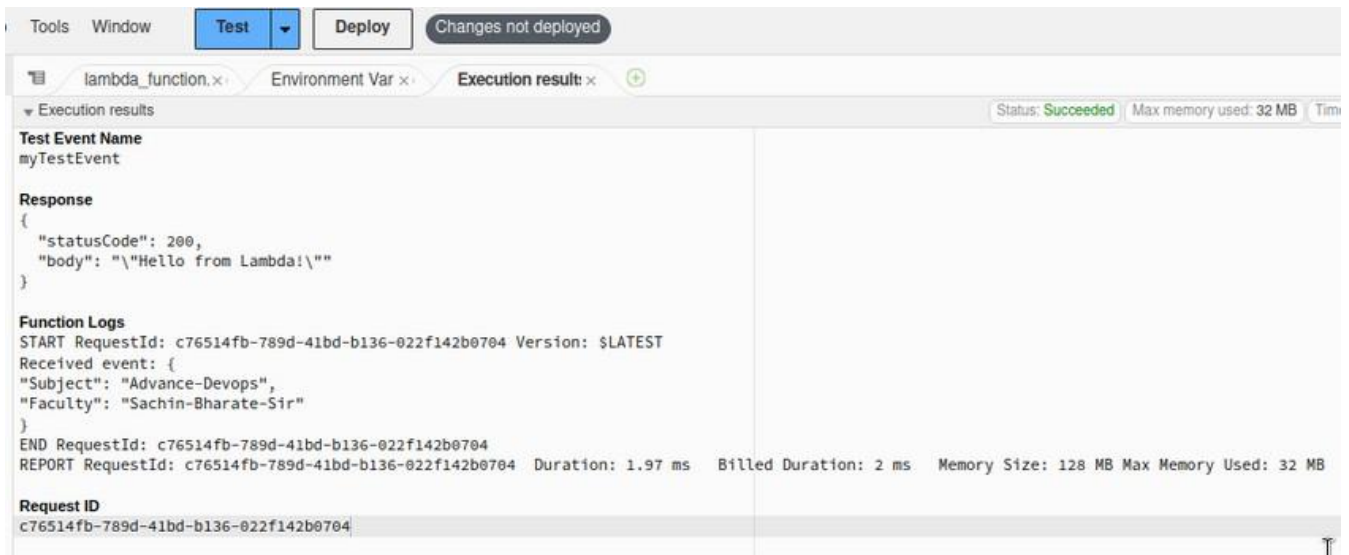
Format JSON

```
1 {
2   "Subject": "Advance-Devops",
3   "Faculty": "Sachin-Bharate-Sir"
4 }
```

Give name to event, You can change event json data accordingly, then click save.

Then again click on test and you will see the output of it.

Note: You can create multiple events for testing purpose.



The output contains that json and it is exact output of our code.

Conclusion: Thus, we have successfully understood AWS Lambda, its workflow, various functions and created our first Lambda functions using Python / Java / Node js.