

# Lesson 10 Lambda & CloudWatch

Michael Yang



# What is Serverless

- ▶ Serverless is a cloud-native development model that allows developers to build and run applications without having to manage servers.
- ▶ We define a serverless system as one that meets the following criteria:
  1. No need to manage and maintain virtual machines
  2. Fully managed service offering scalability and high availability
  3. Billed per request and by resource consumption

# AWS Lambda

- ▶ AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You can trigger Lambda from over 200 AWS services and software as a service (SaaS) applications, and only pay for what you use.

# How It works

- ▶ AWS Lambda is the basic building block of the serverless platform provided by AWS. The first step in the process is to run your code on Lambda instead of on your own server.

To execute your code with AWS Lambda, follow these steps:

- 1 Write the code.
- 2 Upload your code and its dependencies (such as libraries or modules).
- 3 Create a function determining the runtime environment and configuration.
- 4 Invoke the function to execute your code in the cloud.

# Lambda Function handler

- ▶ The Lambda function *handler* is the method in your function code that processes events. When your function is invoked, Lambda runs the handler method. Your function runs until the handler returns a response, exits, or times out.

## Example TypeScript handler

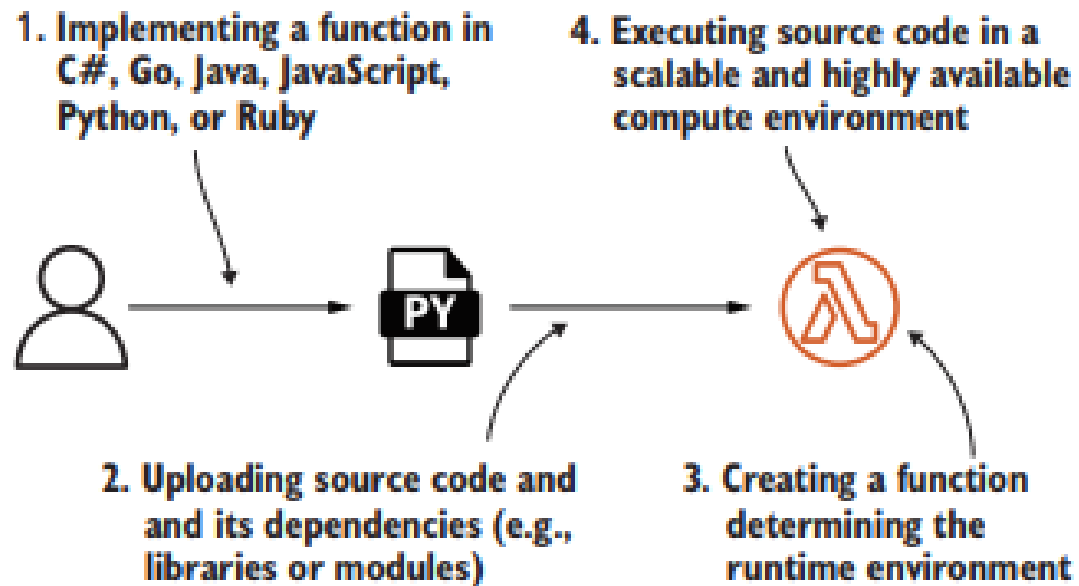
```
import { Handler } from 'aws-lambda';

export const handler: Handler = async (event, context) => {
  console.log('EVENT: \n' + JSON.stringify(event, null, 2));
  return context.logStreamName;
};
```

You can use the following general syntax when creating a function handler in Python:

```
def handler_name(event, context):
    ...
    return some_value
```

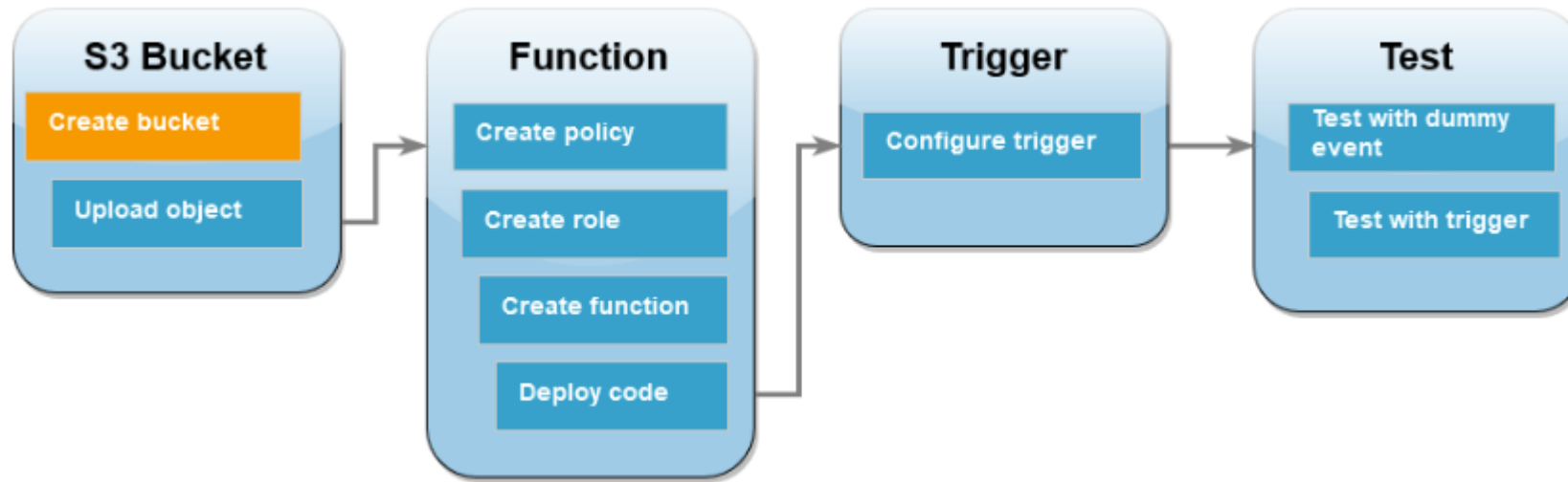
# HelloWorld Example



# AWS Lambda vs AWS EC2

	AWS Lambda	Amazon EC2
Granularity of virtualization	Small piece of code (a function).	An entire operating system.
Scalability	Scales automatically. A throttling limit prevents you from creating unwanted charges accidentally and can be increased by AWS support if needed.	As you will learn in chapter 17, using an Auto Scaling group allows you to scale the number of EC2 instances serving requests automatically, but configuring and monitoring the scaling activities is your responsibility.
High availability	Fault tolerant by default. The computing infrastructure spans multiple machines and data centers.	Virtual machines are not highly available by default. Nevertheless, as you will learn in chapter 13, it is possible to set up a highly available infrastructure based on EC2 instances as well.
Maintenance effort	Almost zero. You need only to configure your function.	You are responsible for maintaining all layers between your virtual machine's operating system and your application's runtime environment.
Deployment effort	Almost zero due to a well-defined API	Rolling out your application to a fleet of virtual machines is a challenge that requires tools and know-how.
Pricing model	Pay per request as well as execution time and allocated memory	Pay for operating hours of the virtual machines, billed per second

# S3 Trigger Example





# What is CloudWatch

- ▶ Amazon CloudWatch is an observability and monitoring service offered by AWS. It allows you to gather and track metrics, collect and monitor log files, set alarms, and respond to changes in your CloudWatch in AWS resources.
- ▶ AWS CloudWatch is an essential service for managing and monitoring your AWS resources efficiently. With proper CloudWatch implementation, you can improve the reliability, performance, and cost-effectiveness of your infrastructure and applications.

# CloudWatch



# CloudWatch



Amazon  
CloudWatch

## **Basic monitoring**

Is free

Polls every 5 minutes

10 metrics

5GB of data ingestion

5GB of data storage

## **Detailed monitoring**

Is chargeable

Charged per instance per month

Polls every minute

# Use Case 1

Can create events based on CloudWatch monitoring, for example trigger Lambda function.



## Use Case 2

Install agents on EC2 instances to send monitoring data about the instance to CloudWatch.



EC2 Instances



Amazon  
CloudWatch

HTTP response codes



Exceptions in  
application logs

# Use Case 3

Set alarms to warn based on resources usage, for example CPU utilization is too high.

