AWS LAMBDA

By The Tech Stuff

Content Table

01)
Introduction

Introduction

A serverless compute service that allows you to run code without provisioning or managing servers. With Lambda, you can execute code in response to events such as changes to data in an Amazon S3 bucket, updates to a DynamoDB table, or custom events from other AWS services.

02)
Key Features

Key Features

- Serverless
- Event Driven
- Scalable
- AWS Integrated
- Flexible

03)
Key Concepts

Key Concepts

Functions

actual code that performs a task. It is written in one of the supported languages and can be triggered by various events.

Runtime

The runtime is responsible for running your function's code.

Trigger

a resource or event source that invokes your Lambda function. Common triggers include S3 events, DynamoDB streams, and API Gateway.

Key Concepts

Concurrency

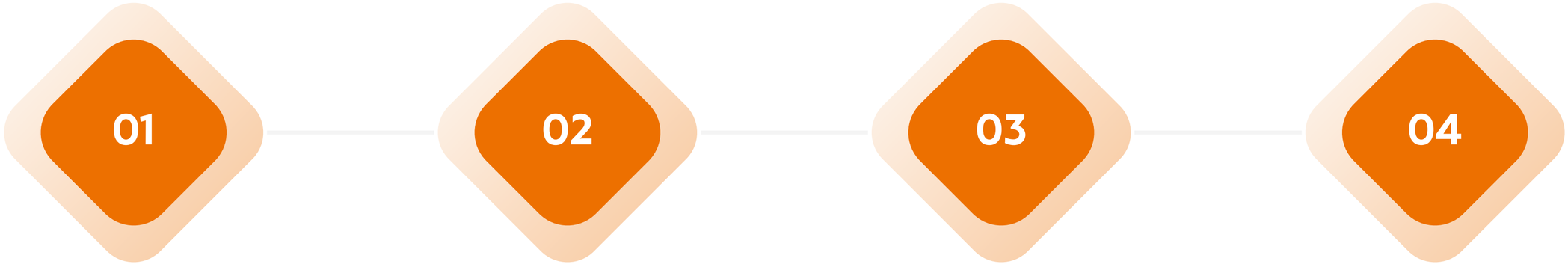
is the number of instances of your function that are running simultaneously. Lambda automatically scales your function's concurrency in response to incoming events.

Execution Role

Lambda functions require an IAM role that grants them permission to access AWS services and resources.

04)
How it Works

How it Works



Upload Code

Set Triggers

Execution

Automatic Scaling

05) Supported Languages

Supported Languages

Node.js

Python

Ruby

Java

.Net

06) Execution Model

Execution Model

The AWS Lambda execution model describes how Lambda functions are executed, managed, and scaled by the AWS infrastructure. Understanding this model is crucial for optimizing performance, managing resources, and ensuring efficient operations.

6.1

Execution Lifecycle

- **Event Source Trigger:** The lifecycle begins when an event source triggers the Lambda function. This can be an AWS service like S3, DynamoDB, or an API Gateway, or a custom event from another application or service.

6.1

Execution Lifecycle

- **Invocation:** Once triggered, AWS Lambda invokes the function by creating an instance of the function handler. There are two types of invocation:
    - **Synchronous Invocation:** The caller waits for the function to process the event and return a response. Examples include API Gateway and ALB.
    - **Asynchronous Invocation:** The function processes the event in the background, and the caller does not wait for the function to complete. Examples include S3 and SNS.

6.1

Execution Lifecycle

- **Container Initialization:** Lambda runs the function code inside a container. The container is created, and the code is loaded. This phase, known as a cold start, can introduce latency as it involves initializing the runtime environment, setting up dependencies, and loading the function code.

6.1

Execution Lifecycle

- **Handler Execution:** The function handler processes the incoming event. The handler is the main entry point of the function code and contains the logic to process the event and generate a response.

6.1

Execution Lifecycle

- **Response or Callback:** Once the handler completes execution, it returns a response (for synchronous invocations) or signals completion via a callback or promise (for asynchronous invocations).

6.2

Cold Starts vs. Warm Starts

- **Cold Start:** Occurs when a new container is created to handle an event. Cold starts incur additional latency due to the need to initialize the runtime environment and load the function code.

6.2

Cold Starts vs. Warm Starts

- **Warm Start:** Occurs when an existing container is reused. Warm starts are faster because they skip the initialization phase, reducing the latency significantly.

07)
Pricing

Pricing

01) Compute Time

02) Free Tier

08)
Use Cases

Use Cases

AWS Lambda is versatile and can be used in various scenarios:

- Web Application
- Data Processing
- IoT Backends
- Automation
- Integration

09) Monitoring and Logging

Monitoring & Logging

AWS Lambda integrates with AWS CloudWatch for monitoring and logging:

01) CloudWatch Logs: Captures logs generated by your function.

02) CloudWatch Metrics: Provides metrics such as invocation count, duration, and error rates.

03) CloudWatch Alarms: Set up alarms to notify you of issues such as high error rates or long execution times.

Thank You!

Email

mayamnaizel2013@gmail.com

Social Media

@the_techstuff

YouTube

The Tech Stuff