AWS Lambda

❖ AWS LAMBDA:

- Lambda is a compute service that lets you **run code without provisioning or managing servers.**
- You pay only for the compute time you consume - there is no charge when your code is not running.
- It runs your code on **high-availability** compute infrastructure and performs all the administration of the compute resources.

LAMBDA BENEFITS:

- Easy Pricing
- Integrated with the whole AWS Stack
- Integrated with many programming languages
- Easy monitoring through AWS CloudWatch
- Easy to get more resources per functions (up to 3GB of RAM!)
- Increasing RAM will also improve CPU and network!

WHEN TO USE LAMBDA:

- Lambda is an ideal compute service for application scenarios that need to scale up rapidly, and scale down to zero when not in demand.
- You can use Lambda for:

  FILE PROCESSING: Use Amazon Simple Storage Service (Amazon S3) to trigger Lambda data processing in real time after an upload.

  STREAM PROCESSING: Use Lambda and Amazon Kinesis to process real-time streaming data for application activity tracking, transaction order processing, clickstream analysis, data cleansing, log filtering, indexing, social media analysis, Internet of Things device data telemetry & metering.

  WEB APPLICATIONS: Combine Lambda with other AWS services to build powerful web applications that automatically scale up and down and run in a highly available configuration across multiple data centers.

  IOT BACKENDS: Build serverless backends using Lambda to handle web, mobile, IoT, and third-party API requests.

  MOBILE BACKENDS: Build backends using Lambda and Amazon API Gateway to authenticate and process API requests. Use AWS Amplify to easily integrate with your iOS, Android, Web, and React Native frontends.

LANGUAGE SUPPORT:

- AWS Lambda natively supports different languages, and provides a Runtime API which allows you to use any additional programming languages to author your functions.
- Supported languages are:
  - Node.js (JavaScript)
  - Python
  - Java (Java 8 compatible)
  - C# (.NET Core)
  - Go
  - C# / Powershell
  - Ruby Code

LAMBDA FUNCTION:

- The code you run on AWS Lambda is uploaded as a **"Lambda function".**
- Each function has associated configuration information, such as its name, description, entry point, and resource requirements.
- The code must be written in a **"stateless"** style **i.e.,** it should assume there is no affinity to the underlying compute infrastructure.

NOTE: Lambda automatically creates default code for the function.

LAMBDA CONFIGURATION:

- After you create a function, you can configure additional capabilities for the function, such as triggers, network access, and file system access.
- You can also adjust resources associated with the function, such as memory and concurrency. These configurations apply to functions defined as .zip file archives and to functions defined as container images.
- Timeout: default 3 seconds, max of 300s (Note: new limit 15 minutes)
- Environment variables
- Virtual private Cloud (VPC)
- Monitoring and Operational Tools
- Allocated memory (128M to 3G)
- Ability to deploy within a VPC + assign security groups
- IAM execution role must be attached to the Lambda function

➢ LAMBDA LIMITS:

EXECUTION:

- Memory allocation: 128 MB – 3008 MB (64 MB increments)
- Maximum execution time: 300 seconds (5 minutes), now 15 minutes but 5 for exam
- Disk capacity in the “function container” (in /tmp): 512 MB
- Concurrency limits: 1000

DEPLOYMENT:

- Lambda function deployment size (compressed .zip): 50 MB
- Size of uncompressed deployment (code + dependencies): 250 MB
- Can use the /tmp directory to load other files at startup
- Size of environment variables: 4 KB