Lambda

# Index

Table of Contents

[Index 2](#_Toc30985440)

[Lambda 3](#_Toc30985441)

[Definition 3](#_Toc30985442)

[Lambda Configurations 3](#_Toc30985443)

[Version Control 4](#_Toc30985444)

[Service Limits 5](#_Toc30985445)

[Cost Model 5](#_Toc30985446)

[Miscellaneous Features 5](#_Toc30985447)

# Lambda

## Definition

* It is a serverless compute service where one can upload their code and create a Lambda function.
* The Lambda function takes care of provisioning and managing the servers that the code runs in; hence the end users do not need to worry about operating systems, patching, scaling, etc.
* It can be used in the following ways:
  + As an event-driver compute service where Lambda runs the code in response to events, such as change in data in Amazon S3 or DynamoDB.
  + As a compute service to run code in response to HTTP requests using Amazon API Gateway or API calls made using AWS SDK.
* Languages supported include Node.js, Java, Python, C#, Go, Ruby and PowerShell.

## Lambda Configurations

* The Lambda runtime
  + Node.js, Java, Python, Go, .NET core, etc.
* The Lambda handler
  + Each function has a handler file and function.
  + The format of the Lambda handler value is filename.functionname.
    - For example, a Python Lambda function might have a file called *‘uptime\_check.py’* with a function called ‘scan’. The handler would be *‘uptime\_check.scan’*.
* Permissions
  + IAM Roles grant the function permission to take action on other AWS resources.
  + Resource-based access control policies allow other services and external accounts to invoke or take other actions on a Lambda function.
* Maximum execution duration
  + Lambda functions can be configured to run up to 15 minutes.
  + The default time-out value is 3 seconds.
  + This feature is useful to limit run-times in case of erroneous code.
* Environment variables
  + Environment variables can be passed to Lambda functions to make the code more dynamic.
  + Lambda encrypts them using the AWS Key Management Service.
  + When the Lambda function is invoked, those values are decrypted and made available to the Lambda code.
* Memory
  + The memory, and by extension CPU, can be configured from 128 MB (base) to up to 3008 MB with 64 MB increments.
* The Lambda function can be deployed within a VPC subnet if chosen to. Appropriate security groups can thus be applied.
* Concurrency
* The number of invocations for the same functional is determined by the concurrency parameter.
* Concurrency can be reserved from anywhere from 0 to 1000. (Soft limit).
  + The maximum unreserved concurrency limit is 1000 invocations.
  + If the number of invocations exceeds the concurrency threshold, then:
    - For synchronous invocations, ThrottleError – 429 is thrown.
    - For asynchronous invocations, post up to two retires, all unprocessed events are sent to a configured Dead Letter Queue (DLQ).
      * The DLQ can be a SNS topic or SQS queue.
      * Relevant IAM permissions have to be attached to the Lambda function to use DLQ.
* For enhanced debugging, one can enable active tracing to integrate the Lambda function with AWS X-Ray.
  + This also adds relevant IAM permissions for using X-Ray to the function role.
* Tags can be associated with any Lambda function.
  + Useful for billing.

## Version Control

* Versions
  + Lambda versions are distinct versions of the functions stored inside of AWS, each with their unique ARNs.
  + Versions are either:
    - The ‘*$LATEST*’ version.
    - A numbered version, like 1 or 2.
      * Numbered versions begin with 1 and are then auto-incremented.
      * Numbered versions are immutable i.e. they cannot be modified.
* Aliases
  + Lambda aliases are like pointers to a specific Lambda version.
  + Using aliases, one can invoke a function without having to know the exact version of the function being referenced.
  + Aliases have static ARNs but can point to any version of the same function.
  + Aliases can be used to split traffic between Lambda versions.
* Benefits of version and aliases:
  + Easier development workflow and management of stages.
  + Avoid having to reconfigure event sources (they can point to a function alias)
  + Rolling back to earlier version becomes easy as updating the alias.
  + Using alias to split traffic between versions can help test new versions in production.

## Service Limits

* A Lambda function can run for up to 15 minutes.
* Memory allocation is in 64 MB increments, starting from 128 MB to 3008 MB.
* One can have disk capacity in the function container (in ‘*/tmp*’ folder) of up to 512 MB.
* The concurrency soft limit for a Lambda function is 1000.
* The deployment size for the Lambda function itself must be:
  + Less than 50 MB compressed in .zip.
  + Less than 250 MB uncompressed (Code + Dependencies).
  + One can load other files at startup in the *‘/tmp’* directory.
* The size of the environment variables must not exceed 4 KB.

## Cost Model

* Billed by the number of requests (per million requests).
* Duration of requests i.e. from the time the code begins to the time it returns or otherwise terminated, rounded to the nearest 100ms.
* Tags can be associated to Lambda functions to help with billing.

## Miscellaneous Features

* Lambda scales out (not up) automatically i.e. multiple invocations trigger multiple Lambda functions.
* Lambda functions are independent, one event equals one function, unless the original function triggers other Lambda functions.
* Lambda functions are stateless i.e. they do not store any state information.
  + One would need something like DynamoDB to make Lambda stateful.
* As architectures can get extremely complicated with Lambda, AWS X-ray can use used to debug the process.
* Some of AWS Lambda best practices include:
  + Perform heavy-duty work outside the function handler.
    - This includes connecting to a database, initializing AWS SDK, pulling dependencies, etc.
  + Use environment variables for:
    - Database connection strings, S3 buckets, etc.
    - Passwords and other sensitive values should always be used as environment variables encrypted with KMS.
  + Minimize deployment package to bare minimum.
  + Break down the function if need be.
  + One should not put the Lambda function in a VPC unless required.
    - An ENI needs to be attached to the Lambda function and an IP from the subnet CIDR block needs to be assigned to it, increasing the startup delay.
    - Each invocation needs a separate ENI. One might run out of IP addresses in the subnet.