AWS Lambda – Function as a Service

Serverless Compute

Content Prepared By: Chandra Lingam, Cotton Cola Designs LLC

amazon web services

Serverless Compute – What is it?

Application still runs on “servers”

All server management is done by AWS

Servers to Serverless

- EC2 – You manage the Server
- RDS – You pick server configuration and AWS manages
- Lambda, SQS, S3 .. – Serverless. Fully managed by AWS

Benefits of Lambda

No Servers to manage – just write your code and upload

Continuous Scaling – Lambda automatically scales by running your function in parallel

No charge when code is not running - don’t pay for idle infrastructure

Run code in response to a trigger

Lambda Language Support

Node.js

Python

Java

C# (.NET Core)

AWS Manages Servers and Execution Environment

Cannot customize OS or Execution environment

Lambda Invocation – Usage Scenarios

Push: S3 to Lambda

Polling: Kinesis Streams to Lambda

Invoke: App to Lambda

Event Source Mapping – Where is it maintained?

Supported Event Sources & Sample Event Data

IAM Permissions

- Lambda Function needs permission to access resources
  - Granted through Execution Role – attached to the function
  - Lambda Assumes Roles when executing the function
- Who can invoke the Lambda Function?
  - Resource Based Policies at Function Level
  - IAM Roles, User, Group Policies

Demo – Hello World

Example - 1

- Returns Hard Corded String

Example – 2

- Returns back the event data that was sent

- Write, Configure, Test and check CloudWatch Logs
- Invocation Example from Console and CLI
- Async and Sync Invocation

Demo – API Gateway Example

- Expose Lambda function as RESTful API
- Configure API Gateway

Demo – S3 to Lambda

- Push Example
- Trigger Lambda Code on object creation
- Lambda reads the object and prints metadata

Demo – SNS to Lambda

- Subscribe Lambda to SNS Topic
- Process Events
- Push example

Demo – Lambda to SQS

- Polling Example
- Scheduled invocation of Lambda function using CloudWatch Event Scheduler
- Polls configured SQS queue
- Fan out – Parallelize processing of messages
- Use Environment Variable to store the Queue URL

Demo – Version and Alias

- Create Multiple Versions of Function
- Latest version invocation
- Specific version invocation
- Alias - Hide version from Event Sources
- Update Alias to point to new versions

AWS Lambda - Containers

- Packages your code and launches a container
- Takes time to setup container and bootstrap – possibly for every lambda invocation
- AWS Lambda tries to reuse containers for subsequent invocations
- Under Reuse:
    - Any declarations remains initialized – for example: database connections
    - Scratch space /tmp is available – can use it to cache
    - Background process or callback that were not completed may resume!

Function Configuration

- Compute Resources Needed – Specify total memory needed. Lambda allocates CPU capacity in proportion
  - Increments of 64 MB
- Maximum Execution Timeout - Configurable up to max 300 seconds (5 minutes)
- IAM Execution Role – Grants necessary permissions for function to access other resources
- Handler Name – Main method where Lambda Execution begins

Invocation Types

- Synchronous (RequestResponse)
- Asynchronous (Event)

You can specify invocation type through code or CLI

When other AWS Services invoke Lambda, invocation type is predetermined for each of these services

Invocation Types – From AWS Services

Some examples…

S3 – Asynchronous invocation

Cognito – Synchronous invocation

Kinesis/DynamoDB Streams – Lambda polls for data and invokes function Synchronously

Concurrent Executions – Non Stream based

- Each published event is a unit of work
- Number of events dictate concurrency
- Concurrency = Number of events per second * function duration

  Example

  Function duration = 3 seconds

  Events per second = 10

  Concurrent Executions = 10 * 3 = 30
- 1000 concurrent execution limit across all your functions

Concurrent Executions - Stream based

Number of shards per stream is the unit of concurrency.

Example: 100 shards would trigger 100 lambda functions running concurrently

One lambda function processes events on a shard in the order that they arrive

Kinesis Streams and DynamoDB Streams are natively supported by Lambda

Throttling & Error Handling

- Default limit = 1,000 concurrent executions (soft limit)
- Any request rate above the limit is throttled
- Stream based
    - Keeps trying until data expires (up to 7 days for Kinesis)
    - Throttled events in a Shard are blocking – strictly processed in-order
    - Non-throttled shard can continue processing

Throttling & Errors – Non Stream Based

- Synchronous (RequestResponse Type)
  - Invoking application receives 429 error
  - Application is responsible for retries
  - AWS Event Sources may have their own retries
- Asynchronous (Event Type)
  - Events are queued
  - AWS Lambda automatically retries for up to six hours with delays

Authoring Functions

Tools Available for Authoring

General Structure of a Lambda function

Deployment Package

Version and Alias

Manage Version and Alias

Environment Variables

Pass settings to Lambda Function using Environment Variables

VPC Support

- Lambda by default runs inside system-managed VPC
- Lambda cannot access resources in a customer VPC
- Function can be configured to run inside customer VPC
  - VPC
  - Subnet(s)
  - Security Group(s) to use
  - Lambda attaches Elastic Network Interface (ENI) based on above configuration to connect to resources
  - ENI can result in additional startup delay
- Ensure VPC has sufficient addresses in subnet and ENI

VPC Support

- Lambda on system-managed VPC can access internet directly
- Lambda on Customer VPC for outbound internet access needs NAT device
    - Attached ENI has only private address
    - Configure Lambda to run in private subnets
    - Ensure NAT instance or Gateway is configured for VPC
    - Required for accessing other AWS Services

Monitoring

Monitoring Lambda Functions

Metrics

Dead Letter Queues

- Async – failed invocation is retried twice and then discarded
- Configure failed invocations to be sent to SQS queue or SNS topic

X-RAY

Trace Lambda Calls with AWS X-Ray

AWS Lambda Limits

Table: Limits

Pricing

Table: Pricing

Recommended Videos

AWS re:Invent 2016: Serverless Architectural Patterns and Best Practices (ARC402)