Session1 : Basic Introduction,Lambda Function,Trigger,Downstream Resources and Runtime

• AWS Lambda is a compute service that lets us run code without provisioning or managing servers
• With AWS Lambda,we can run code for virtually any type of application or backend-service all with zero administration
• AWS Lambda manages all the administration,It manages :
☐ Provisioning and capacity of the compute fleet that offers a balance of memory,CPU,Network and other resources
☐ Server and O.S maintenance
☐ High availability and automatic scaling
☐ Monitoring fleet health
☐ Applying security patches
☐ Deploying security patches
☐ Deploying our Code
☐ Monitoring and logging our Lambda functions
☐ AWS Lambda runs our code on a high availability compute infrastructure

• AWS Lambda runs our code on a high-availability compute infrastructure
• AWS Lambda executes our code only when needed and scales automatically,from a few requests per day to thousands per second
• We pay only for the compute time we consume.No charges when our code is not running
• All we need to do is supply our code in the form of one or moe lambda functions to AWS Lambda,in one of the languages that AWS Lambda supported(Currently Node JS,JAVA,Powershell,C#,Ruby,Python and GO) and the service can run the code on our behalf

• Typically the lifecycle for an AWS Lambda based application includes authoring code,deploying code to AWS Lambda and then monitoring and troubleshooting
• This is in exchange for flexibility means we cannot log into Compute instances or customize the O.S or language runtime
• If we do want to manage our own compute,we can use EC2 or Elastic Beanstalk

How Lambda Works :

• First we upload our code to lambda in one or more lambda functions
• AWS Lambda will then execute the code in our behalf
• After the code is invoked lambda automatically take care of provisioning and managing the required servers

AWS Lambda vs AWS EC2

AWS Lambda :

• AWS lambda is a Platform as a Service
• It supports only limited languages like NodeJs,Java,Python,Ruby,GO,C#,Powershell
• We write our code and push the code into AWS Lambda
• We cannot log into compute instances,choose customized O.S or language platform

AWS EC2 :

• It is a IAAS
• No environment restrictions,we can run any code or language

• For the first time in EC2,we have to choose the O.S and install all the software required and then push our code in EC2
• We can select variety of O.S,instance types,Network & security patches,RAM & CPU etc

Important Terms used in AWS Lambda :

1.Functions : A function is a resource that we can invoke to run our code in AWS Lambda.A function has code that processes events and a runtime that passes request and responses between lambda and the function code

2.Runtime : Lambda Runtime allows functions in different languages to run in the same base execution environment.The runtime sits in between the lambda server and our function code,relaying invocation events,context information and responses between the two

3.Event : It is a JSON formtted document that contain data for a function to process

4.Event Source/Trigger : An AWS service such as Amazon SNS,or a custom service that triggers our function and executes its logic

5.Downstream Resources : An AWS service,such as DynamoDB tables or S3 buckets,that our lambda function calls once it is triggered

6.Concurrency : No.of request that our function is serving in any given time..Default is 1,000 times

Session 2 : AWS Lambda Invocation Types

When Lambda Triggers :

• We can use AWS Lambda to run our code in response to :
☐ Events such as changes to data in amazon s3 ucket or an amazon dynamodb table
☐ To run our code in response to HTTP request using amazon AOI gateway
☐ With these capabilities,we can use lambda to easily build data processing triggers for AWS services like amazon s3 and amazon dynamodb,process streaming data stored in kinesis or create our own backend that opeartes at AWS scale,performance and security

Example of S3 :

• The user create an object in a bucket
• Amazon S3 detects the object created event
• Amazon S3 invokes our lambda functions using the permission provided by the execution role
• Amazon s3 knows which lambda function to invoke based on the event source mapping that is stored in the bucket notification configuration

AWS Lambda Function Configuration :

• A Lambda function consists of code and any associated dependencies
• In Addition,a lambda function also has configuration inforamtion associated with it
• Initially,we specify the configuration information when we create a lambda function
• Lambda provides an API for us to update some of the configuration data

Lambda function configuration information includes the following key elements :

• Compute resource that we need.We only specify the amount of memory we want to allocate from our lambda function
• Aws lambda allocates,CPU power proportional to the memory by using the same ratio as a general purpose amazon ec2 instance types,sucha s an M3 type
• We can update the configuration and request additional memory in 64mb increments(64 multiples) from 128 to 3008 MB
• Functions larget than 1536MB are allocated multiple CPU threads

Maximum Execution Timeout :

• We pay for the AWS resources that are used to run our lambda function
• To prevent our lambda function from running indefinitely,we specify a timeout
• When the specified timeout is reached,AWS lambda terminates our lambda function
• Default is 3 seconds and maximum is 900 Seconds (15 Min)

IAM Role : This is the role that AWS lambda assume when it executes the lambda function on our behalf

AWS Lambda Function - Service it can Access :

• Aws Services or non-aws services
• Aws services running in AWS VPC(eg-redshift,elasticcache,rds instance)
• Non-aws services running on ec2 instance in an aws vpc
• Aws lambda run our function code securely within a vpc by deafult
☐ However,to enable our lambda function to access resources inside our private vpc,we must provide additional vpc specific configuration information that includes vpc subnet ID and security group ID's

Different ways to invoke Lambda Function : Synchronous invoke(push),Asynchronous invoke(event),Poll-based invoke(pull based)

Synchronous Invocation :

• Synchronous invoke are the most straight forward way to invoke our lambda function
• In this model our function executes immediately when we perform the lambda invoke API call
• Invokation flag specifies a value of ‘request response’
• We want for the function to process the event and return a response

Here is the list of services that invoke lambda function synchronously

☐ Elastic lad balancer
☐ Cloudfront
☐ Amazon cognito
☐ Api gateway
☐ Amazon Lex
☐ Kinesis data firehose

Asynchronous invocation :

• In this incokation lambda places the event in a queue and returns a success response without additional information
• Lambda queues the event for processing and returns a response immediately
• we can configure lambda to send an invocation record to another service like SQS,SNS,Lambda and eventbridge

Here is list of functions that invoke lambda function asynchronously
☐ Amazon S3
☐ SNS
☐ SES
☐ Cloudformation
☐ Cloudwatch logs
☐ Cloudwatch events
☐ AWS codecommit
☐ Aws config

Poll-based Invocation :

• This invocation model is designed to allow us to integrate with aws stream and queue based service with no code or server management lambda will poll the following service on our behalf,retrive records and invoke our function
• The following are supported services
☐ Amazon Kinesis
☐ SQS
☐ DynamaDB streams

Session 3 : Setup S3 Trigger with Lambda(Lab)