**Lambda**

* + Lambda lets you run code as functions without provisioning or managing servers
  + Lambda-based applications(also referred to as serverless applications) are composed of functions triggered by events
  + With serverless computing, your application still runs on servers, but all the server management is done by AWS
  + Lambda functions:
    - Consist of code and any associated dependencies
    - Configuration information is associated with the function
    - You specify the configuration information when you create the function
    - API provided for updating configuration data
  + You specify the amount of memory you need allocated to your Lambda functions
  + AWS Lambda allocates CPU power proportional to the memory you specify using the same ratio as general purpose EC2 type
  + To enable your lambda function to access resources inside your VPC you must provide additional VPC-speific configuration information that includes VPC subnet IDs and security group IDs
    - Lambda uses this information to set up ENI's that enable your function
  + Compute resources:
    - You can request additional memory in 64MB increments from 128MB to 3008MB
    - Functions larger than 1536MB are allocated multiple CPU threads, and multi-threaded or multi-process code is needed to take advantage
  + There is a maximum execution timeout
    - Max is 15 minutes(900 seconds)
    - default is 3 seconds
    - You pay for the time it runs
    - lambda terminates the function at timeout
  + Code is invoked using API calls made using AWS SDKs
  + Lambda assumes an IAM role when it executes the function
  + Lambda components
    - Lambda function comprised of custom code and dependent libraries
    - Event sources such as SNS or custom service that triggers your function and executes its logic
    - Downstream resources such as DynamoDB or S3 buckets that your Lambda function calls once its triggered
    - Log streams are custom logging statements that allow you to analyze the execution flow and performance of your Lambda function
  + Lambda is an event-driven compute service where AWS Lambda runs code in response to events such as changes to data in an S3 bucket or DynamoDB table
  + An event source is an AWS service or developer-created application that produces events that trigger an AWS Lambda function to run
  + Event sources are mapped to Lambda functions
  + Lambda can run code in response to HTTP requests using API Gateway or API calls made using AWS SKD's
  + Lambda supports code written in
    - Node.js(JavaScript)
    - Python
    - Java
    - C#
    - Ruby
    - Go
    - Powershell
  + Lambda stores code in S3 and encrypts it at rest
  + Lambda functions are serverless and indepent. 1 event = 1 function
    - Functions can trigger other functions
  + Lambda works globally
  + Lambda functions provide access only to a single VPC. If multiple subnets are specified they must all be in the same VPC
  + Lambda functions configured to access resources in a particular VPC will not have access to the Internet as default. If you need access to external endpoints, you will need to create a NAT in your VPC to forward this traffic and configure your SG to allow this outbound traffic
  + Each Lambda function has a unique Amazon Resource Name(ARN) which cannot be changed after publishing
  + 1000 max concurrent executions per account
  + Lambda automatically monitors Lambda functions and reports metrics through CW
  + Lambda tracks number of requests, latency per request and the number of requests resulting in an error
  + You can view request rates and error rates using Lambda console, CloudWatch console
  + X-Ray is an AWS service that can be used to detect, analyse and optimse performance issues with Lambda applications
  + X-ray collects metadata from Lambda service and any upstream and downstream services that make up your application
  + Charges
    - Based on:
      * number of requests
      * First 1 million are free
      * After that $0.20 per 1 million
      * Duration: calculated form the time your code begins execution until it returns or terminates. Depends on the amount of memory allocated to a function
  + Lambda@Edge
    - Allows you to run code across AWS locations globally without provisioning or managing servers, responding to end users at the lowest network latency
    - Allows you to run Node.js and Python Lambda functions to customize content that CloudFront delivers, executing functions in AWS locations closer to the viewer
    - Functions run in response to CloudFront events. Lambda functions can change CF requests and responses at the following points
      * After CF receives a request from a viewer(viewer request)
      * Before CF forwards the request to the origin(origin request)
      * After CF receives the response from the origin(origin response)
      * Before CF forwards the response to the viewere(viewer response)