This study guide is based on the video lesson available on TrainerTests.com

Study Guide: AWS Lambda Invocation and Event-Driven Architecture

I. Introduction

AWS Lambda, a cornerstone of serverless computing, introduces a paradigm shift in application development by operating in an event-driven manner. In this document, we will delve into the intricacies of how AWS Lambda functions are invoked, emphasizing their event-driven nature and highlighting the array of AWS services that can trigger Lambda functions.

II. Invocation of AWS Lambda Functions

A. Event-Driven Execution

AWS Lambda functions are executed in response to specific events or triggers. These events can originate from various AWS services or custom sources, creating a highly flexible and scalable architecture. The event-driven nature of Lambda ensures that compute resources are allocated only when needed, optimizing efficiency and cost-effectiveness.

B. Pay-Per-Use Model

One of the distinctive features of AWS Lambda is its pay-per-use pricing model. You are billed only for the compute time consumed during the execution of your functions. This aligns with the serverless philosophy, where resources are provisioned dynamically based on demand, eliminating the need for maintaining idle infrastructure.

III. Event Sources for AWS Lambda

AWS Lambda can seamlessly integrate with a multitude of AWS services, serving as an event source for Lambda functions. The following AWS services can trigger Lambda functions based on specific events or changes:

A. API Gateway

API Gateway can invoke Lambda functions in response to HTTP requests, making it an ideal choice for building serverless APIs. Lambda functions can be connected to specific API Gateway endpoints, enabling the execution of business logic upon API invocation.

B. Kinesis

AWS Lambda can process and respond to streaming data from Amazon Kinesis streams. By associating a Lambda function with a Kinesis stream, real-time data processing and analytics become achievable without the need for manual intervention.

C. DynamoDB

Changes in a DynamoDB table, such as inserts, updates, or deletes, can trigger Lambda functions. This enables developers to build event-driven workflows that respond to changes in their NoSQL database.

D. S3

AWS Lambda can be triggered when objects are created, modified, or deleted in an Amazon S3 bucket. This integration allows for serverless processing of data stored in S3, opening up possibilities for tasks like image processing, log analysis, and more.

E. SNS (Simple Notification Service)

SNS topics can serve as triggers for Lambda functions. When an event occurs on an SNS topic, such as the publication of a message, associated Lambda functions are invoked, enabling event-driven communication between services.

F. SQS (Simple Queue Service)

Lambda functions can be connected to SQS queues, responding to messages as they arrive. This integration facilitates the implementation of asynchronous and decoupled architectures, where messages in the queue trigger specific Lambda functions.

IV. Conclusion

AWS Lambda's event-driven architecture and diverse integration capabilities empower developers to build scalable, efficient, and cost-effective serverless applications. By leveraging Lambda functions in response to specific events from various AWS services, organizations can unlock new levels of agility and responsiveness in their cloud-based applications. Understanding the event-driven nature of Lambda is crucial for architects and developers seeking to harness the full potential of serverless computing on the AWS platform.