



TrainerTests.com

This study guide demonstrates the lesson from *Introduction to Serverless Using Lambda – Eliminate Virtual Servers!*

My full AWS Architect Associate course can be found here:

<https://www.udemy.com/course/ultimateaws/?referralCode=7ED214B795C444141361>

Serverless and Lambda Study Guide

Introduction

The landscape of application development has been revolutionized by the advent of serverless computing, a paradigm shift from traditional server-based approaches. This lesson introduces the fundamental concepts of serverless applications, focusing on AWS Lambda, Amazon Web Services' flagship serverless compute service. By contrasting Lambda with traditional server services like EC2 (Elastic Compute Cloud), we will explore the key advantages of serverless computing, including cost efficiency, scalability, and ease of management.

Understanding Serverless Computing

Serverless computing is a cloud-computing execution model in which the cloud provider manages the execution environment, freeing developers from the complexities of server management.

- **Abstraction of Servers:** Serverless computing doesn't mean the absence of servers, but rather the abstraction of server management from the developer.
- **Event-Driven Architecture:** Serverless applications are typically event-driven, automatically triggering and executing code in response to specific events.

AWS Lambda: A Deep Dive

AWS Lambda is a prime example of serverless computing. It allows you to run code in response to events without the need to provision or manage servers.

- **Function-as-a-Service (FaaS):** Lambda operates on a FaaS model, where you upload your code, and AWS takes care of everything required to run and scale the code execution.
- **Event-Driven Triggers:** Lambda functions can be triggered by various AWS services, such as S3 (Simple Storage Service) and DynamoDB, or by HTTP requests through API Gateway.

Contrasting Lambda with EC2

To appreciate the benefits of AWS Lambda, it's essential to understand how it differs from traditional services like EC2.

- **EC2 (Traditional Server Model):** In EC2, you provision and manage server instances. You are responsible for server health, scaling, and ensuring the server software is up to date.
- **Lambda (Serverless Model):** With Lambda, there are no servers to manage. You are only responsible for your code. AWS handles the scaling, high availability, and maintenance of the underlying infrastructure.

Advantages of Serverless Computing

Serverless computing, as exemplified by AWS Lambda, offers several key benefits:

Cost Efficiency

- **Pay-Per-Use Model:** You only pay for the compute time you consume with Lambda, as opposed to paying for reserved compute capacity with EC2.
- **Reduced Overhead:** The absence of server management reduces the cost associated with infrastructure maintenance.

Scalability

- **Automatic Scaling:** Lambda automatically scales your application by running code in response to each trigger. This can be a single request or thousands at once.
- **Handling Traffic Spikes:** Serverless applications can handle sudden spikes in traffic more gracefully, as the cloud provider manages the scaling.

Ease of Management

- **No Server Management:** The elimination of server management tasks simplifies deployment and operational management.
- **Rapid Deployment:** You can quickly deploy new versions of applications without worrying about the underlying infrastructure.

Use Cases for AWS Lambda

AWS Lambda is suited for various use cases, such as:

- **Web Applications:** Building serverless backends for web applications.
- **Data Processing:** Real-time file processing, stream processing, and batch processing.
- **Automation:** Automating AWS services and backend tasks.

Conclusion

Serverless computing, particularly through AWS Lambda, represents a significant shift in application development and deployment. By abstracting the complexities of server management, Lambda offers a streamlined, cost-effective, and scalable solution for running code in the cloud. This model allows developers to focus on writing code and creating value, rather than managing and operating servers, making serverless computing an increasingly attractive option in a wide range of applications.

See slides below:

What is Serverless?



Amazon DynamoDB

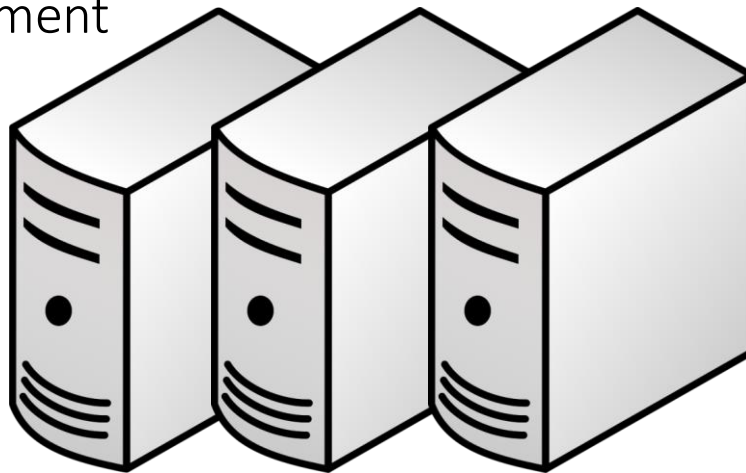


Amazon S3

What is Serverless?



- EC2 Charges
- OS Management
- ELB



What is Serverless?

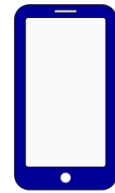


Eliminate Management Requirements



- Operating Systems
- Scalability
- Availability
- Fault Tolerance
- No need to pay for idle resources
- Stateless

Invoking Lambda



AWS Services that Invoke Lambda



API Gateway



Amazon Kinesis



Amazon DynamoDB



Amazon S3



Simple Notification Service
SNS



Simple Queue Service
SQS

For more details see my full AWS Architect Associate course:

<https://www.udemy.com/course/ultimateaws/?referralCode=7ED214B795C444141361>