**Lecture Notes for AWS Developer**

**Module-6**

**AWS SERVERLESS SERVICE**

A serverless architecture is a way to build and run applications and services without having to manage infrastructure. Your application still runs on servers, but all the server management is done by AWS. You no longer have to provision, scale, and maintain servers to run your applications, databases, and storage systems.

Serverless is a way to describe the services, practices, and strategies that enable you to build more agile applications so you can innovate and respond to change faster. With serverless computing, infrastructure management tasks like capacity provisioning and patching are handled by AWS, so you can focus on only writing code that serves your customers.

**Why to use serverless architectures?**

By using a serverless architecture, your developers can focus on their core product instead of worrying about managing and operating servers or runtimes, either in the cloud or on-premises. This reduced overhead lets developers reclaim time and energy that can be spent on developing great products which scale and that are reliable.

**AWS LAMBDA**

AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, or managing runtimes. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code as a ZIP file or container image, and Lambda automatically and precisely allocates compute execution power and runs your code based on the incoming request or event, for any scale of traffic. You can set up your code to automatically trigger from 140 AWS services or call it directly from any web or mobile app. You can write Lambda functions in your favorite language (Node.js, Python, Go, Java, and more) and use both serverless and container tools, such as AWS SAM or Docker CLI, to build, test, and deploy your functions.

**Benefits**

**No servers to manage**

AWS Lambda automatically runs your code without requiring you to provision or manage infrastructure. Just write the code and upload it to Lambda either as a ZIP file or container image.

**Continuous scaling**

AWS Lambda automatically scales your application by running code in response to each event. Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload, from a few requests per day, to hundreds of thousands per second.

**Cost optimized with millisecond metering**

With AWS Lambda, you only pay for the compute time you consume, so you’re never paying for over-provisioned infrastructure. You are charged for every millisecond your code executes and the number of times your code is triggered. With Compute Savings Plan, you can additionally save up to 17%.

**Consistent performance at any scale**

With AWS Lambda, you can optimize your code execution time by choosing the right memory size for your function. You can also keep your functions initialized and hyper-ready to respond within double digit milliseconds by enabling Provisioned Concurrency.

**Lambda@Edge**

Lambda@Edge is a feature of Amazon CloudFront that lets you run code closer to users of your application, which improves performance and reduces latency. With Lambda@Edge, you don't have to provision or manage infrastructure in multiple locations around the world. You pay only for the compute time you consume - there is no charge when your code is not running.

With Lambda@Edge, you can enrich your web applications by making them globally distributed and improving their performance — all with zero server administration. Lambda@Edge runs your code in response to events generated by the Amazon CloudFront content delivery network (CDN). Just upload your code to AWS Lambda, which takes care of everything required to run and scale your code with high availability at an AWS location closest to your end user.

**Benefits**

**Build more responsive applications**

Lambda@Edge runs your code globally at AWS locations close to your users, so you can deliver full-featured, customized content with high performance, and low latency.

**No servers to manage**

You can automatically scale and run your code at AWS locations around the world without requiring you to provision, scale, or manage origin servers at multiple locations, or set up any load balancing or domain name system (DNS) routing services. You can add new functionalities without making any changes to your existing applications running at your origin. Lastly by using Lambda@Edge and Amazon CloudFront, you have less origin infrastructure to manage than with traditional CDNs.

**Customize your content delivery**

With Lambda@Edge, you can customize the content delivered through the Amazon CloudFront CDN, and you can customize your compute resources and execution time, based on your application performance needs.

**How it works**



**AWS CodeDeploy**

AWS CodeDeploy is a fully managed deployment service that automates software deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers. AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications. You can use AWS CodeDeploy to automate software deployments, eliminating the need for error-prone manual operations. The service scales to match your deployment needs.

**Benefits**

**Automated deployments**

AWS CodeDeploy fully automates your software deployments, allowing you to deploy reliably and rapidly. You can consistently deploy your application across your development, test, and production environments whether deploying to Amazon EC2, AWS Fargate, AWS Lambda, or your on-premises servers. The service scales with your infrastructure.

**Minimize downtime**

AWS CodeDeploy helps maximize your application availability during the software deployment process. It introduces changes incrementally and tracks application health according to configurable rules. Software deployments can easily be stopped and rolled back if there are errors.

**Centralized control**

AWS CodeDeploy allows you to easily launch and track the status of your application deployments through the AWS Management Console or the AWS CLI. CodeDeploy gives you a detailed report allowing you to view when and to where each application revision was deployed. You can also create push notifications to receive live updates about your deployments.

**Easy to adopt**

AWS CodeDeploy is platform and language agnostic, works with any application, and provides the same experience whether you’re deploying to Amazon EC2, AWS Fargate, or AWS Lambda. You can easily reuse your existing setup code. CodeDeploy can also integrate with your existing software release process or continuous delivery toolchain (e.g., AWS CodePipeline, GitHub, Jenkins).