|  |  |
| --- | --- |
| **AWS Certified Developer Associate** |  |

**Course Objective:**

Candidates will understand the Development of Application and Design architecture of APP infra in AWS Cloud.

**Duration:**

4 days (32 Hours)

**Delivery Mode:**

Theoretical + Demo + Lab + Exercises + MCQs

**Target Audience:**

The course is designed for people who have basic knowledge of Cloud and Traditional infrastructure.

**Knowledge Prerequisites:**

AWS Cloud basic knowledge with services – Regions/AZ, EC2, ELB, Storage – EBS, S3, Serverless Service Overview.

**Lab Prerequisites:**

Laptop with open Internet connectivity

AWS account for each individual with Admin Access (Can be provided from our end, based on separate commercials)

**Linked Certification:**

<https://aws.amazon.com/certification/certified-developer-associate/>

**Day 1:**

**Architectural Concepts of AWS**

* AWS Fundamentals & Architectural Concepts
* AWS Regions & Services offered
* LAB: AWS Account setup and portal walkthrough
* AWS considerations for HA and DR
* Best practices for deployment/managing applications in different regions
* LAB: Working with AWS

**Different tools to work with Cloud**

* AWS GUI
* Working with CLI and where to use CLIs
* Benefits of CLI
* AWS CDK
* AWS SDK for Developers
* Lab: Setup of .Net SDK with Visual Studio
* LAB: Setup Boto3 for Python SDK
* LAB: AWS GUI, CLI, SDK, CDK
* Cloudwatch – The AWS monitoring giant
* LAB – Working with Cloud Watch

**Overview of AWS Core services**

* AWS EC2 instances and how they are different than traditional environment
* LAB: Working with EC2
* Lab Exercise: Launch EC2 Instance using SDKs (boto3 or .net)

**Overview of AWS Core services**

* AWS Ec2 Instance configurations
* AWS Snapshots, AMI and Encryption of Snapshots
* AWS KMS Service
* LAB: Working with Security Group and NACL

**Day 2:**

**AWS Storage Services**

* AWS Object and File Storage services
* LAB: AWS S3
* AWS S3 Resource Policy
* Lifecycle Policies
* LAB: Working with EFS

**AWS Network Service Overview**

* VPC Overview
* Subnet Overview – Public, private and VPN Only
* Connectivity details

**Application Authentication services**

* IAM - AWS access management capabilities
* Application authentication ways
* Security & Policies
* AWS Single Sign-on
* REST API Authentication
* Signature Version 4

**Day 3:**

**Advance Authentication and Secrets**

* AWS Secret Manager to store credentials.
* Credentials Rotation
* AWS Cognito Service
* AWS Cognito Post Authentication Triggers
* Authentication Calls to AWS Services
* Using Instance roles to assign permissions and Metadata to work with Credentials.
* LAB – IAM services
* LAB: Working with Access Keys and Secret keys and Pros/cons of same
* LAB: Using AWS Roles over AK/SK
* Lab: Sample Application development with AWS SDK and with AWS roles authentication

**Serverless Architecture and Microservices**

* AWS Lambda
* AWS Serverless Application Model (SAM)
* Writing code for serverless
* Migrating application on serverless
* Why and when to use lambda.
* Restrictions and limitations of Lambda
* Lambda Layers for Dependencies
* API Gateway Essentials
* API Gateway Stage Variables
* Gateway deployment and various stages
* LAB: Deploying a sample application using Lambda and API Gateway
* Canary deployment in API gateway
* Authentication and Authorization in API Gateway
* Policy in API Gateway
* AWS Step Functions
* AWS SWF
* Identifying RCA (Root cause) of Application latency

**Day 4:**

**Application Services**

* Elastic Beanstalk
* LAB: Deploying an application on Elastic Beanstalk
* Simple Notification Services (SNS)
* SNS APIs and Error
* AWS AppSync
* LAB: Working with SNS

**AI Services**

* Bedrock
* Key benefits and use cases of Bedrock
* Foundation models in Bedrock
* Developing application using Bedrock

**AWS DevOps Toolkit**

* AWS Code Commit
* AWS code Build
* AWS Code Deploy
* AWS Code Pipeline
* AWS Code star
* X-Ray
* Java APP deployment using AWS DevOps Pipeline
* Integrating testing in Pipelines
* Working with Cloud9 – AWS IDE
* LAB: Deploying a sample application and creating CICD pipeline using Code commit, deploy, build, pipeline and Cloud9

**Containerization**

* Why Containerization
* Docker on AWS
* AWS EKS - Kubernetes on AWS
* AWS Container Registry
* AWS ECS Service
* Fargate Cluster Service
* Lab: Application Deployment on AWS ECS Cluster
* Application deployment on Kubernetes cluster
* LAB: AWS EKS

**Discussion on Production Scenarios**

**Q & A**