



# AWS Cloud & DevOps Engineering

## Professional Course Curriculum

*AWS Certified Solutions Architect – Associate | DevOps | Containers | CI/CD | Python | Cloud-Native*

|                      |                                                                          |
|----------------------|--------------------------------------------------------------------------|
| <b>Duration</b>      | 6.0 Months (17 Modules)                                                  |
| <b>Delivery</b>      | AWS Instructor-Led · Live Labs · Hands-On Projects                       |
| <b>Certification</b> | AWS Certified Solutions Architect – Associate (SAA-C03)                  |
| <b>Prerequisites</b> | Basic IT literacy; no prior cloud, coding, or Python experience required |
| <b>Format</b>        | Weekly live sessions + self-paced modules + capstone project             |
| <b>Languages</b>     | Python 3 · Bash/Shell · YAML · HCL (Terraform)                           |

## Intended Audience

This programme is designed for:

- Entry-level IT professionals seeking Cloud & DevOps engineering skills and AWS certification
- Solution Architects and Design Engineers new to AWS looking to accelerate their certification
- Software developers transitioning into cloud-native and DevOps engineering roles
- IT professionals aiming to build CI/CD pipelines and automate cloud infrastructure at scale
- Anyone targeting roles such as Cloud Engineer, DevOps Engineer, or Solutions Architect

## Course Modules

MOD  
ULE  
01

### Introduction to IT & Cloud Foundations

- On-Premise Infrastructure Overview and Stack IT Business Process
- Stack IT ON-PREM Server Architecture
- Development Methodologies: Agile (Ceremonies, Sprints, Artifacts) and Waterfall
- Software Development Life Cycle (SDLC) — In-Depth
- Server Fundamentals: Components, Types, and Roles
- Cloud Computing Key Terms: IaaS, PaaS, SaaS, Elasticity, Scalability, HA
- Public vs. Private vs. Hybrid Cloud Models

MOD  
ULE  
02**Linux Fundamentals for Cloud Engineers****Command Line & System Administration**

- The Command Line: What it is, how it works, and how to access it
- Basic Navigation: Linux directory system and filesystem hierarchy
- File Characteristics: Files vs directories, hidden files, symbolic links
- Manual Pages: Using man to master Linux commands
- File Manipulation: Create, remove, rename, copy, and move files/directories
- Vi/Vim Text Editor: Modes, editing, saving, and quitting
- Wildcards & Globbing: Referencing multiple files efficiently
- Permissions: chmod, chown, umask, and access control
- Filters: sort, cut, awk, sed, head, tail, uniq
- Grep & Regular Expressions: Pattern matching and data processing
- Piping & Redirection: Chaining commands with |, >, >>, <
- Process Management: ps, top, kill, background jobs, nohup
- Shell Scripting: Variables, loops, conditionals, functions
- Advanced Scripting: Cron jobs, automation, error handling
- SSH & Remote Access: Key-based authentication, scp, sftp

MOD  
ULE  
03**Introduction to AWS & Core Services****AWS Foundations & IAM**

- Why AWS? Cloud market share, certifications, and career value
- AWS Account Setup and Console Navigation
- Identity and Access Management (IAM) 101 — Users, Groups, Roles, Policies
- IAM Step-by-Step: Creating users, assigning permissions, MFA
- Creating a Billing Alarm and Cost Explorer
- AWS Pricing Models: On-Demand, Reserved, Spot, Savings Plans

**Simple Storage Service (S3)**

- Create an S3 Bucket — Step-by-Step
- S3 Storage Classes: Standard, IA, One-Zone IA, Glacier, Intelligent-Tiering
- S3 Security: Bucket Policies, ACLs, Block Public Access, Pre-Signed URLs
- S3 Encryption: SSE-S3, SSE-KMS, SSE-C, Client-Side Encryption
- S3 Versioning and Lifecycle Management — Step-by-Step
- S3 Object Lock, Glacier Vault Lock, and Compliance
- S3 Performance, Multipart Upload, and Transfer Acceleration
- S3 Select, Glacier Select, DataSync, and Storage Gateway
- CloudFront CDN: Create Distribution, Signed URLs, and Cookies
- Snowball and Snowball Edge: Large-scale data transfer
- AWS Organizations and Consolidated Billing
- Cross-Region Replication and Cross-Account Bucket Sharing
- Athena vs Macie: Querying and securing S3 data
- HANDS-ON LAB: Static Website Hosting with Amazon S3

- HANDS-ON LAB: IAM Identity and Access Management

**MOD  
ULE  
04****EC2 & Compute Services****Elastic Compute Cloud (EC2)**

- EC2 Instance Types: General, Compute, Memory, Storage, and GPU Optimised
- Launch First EC2 Instance: Parts 1 & 2 — Step-by-Step
- Security Groups: Inbound/Outbound rules — Step-by-Step
- EBS 101: Volume types — gp3, io2, st1, sc1, and use cases
- EBS Volumes, Snapshots, and Encrypted Root Volumes — Step-by-Step
- AMI Types: EBS-backed vs Instance Store
- ENI vs ENA vs EFA: Networking interfaces explained
- Spot Instances, Spot Fleets, and EC2 Hibernate
- EC2 Placement Groups: Cluster, Spread, Partition
- CloudWatch 101: Metrics, Alarms, and Custom Dashboards — Step-by-Step
- AWS CLI Configuration and Usage — Step-by-Step
- IAM Roles for EC2, Bootstrap Scripts, and Instance Metadata
- EFS: Elastic File System and Amazon FSx (Windows/Lustre)
- HPC on AWS and AWS WAF Introduction
- HANDS-ON LAB: EC2 Instances, Tags, and Resource Groups
- HANDS-ON LAB: EC2 Roles and Instance Profiles

**MOD  
ULE  
05****Databases on AWS****Managed Database Services**

- Relational vs Non-Relational Databases: When to use each
- Create RDS Instance — Step-by-Step
- RDS: Backups, Multi-AZ, and Read Replicas — Step-by-Step
- Aurora: Architecture, Global Database, and Serverless v2
- DynamoDB: Tables, Items, Indexes, Streams, and DAX
- Advanced DynamoDB: Global Tables, TTL, and On-Demand Pricing
- Redshift: Data Warehousing, RA3 nodes, Spectrum
- ElastiCache: Redis vs Memcached — Caching strategies
- Database Migration Service (DMS): Homogeneous and heterogeneous migrations
- Caching Strategies: Lazy loading, Write-through, Session store
- EMR: Managed Hadoop/Spark for big data
- HANDS-ON LAB: RDS Multi-AZ and Read Replica Deployment

**MOD  
ULE  
06****Networking: Route 53, VPC & Advanced IAM****DNS & Route 53**

- DNS 101: A, AAAA, CNAME, MX, TXT records and TTL

- Register a Domain Name — Step-by-Step
- Route 53 Routing Policies: Simple, Weighted, Latency, Failover, Geolocation, Geoproximity, Multivalue
- Route 53 Health Checks and DNS-based Failover
- Networking Fundamentals: OSI Model, IPv4/IPv6, CIDR Notation

## Virtual Private Cloud (VPC)

- VPC Overview: Subnets, Route Tables, Internet Gateways, DHCP Options
- Create Custom VPC: Parts 1 & 2 — Step-by-Step
- NAT Instances vs NAT Gateways: When to use each
- Network ACLs vs Security Groups: Stateless vs Stateful
- VPC Flow Logs: Enabling, analysing, and troubleshooting
- Bastion Hosts and Jump Servers: Secure admin access
- AWS Direct Connect and Direct Connect Gateway
- Global Accelerator: Anycast routing and performance
- VPC Endpoints: Gateway (S3/DynamoDB) and Interface (PrivateLink)
- AWS Transit Gateway: Hub-and-spoke inter-VPC routing
- AWS VPN CloudHub and Site-to-Site VPN
- HANDS-ON LAB: Custom VPC with Public/Private Subnets
- HANDS-ON LAB: VPC Flow Logs for Network Monitoring

## Advanced IAM

- AWS Directory Service: AD Connector, Simple AD, AWS Managed AD
- IAM Policies: Identity-based, Resource-based, and Permission Boundaries
- AWS Resource Access Manager (RAM): Cross-account resource sharing
- AWS IAM Identity Centre (SSO): Centralised access management
- Cross-Account Access Patterns and Trust Relationships

### MOD ULE 07

## High Availability, Auto Scaling & CloudFormation

### HA Architecture

- Elastic Load Balancers: ALB, NLB, and CLB — differences and use cases
- Load Balancers and Health Checks — Step-by-Step
- Advanced Load Balancer Theory: Sticky sessions, Connection draining, SNI, X-Forwarded headers
- Auto Scaling Groups: Target tracking, Step scaling, Scheduled scaling
- Launch Configurations vs Launch Templates
- Building a Fault-Tolerant WordPress Site: Setup → EC2 → Resilience → Auto Scaling → Cleanup
- CloudFormation IaC: Templates, Stacks, StackSets, Nested Stacks, Change Sets
- Elastic Beanstalk: Managed PaaS deployment
- High Availability with Bastion Hosts and On-Premises AWS Strategies
- HANDS-ON LAB: Getting Started with CloudFormation

### MOD ULE 08

## Serverless, Application Integration & Security

### Serverless Computing

- AWS Lambda: Functions, Triggers, Layers, Concurrency, and Cold Starts

- Build a Serverless Web Page with API Gateway + Lambda — Step-by-Step
- Serverless Application Model (SAM): Local testing and deployments
- API Gateway: REST vs HTTP vs WebSocket APIs, Throttling, Usage Plans
- Event Processing Patterns: Fan-out, Saga, Event Sourcing

## Application Integration

- SQS: Standard vs FIFO queues, Visibility Timeout, Dead Letter Queues
- SNS: Topics, subscriptions, and fan-out architecture patterns
- Simple Workflow Service (SWF) and Kinesis Data Streams
- Amazon Cognito: User Pools, Identity Pools, and Web Identity Federation
- Elastic Transcoder and Media Processing
- HANDS-ON LAB: Scaling EC2 Using SQS

## Security Services

- AWS Shared Responsibility Model
- KMS: CMKs, Key Rotation, and Envelope Encryption
- CloudHSM: Dedicated hardware security modules
- Systems Manager Parameter Store and Secrets Manager
- AWS Shield Standard & Advanced: DDoS protection layers
- WAF Rules, Rule Groups, and Managed Rule sets
- Amazon GuardDuty, Amazon Inspector, and AWS Security Hub

### MOD ULE 09

## Docker & Containerisation

### Docker Fundamentals

- What is Containerisation? Containers vs Virtual Machines — Key differences
- Docker Architecture: Daemon, CLI, Images, Containers, and Registries
- Installing Docker: Linux, macOS, Windows (Docker Desktop)
- Core Docker Commands: run, build, pull, push, stop, rm, ps, exec, logs
- Working with Docker Images: Layers, caching, tagging, and versioning
- Writing Dockerfiles: FROM, RUN, COPY, EXPOSE, ENV, CMD, ENTRYPOINT
- Dockerfile Best Practices: Multi-stage builds, minimal base images, .dockerignore
- Docker Volumes: Persistent storage, bind mounts, and tmpfs
- Docker Networking: Bridge, Host, Overlay, and Macvlan networks

### Docker Compose & Registry

- Docker Compose: docker-compose.yml structure, services, health checks, depends\_on
- Multi-Container Applications: App + Database + Cache pattern
- Amazon Elastic Container Registry (ECR): Push, pull, and lifecycle policies
- Docker Scout: Image vulnerability scanning and software bill of materials
- Docker Security: Non-root users, read-only filesystems, resource limits
- Docker in CI/CD Pipelines: Build, tag, and push automation
- HANDS-ON LAB: Containerise a Node.js Application with Docker
- HANDS-ON LAB: Multi-Stage Docker Build for Production Images

### MOD ULE 10

## Amazon ECS — Elastic Container Service

## ECS Core Concepts

- ECS Architecture: Clusters, Task Definitions, Services, and Tasks
- ECS Launch Types: EC2 vs AWS Fargate — cost and operational trade-offs
- Task Definitions: CPU/Memory, networking mode, IAM task roles, volumes
- ECS Services: Desired count, deployment strategies, and health checks
- ECS Clusters: Creating and managing EC2 and Fargate clusters
- ECS Service Auto Scaling: Target tracking and step scaling policies
- Load Balancing with ECS: ALB, NLB, and service discovery

## ECS Advanced & Operations

- AWS Fargate: Serverless containers — pricing model and task sizing
- ECS with ECR: Pulling images securely from private registries
- ECS Task Placement Strategies and Constraints
- ECS Rolling Updates and Blue/Green Deployments with AWS CodeDeploy
- ECS Logging: CloudWatch Logs, FireLens, and log routing configuration
- ECS Secrets: Integration with Parameter Store and Secrets Manager
- ECS Networking: awsvpc networking mode, security groups per task
- Monitoring ECS: CloudWatch Container Insights and custom alarms
- HANDS-ON LAB: Deploy a Containerised Application on ECS Fargate
- HANDS-ON LAB: ECS Blue/Green Deployment with CodeDeploy

### MOD ULE 11

## Kubernetes & Amazon EKS

### Kubernetes Core Concepts

- What is Kubernetes? Container orchestration at scale — architecture overview
- Kubernetes Components: Control Plane (API Server, etcd, Scheduler, Controller Manager)
- Worker Node Components: kubelet, kube-proxy, Container Runtime
- Core Objects: Pods, ReplicaSets, Deployments, StatefulSets, DaemonSets, Jobs, CronJobs
- Services: ClusterIP, NodePort, LoadBalancer, ExternalName
- ConfigMaps and Secrets: Externalising application configuration
- Namespaces: Logical isolation and resource quota management
- Persistent Volumes, PVCs, and StorageClasses
- Resource Requests and Limits: CPU and memory management

### kubectl & Manifest Files

- kubectl: Core commands — get, describe, apply, delete, exec, logs, port-forward
- Writing YAML Manifests: Deployments, Services, and Ingress resources
- Horizontal Pod Autoscaler (HPA) and Vertical Pod Autoscaler (VPA)
- Rolling Updates and Rollbacks: Zero-downtime deployment strategies
- Liveness, Readiness, and Startup Probes: Health checking in Kubernetes
- Kubernetes RBAC: Roles, ClusterRoles, RoleBindings, ServiceAccounts
- Network Policies: Pod-level traffic control and micro-segmentation
- Ingress Controllers: NGINX Ingress and AWS ALB Ingress Controller

### Amazon EKS — Managed Kubernetes

- EKS Architecture: Managed control plane, managed node groups, Fargate profiles

- Creating an EKS Cluster: eksctl, AWS Console, and Terraform
- EKS Node Groups: Managed vs self-managed, Spot instance integration
- EKS Fargate Profiles: Serverless Kubernetes pods on AWS
- EKS Networking: VPC CNI plugin, Pod networking, CoreDNS
- EKS IAM Integration: IRSA (IAM Roles for Service Accounts)
- EKS Cluster Autoscaler and Karpenter: Cost-efficient node provisioning
- EKS Add-ons: CoreDNS, kube-proxy, AWS VPC CNI, EBS CSI, EFS CSI drivers
- EKS Logging and Observability: Control plane logs, Container Insights, Prometheus
- Helm Package Manager: Charts, Repositories, and Values files
- HANDS-ON LAB: Deploy a Microservices Application on EKS
- HANDS-ON LAB: EKS with ALB Ingress Controller and Cluster Auto Scaling

**MOD  
ULE  
12****Jenkins & CI/CD Pipeline Automation****Jenkins Fundamentals**

- What is CI/CD? Continuous Integration, Delivery, and Deployment explained
- Jenkins Architecture: Controller, Agents, Executors, and Workspaces
- Installing Jenkins: Docker, EC2, and Kubernetes (Helm chart)
- Jenkins Dashboard: Jobs, Builds, Console Output, and Blue Ocean UI
- Freestyle Projects: Build steps, triggers, and post-build actions
- SCM Integration: GitHub, GitLab, and Bitbucket webhooks and polling
- Jenkins Credentials Store: Secrets, SSH keys, API tokens, and Username/Password
- Essential Jenkins Plugins: Git, Docker, AWS Steps, Slack, SonarQube Scanner

**Declarative Pipelines & Jenkinsfile**

- Pipeline as Code: Jenkinsfile fundamentals and best practices
- Declarative Pipeline syntax: pipeline, agent, stages, steps, post blocks
- Scripted Pipeline: Groovy DSL and advanced pipeline logic
- Multi-Branch Pipelines: Automatic branch detection and PR builds
- Parallel Stages: Speeding up pipelines with concurrent execution
- Pipeline Parameters, Input Steps, and Approval Gates
- Environment Variables, withCredentials, and Credentials Binding
- Triggering Pipelines: Webhooks, scheduled builds, and upstream triggers
- Shared Libraries: Reusable pipeline code across projects

**Jenkins & AWS Integration**

- Building Docker Images in Jenkins Pipelines
- Pushing Images to Amazon ECR from Jenkins
- Deploying to ECS from Jenkins: Rolling and Blue/Green strategies
- Deploying to EKS from Jenkins using kubectl and Helm
- Jenkins with AWS CLI and IAM Instance Profiles
- Distributed Builds: Jenkins agents on EC2 and ECS Fargate
- HANDS-ON LAB: Full CI/CD Pipeline — Git to Docker to ECS Deployment
- HANDS-ON LAB: Multi-Branch Pipeline with GitHub Webhooks and Approvals

**MOD  
ULE  
13****SonarQube — Code Quality & Security Analysis**

## SonarQube Fundamentals

- What is Code Quality? Technical debt, maintainability, reliability, and security
- SonarQube Architecture: Server, Database, Compute Engine, and Scanners
- Installing SonarQube: Docker Compose, EC2, and SonarCloud (SaaS)
- SonarQube Dashboard: Projects, Issues, Measures, and Activity views
- Quality Gates: Configuring pass/fail conditions for CI/CD pipeline gating
- Quality Profiles: Language-specific rule sets and custom rule configuration
- Issue Types: Bug, Vulnerability, Code Smell, and Security Hotspot

## SonarQube Scanners & Analysis

- SonarScanner CLI: Manual project analysis and properties configuration
- SonarScanner for Maven, Gradle, and npm projects
- sonar-project.properties: Complete configuration reference
- Code Coverage Integration: JaCoCo (Java), Istanbul (JavaScript), pytest-cov (Python)
- Duplicate Code Detection and Cognitive Complexity metrics
- Security Vulnerability Detection: OWASP Top 10 and CWE mapping
- Branch Analysis, Pull Request Decoration, and New Code definition

## SonarQube in CI/CD Pipelines

- Jenkins + SonarQube: SonarQube Scanner Plugin setup and configuration
- Quality Gate as a Pipeline Gate: Breaking the build on Quality Gate failure
- SonarQube Webhooks: Notifying Jenkins and Slack of analysis results
- SonarCloud: Cloud-hosted analysis for public and private repositories
- Shift-Left Security: Catching vulnerabilities in development, not production
- HANDS-ON LAB: Integrate SonarQube into Jenkins CI/CD Pipeline
- HANDS-ON LAB: Quality Gate Enforcement — Fail the Build on Critical Issues

### MOD ULE 14

## AWS DevOps Toolchain & Infrastructure as Code

### AWS Native CI/CD Services

- AWS CodeCommit: Managed Git repositories and branch protection
- AWS CodeBuild: Managed build service, buildspec.yml, and caching
- AWS CodeDeploy: EC2, Lambda, and ECS deployment strategies
- AWS CodePipeline: Visual pipeline orchestration and stage actions
- AWS CodeArtifact: Managed artifact and package repository
- AWS CodeStar: Unified project management and DevOps dashboard

### Infrastructure as Code (IaC)

- AWS CloudFormation: Templates, Stacks, StackSets, and Nested Stacks
- CloudFormation Change Sets, Drift Detection, and Stack Policies
- Terraform on AWS: Providers, Resources, State Management, and Modules
- Terraform Workspaces and Remote State with S3 + DynamoDB locking
- AWS CDK: Infrastructure as Code with TypeScript and Python
- AWS Systems Manager: Session Manager, Run Command, Patch Manager, Inventory
- HANDS-ON LAB: Full AWS CI/CD Pipeline — CodePipeline + CodeBuild + CodeDeploy to ECS
- HANDS-ON LAB: Terraform EKS Cluster Deployment from scratch

## Python for Cloud & DevOps Engineering

### Python Fundamentals

- Why Python? Role of Python in Cloud, DevOps, and Automation engineering
- Python Installation: Setting up Python 3, pip, virtualenv, and VS Code
- Python Syntax: Indentation, comments, and code structure
- Data Types: Strings, Integers, Floats, Booleans, and type conversion
- Variables and Constants: Naming conventions and best practices
- Operators: Arithmetic, comparison, logical, bitwise, and assignment
- Input and Output: print(), input(), f-strings, and string formatting
- Control Flow: if, elif, else — conditional logic
- Loops: for loops, while loops, break, continue, pass
- Functions: Defining, calling, parameters, return values, and scope
- Lambda Functions: Anonymous functions and use cases
- Exception Handling: try, except, finally, raise, and custom exceptions

### Data Structures & Collections

- Lists: Indexing, slicing, appending, sorting, list comprehensions
- Tuples: Immutable sequences and when to use them
- Dictionaries: Key-value pairs, nesting, iteration, and comprehensions
- Sets: Unique elements, union, intersection, and difference operations
- Stacks and Queues: Implementing with lists and collections.deque
- Working with JSON: json.loads(), json.dumps(), reading/writing JSON files
- Working with CSV: csv module, reading and writing tabular data
- File I/O: open(), read(), write(), with context manager

### Object-Oriented Python

- Classes and Objects: Defining classes, creating instances
- Constructors: \_\_init\_\_ method and instance attributes
- Instance Methods, Class Methods, and Static Methods
- Inheritance and Polymorphism: Extending classes, method overriding
- Encapsulation: Private attributes and property decorators
- Magic Methods: \_\_str\_\_, \_\_repr\_\_, \_\_len\_\_, \_\_eq\_\_
- Modules and Packages: Importing, creating, and organising code
- Python Standard Library: os, sys, shutil, pathlib, datetime, re

### Python for Scripting & Automation

- Automating File and Directory Operations with os and pathlib
- Regular Expressions with re: Pattern matching, search, replace
- Working with APIs: requests library — GET, POST, PUT, DELETE
- Parsing JSON API Responses and handling HTTP status codes
- Environment Variables: os.environ and python-dotenv for config management
- Subprocess Module: Running shell commands and scripts from Python
- Scheduling Tasks: schedule library and cron job automation with Python
- Logging: logging module — levels, handlers, formatters, and rotating logs
- Unit Testing with pytest: Test functions, fixtures, assertions, and coverage

## Python & AWS (Boto3)

- Introduction to Boto3: AWS SDK for Python — installation and setup
- AWS Authentication: IAM credentials, profiles, and IAM roles with Boto3
- Boto3 Session, Client, and Resource interfaces — when to use each
- S3 with Boto3: Create buckets, upload/download files, list objects, presigned URLs
- EC2 with Boto3: Launch instances, describe, start, stop, and terminate
- IAM with Boto3: Create users, roles, and manage policies programmatically
- Lambda with Boto3: Invoke functions and manage deployments from Python
- CloudWatch with Boto3: Create alarms, query metrics, and filter log events
- DynamoDB with Boto3: Put, get, query, scan items using Table resource
- SSM Parameter Store with Boto3: Read and write secure configuration values
- SQS with Boto3: Send, receive, and delete messages from queues
- SNS with Boto3: Publish messages and manage topic subscriptions
- Error Handling in Boto3: ClientError, BotoCoreError, waiters, and retries
- HANDS-ON LAB: Python Script to Automate EC2 Start/Stop on a Schedule
- HANDS-ON LAB: S3 File Sync Tool with Boto3 and Progress Tracking
- HANDS-ON LAB: CloudWatch Log Analyser — Parse and Alert on Error Patterns

## Python in DevOps Pipelines

- Python Scripts in Jenkins Pipelines: sh() steps and Python-based build logic
- Python for Infrastructure Validation: Checking AWS resources post-deploy
- Writing Custom SonarQube Quality Reports with Python and the API
- Python with Docker: Building images and managing containers via docker-py SDK
- Kubernetes Python Client: Listing pods, creating deployments, checking health
- Python-based CLI Tools: argparse, click — building reusable DevOps utilities
- HANDS-ON LAB: Build a Python CLI DevOps Toolkit for AWS resource management
- HANDS-ON LAB: Python Kubernetes Health-Check Script integrated into CI/CD

### MOD ULE 16

## Cloud Migration Strategies

## Migration Frameworks & Patterns

- The 6 R's of Migration: Rehost, Replatform, Repurchase, Refactor, Retire, Retain
- Oracle Database Migration — On-Premises to AWS: Lift & Shift and Lift & Shape
- MySQL Database Migration — On-Premises to AWS: Lift & Shift and Lift & Shape
- AWS Database Migration Service (DMS): Continuous data replication
- AWS Server Migration Service (SMS): Incremental VM replication to AWS
- AWS Application Migration Service (MGN): Automated lift-and-shift
- AWS Migration Hub: Centralised discovery and migration tracking
- Containerising Legacy Applications for ECS/EKS migration

### MOD ULE 17

## Capstone Project & Certification Preparation

## Capstone Project: End-to-End Cloud-Native DevOps Platform

- Design a multi-tier VPC: public/private subnets, NAT Gateway, flow logs
- Containerise a full-stack application with Docker and push to ECR

- Deploy on ECS Fargate and/or EKS with ALB Ingress and Auto Scaling
- Build a complete CI/CD pipeline: GitHub → Jenkins → SonarQube → ECR → ECS/EKS
- Write Python Boto3 automation scripts for resource provisioning and health checks
- Implement Infrastructure as Code with Terraform or CloudFormation
- Configure CloudWatch dashboards, alarms, and Container Insights
- Apply IAM least-privilege, KMS encryption, and Security Hub findings
- Live architectural walkthrough and peer review presentation

### AWS SAA-C03 Exam Preparation

- SAA-C03 Exam Blueprint: 4 Domains — Design Secure, Resilient, High-Performing, Cost-Optimised Architectures
- 2 x Full-Length Practice Examinations (130 questions each, timed)
- Detailed answer explanations for all practice questions
- Exam strategy: Elimination techniques, flagging, and time management
- Targeted revision sessions for identified weak areas
- AWS Well-Architected Framework: 6 Pillars deep-dive
- Key Whitepapers: Security, Reliability, and Cost Optimisation

## Career Outcomes & Target Roles

Graduates of this programme are prepared for the following in-demand cloud and DevOps engineering roles:

| Role                             | Target Salary Range (USD) |
|----------------------------------|---------------------------|
| Cloud Solutions Architect        | \$130,000 – \$200,000+    |
| DevOps Engineer                  | \$120,000 – \$185,000+    |
| Cloud Infrastructure Engineer    | \$110,000 – \$170,000+    |
| Kubernetes / EKS Engineer        | \$130,000 – \$195,000+    |
| Site Reliability Engineer (SRE)  | \$140,000 – \$210,000+    |
| CI/CD & Automation Engineer      | \$115,000 – \$175,000+    |
| Python Cloud Automation Engineer | \$115,000 – \$180,000+    |
| Cloud Security Engineer          | \$130,000 – \$200,000+    |
| Platform Engineer                | \$125,000 – \$190,000+    |