

AWS DevOps, EKS and Terraform

Course Summary

This 5-day hands-on course provides practical knowledge of AWS DevOps, Terraform, Docker, ECS/Fargate, and Amazon EKS. Participants will learn how to build CI/CD pipelines, automate AWS infrastructure using Terraform, containerize applications, deploy workloads on ECS/Fargate and EKS, and implement end-to-end application delivery on AWS. The course progresses from AWS DevOps and CI/CD fundamentals to Infrastructure as Code with Terraform, followed by containerization and ECS, and finally Kubernetes and Amazon EKS. The program concludes with an end-to-end implementation combining Terraform, CI/CD, ECR, and EKS.

Course Objectives

By the end of this course, participants will be able to:

- Understand DevOps principles and AWS DevOps services.
- Build and manage CI/CD pipelines using CodeCommit, CodeBuild, CodeDeploy, and CodePipeline.
- Implement different application deployment strategies, including blue/green deployments.
- Use Terraform for Infrastructure as Code and automate AWS infrastructure provisioning. Provision EKS infrastructure using Terraform.
- Manage Terraform variables, state, remote backends, modules, and reusable configurations.
- Understand Docker containerization and build application container images.
- Use Amazon ECR to store and manage container images.
- Deploy containerized applications using Amazon ECS and AWS Fargate.
- Understand Kubernetes architecture and core concepts.
- Create and manage Amazon EKS clusters and managed node groups.
- Deploy, configure, expose, and scale applications running on EKS.
- Implement EKS networking, ingress, IAM, security, and autoscaling concepts.
- Build an end-to-end CI/CD pipeline for deploying containerized applications to EKS.
- Apply AWS DevOps, Terraform, container, and EKS best practices and troubleshooting techniques.

Course Outline:

Day 1 — AWS DevOps & CI/CD

- DevOps Principles & AWS DevOps Services
- CodeCommit
- CodeBuild
- CodeDeploy
- CodePipeline
- IAM for CI/CD
- Build Artifacts
- Deployment Strategies
- CI/CD Best Practices

Labs:

1. Build CI/CD Pipeline: CodeCommit → CodeBuild → CodeDeploy → EC2
 2. Implement Blue/Green Deployment
-

Day 2 — Terraform on AWS

- Infrastructure as Code
- Terraform Architecture & Workflow
- Providers & Resources
- Variables, Locals & Outputs
- Terraform State
- Remote Backend
- Terraform Modules
- Workspaces
- Terraform Best Practices

Labs:

1. Provision AWS Infrastructure Using Terraform
2. Terraform Remote State & Reusable Modules

Day 3 — Docker, ECR & ECS/Fargate

- Container Fundamentals
- Docker Architecture
- Dockerfile
- Images & Containers
- Docker Networking
- Amazon ECR
- ECS Architecture
- Task Definitions
- ECS Services
- Fargate
- Application Load Balancer
- CloudWatch Logs

Labs:

1. Dockerize Application & Push Image to Amazon ECR
2. Deploy Containerized Application to ECS/Fargate with CI/CD

Day 4 — Amazon EKS Fundamentals

- Kubernetes Fundamentals
- Kubernetes Architecture
- EKS Architecture
- EKS Control Plane
- Managed Node Groups
- EKS Networking
- kubectl
- Pods
- Deployments

- Services
- ConfigMaps
- Secrets
- ECR Integration

Labs:

1. Create EKS Cluster & Deploy Application
 2. Expose and Configure Application on EKS
-

Day 5 — Advanced EKS, Terraform & CI/CD

- EKS Networking Deep Dive
- AWS Load Balancer Controller
- Ingress
- EKS Pod Identity
- EKS Security
- HPA
- Node Autoscaling
- Karpenter Overview
- Helm
- Terraform with EKS
- CI/CD to EKS
- Monitoring & Troubleshooting
- EKS Best Practices

Labs:

1. Provision EKS Infrastructure Using Terraform
2. End-to-End CI/CD Pipeline: CodeCommit → CodeBuild → ECR → EKS