CONCEPT

DevOps and CI/CD Pipelines: Continuous Integration, Delivery and Kubernetes Deployment

21 – 25 June 2027
London - Central London four-star



DevOps and CI/CD Pipelines: Continuous Integration, Delivery and Kubernetes Deployment

Ref: 611_17966 **Date:** 21 – 25 June 2027 **Location:** London - Central London four-star **Fees:** 25300 SAR

Introduction:

Software teams lose weeks when code waits in long-lived branches, builds break at integration time and releases depend on manual server changes nobody can repeat. DevOps and CI/CD pipelines replace those hand-offs with a version-controlled, automated flow from commit to production. This Core Concept course trains developers, operations and platform engineers to structure Git branching, build continuous integration pipelines, package applications as Docker images, deploy them to Kubernetes, provision environments as code and select release strategies. Participants finish with a CI/CD Pipeline and Deployment Design for a case application.

Course Objectives:

- Map the commit-to-production value stream of a service and baseline it with DORA delivery metrics
- Select a Git branching strategy and pull request workflow that supports frequent integration into the main branch
- Build a pipeline-as-code continuous integration workflow that compiles, runs automated test stages and publishes versioned artifacts
- Package an application as a Docker image and deploy it to Kubernetes using pods, services and deployments
- Provision environments with infrastructure as code and configuration management, and choose blue-green, canary or feature flag releases
- Produce a CI/CD Pipeline and Deployment Design for a case application, including secrets handling and feedback loops

Target Audience:

- Software developers who commit code daily and own the build and packaging of their services
- Operations and infrastructure engineers who provision servers and environments and carry out deployments
- Platform engineers who provide shared pipeline runners, container registries and cluster environments to product teams
- Build and release engineers who maintain pipeline definitions, artifact repositories and release schedules
- Technical team leads who decide how their team integrates, packages and ships changes

Course Outline:

Day 1: DevOps Principles, Delivery Flow and Version Control

- DevOps Principles: Shared Ownership Between Development and Operations
- Value Stream Mapping of the Commit-to-Production Path
- DORA Delivery Metrics: Change Lead Time, Deployment Frequency and Change Fail Rate
- Git Workflow Fundamentals: Commits, Merges, Rebases and Pull Requests
- Branching Strategies Compared: Trunk-Based Development and Feature Branch Models

Day 2: Continuous Integration Pipeline Architecture

- Pipeline-as-Code Definition Files: Stages, Jobs and Triggers
- Build Automation, Dependency Caching and Reproducible Builds
- Automated Test Stages in the Pipeline: Unit, Integration and Smoke Runs
- Artifact Versioning and Binary Repository Management
- Build Runner Categories: Hosted, Self-Managed and Ephemeral Agents

Day 3: Containers, Kubernetes and Continuous Delivery

- Dockerfile Authoring, Multi-Stage Builds and Image Layer Optimisation
- Container Registry Tagging, Image Promotion and Retention Rules
- Kubernetes Objects: Pods, Services and Deployments in Declarative YAML
- Rolling Updates, Rollback and Replica Scaling on a Kubernetes Cluster
- Continuous Delivery Versus Continuous Deployment: Environment Promotion and Approval Steps

Day 4: Infrastructure as Code, Release Strategies and Pipeline Security

- Infrastructure as Code: Declarative Provisioning, State Files and Drift Detection
- Configuration Management with Idempotent Playbooks and Desired-State Tools
- Release Strategies: Blue-Green Switching, Canary Traffic Splits and Feature Flags
- Pipeline Secrets Handling with Dependency and Image Vulnerability Scanning Overview
- Deployment Feedback Loops: Pipeline Dashboards and Post-Release Health Checks

Day 5: Lab: CI/CD Pipeline and Deployment Design for a Case Application

- Case Application Brief: Service Architecture and Current Release Process Assessment
- Lab: Integration Pipeline and Container Image Build for the Case Service
- Lab: Kubernetes Deployment Manifests and Environment Promotion Path
- CI/CD Pipeline and Deployment Design Drafting: Stages, Tool Categories and Release Strategy
- Pipeline Design Presentation and Peer Challenge

Skills You Will Gain:

- Value Stream Mapping
- Git Branching Strategy
- Pipeline-as-Code Authoring
- Artifact Management
- Container Image Building
- Kubernetes Workload Deployment
- Infrastructure Provisioning Automation
- Progressive Release Planning

Why Attend This Course:

- Return with a CI/CD Pipeline and Deployment Design built for a realistic case application
- Replace manual, error-prone release steps with automated pipelines your team can rerun on every commit
- Gain hands-on lab practice writing pipeline definitions, Dockerfiles and Kubernetes manifests
- Compare delivery practices with developers and engineers from other organisations and sectors

Conclusion:

Fast, dependable software delivery comes from small changes flowing through automated, repeatable pipelines rather than from large manual releases. The course moves from DevOps principles, delivery metrics and Git branching, through continuous integration pipelines and artifact handling, to Docker images, Kubernetes workloads and continuous delivery, then to infrastructure as code, configuration management, release strategies and pipeline secrets. The final day applies these methods in a lab and produces a CI/CD Pipeline and Deployment Design for a case application.