DEVOPS ADVANCED 3 DAYS

Containerization with Docker

Build a strong foundation in containerization using Docker by understanding its architecture and core components. Learn how to build, run, network, secure, and deploy containerized applications while managing data persistence and container networking effectively.

Why Docker?

Improved efficiency

Witness increased productivity and faster time to market for products and services with the efficient deployment of applications and services.

Scalability

Docker's containerization allows businesses to easily scale their applications up or down based on demand.

Simplified deployment

Reduce the risk of configuration errors and inconsistencies between development, testing, and production environments.

What you'll be able to do

- Gain an understanding of Docker's architecture, including Docker Engine, Docker CLI, and Docker Registry
- Learn how to create and manage containers using Docker commands
- Learn how to connect containers together to create a network, and configure container networks
- Learn how to use volumes to persist data in Docker containers and share data between containers
- Learn how to deploy applications using Docker
- Learn how to implement security best practices when working with Docker

Who this is for

- Developers
- System Administrators
- DevOps Professionals
- Cloud Engineers
- Data Engineers
- Security Engineers
- IT architects
- Quality assurance engineers

Curriculum

01 Containerization Fundamentals and Docker Architecture

- Why containerization exists and the problems it solves for delivery teams
- Docker Engine, CLI, and registry roles in the architecture
- Containers versus virtual machines: isolation, density, and trade-offs
- When Docker is the right packaging choice for applications and services

02 Docker Images and Containers

- What an image contains: code, runtime, tools, libraries, and settings
- Running containers as instances of images across environments
- Essential docker run, ps, logs, exec, and inspect workflows
- Image layers and tagging basics for repeatable builds

03 Building Images with Dockerfile

- Dockerfile instructions for base image, packages, copy, and entrypoint
- Multi-stage builds and keeping images small
- Build, tag, and validate images locally before sharing
- **Hands-on:** Build an application image from a Dockerfile and run it as a container

04 Docker Networking

- Built-in networking for container-to-container and host communication
- Bridge networks, port publishing, and service discovery basics
- Connecting multiple containers for a simple application topology
- Common networking pitfalls when moving from local to shared environments

05 Docker Volumes and Data Persistence

- Why container filesystems are ephemeral and when that matters
- Named volumes and bind mounts for persistent and shared data
- Sharing data between containers safely
- Backup and cleanup considerations for volume data

06 Docker Hub and Image Distribution

- Using Docker Hub and registries to store and share images
- Pulling official images and publishing team images
- Tagging strategies for discoverability and rollback
- Access and trust basics when consuming public images

07 Multi-Container Apps with Docker Compose

- Defining services, networks, and volumes in Compose YAML
- Starting, stopping, and managing multi-container stacks with one command
- Environment configuration patterns for local development
- **Hands-on:** Run a multi-service application with Docker Compose

08 Orchestration Basics with Docker Swarm

- What Swarm provides for clustering and service orchestration
- When Swarm is enough versus when Kubernetes becomes the target platform
- Service definitions, replicas, and basic rolling update concepts
- Practical limits of Swarm for modern enterprise platforms

09 Security and Docker Best Practices

- Host and container hardening features: user namespaces, seccomp, and AppArmor awareness
- Keeping images minimal, preferring trusted bases, and setting resource constraints
- Avoiding secrets in images and running as non-root where possible
- Practical checklist for secure day-to-day Docker usage

10 Logging, Operations, and Docker in Production

- Logging options: local drivers, syslog, and common shipping patterns
- Production concerns: monitoring, scaling, backup, and security operations
- Operational habits that keep container platforms reliable
- **Hands-on:** Apply a production readiness checklist to a Compose-based application