Faysal Aziz

Docker Foundations

Building Modern Containerized Applications in 2 days

FA-0178 · 2 days

DevOps, Cloud & Infrastructure

Introduction

Practical Containerization Skills for Modern Developers and IT Professionals

Docker has transformed the way modern applications are built, tested, deployed, and maintained. As organizations increasingly adopt cloud-native architectures, DevOps practices, microservices, and scalable deployment pipelines, containerization has become a core operational requirement rather than a niche engineering skill. Docker provides a lightweight and highly portable platform that enables developers and infrastructure teams to package applications consistently across development, testing, staging, and production environments.

This two-day introductory course is designed to provide participants with a strong practical foundation in Docker technologies, container concepts, and real-world deployment workflows. The training focuses on hands-on operational knowledge aligned with current industry practices, including container image management, Docker networking, storage, Docker Compose, container security considerations, and integration into modern DevOps workflows. Current enterprise trends such as cloud-native development, container security hardening, remote development environments, and Kubernetes integration concepts are also incorporated to ensure relevance to today's infrastructure landscape.

The course is delivered by an instructor with over 30 years of industry experience who emphasizes practical, production-oriented knowledge and real operational scenarios instead of purely academic theory.

Learning outcomes

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

- Understand the principles of containerization and modern application deployment

- Differentiate between containers and traditional virtual machines
- Install and configure Docker environments on Linux and Windows systems
- Work with Docker Engine, Docker Desktop, and Docker CLI tools
- Build, manage, and optimize Docker container images
- Create and manage Dockerfiles using industry best practices
- Deploy and manage standalone containers
- Configure Docker networking and persistent storage
- Use Docker Compose to manage multi-container applications
- Work with Docker Hub and container registries
- Understand container security fundamentals and image hardening concepts

Prerequisites

- Basic understanding of operating systems and system administration
- Familiarity with command-line interfaces
- Basic understanding of networking concepts
- Exposure to software development or IT infrastructure environments is beneficial
- Basic Linux knowledge is recommended but not mandatory
- No prior Docker or containerization experience is required

Training outline

1. Introduction to Containerization and Modern Application Deployment

Evolution of application deployment models

1. Traditional deployments
2. Virtual machines versus containers
3. Infrastructure efficiency challenges
4. Application portability concerns
5. Modern cloud-native application requirements

Understanding Containerization Concepts

1. Operating system level virtualization

2. Container runtime concepts
3. Isolation and namespaces
4. Resource allocation and control groups
5. Stateless versus stateful workloads

Docker Overview and Industry Adoption

1. Docker architecture overview
2. Docker Engine components
3. Docker Desktop overview
4. Docker ecosystem and tooling
5. Current container industry trends
6. Cloud-native development practices
7. Containers in DevOps workflows
8. Enterprise adoption patterns

Containers and Modern Infrastructure

1. Microservices architecture overview
2. CI/CD integration concepts
3. Hybrid cloud deployment models
4. Remote development environments
5. Introduction to orchestration platforms
6. Docker and Kubernetes relationship

2. Docker Installation and Environment Preparation

Docker Installation Procedures

1. Linux installation workflows
2. Windows installation workflows
3. Docker Desktop installation
4. Verifying Docker installation
5. Docker version management

Docker Environment Configuration

1. Docker daemon concepts
2. Docker service management
3. User permissions and groups
4. Docker CLI configuration
5. Environment validation procedures

Development Environment Setup

1. Terminal and shell preparation
2. Visual Studio Code integration
3. Docker extensions and plugins
4. Container development workflows
5. Local development best practices

3. Docker Architecture and Core Components

Docker Engine Fundamentals

1. Docker daemon
2. Docker client
3. Docker API
4. Docker runtime operations
5. Container lifecycle management

Docker Objects and Components

1. Images
2. Containers
3. Networks
4. Volumes
5. Registries

Understanding Container Lifecycle

1. Creating containers

2. Starting containers
3. Stopping containers
4. Restarting containers
5. Removing containers
6. Inspecting running workloads

4. Working with Docker Images

Docker Image Fundamentals

1. Layered image architecture
2. Image immutability concepts
3. Base images and parent images
4. Official images versus custom images
5. Image repositories and tags

Managing Images

1. Pulling images from Docker Hub
2. Searching for images
3. Tagging images
4. Removing images
5. Image inspection operations

Docker Hub and Container Registries

1. Docker Hub overview
2. Public versus private registries
3. Authentication and access control
4. Publishing custom images
5. Registry operational considerations

Image Optimization and Best Practices

1. Lightweight image selection
2. Multi-stage builds
3. Reducing image size

4. Dependency management
5. Version consistency strategies

5. Building Containers with Dockerfiles

Dockerfile Fundamentals

1. Dockerfile syntax
2. Docker build process
3. Layer caching concepts
4. Build context management
5. Docker build lifecycle

Dockerfile Instructions

1. FROM
2. RUN
3. COPY
4. ADD
5. CMD
6. ENTRYPOINT
7. ENV
8. EXPOSE
9. WORKDIR
10. USER

Building Custom Images

1. Creating application images
2. Building web server containers
3. Packaging dependencies
4. Runtime environment preparation
5. Image testing procedures

Dockerfile Best Practices

1. Efficient layering

2. Security considerations
3. Image maintainability
4. Build reproducibility
5. Enterprise image standards

6. Running and Managing Containers

Container Operations

1. Running containers
2. Interactive containers
3. Detached mode operations
4. Port mappings
5. Environment variables
6. Resource constraints

Monitoring and Troubleshooting Containers

1. Viewing logs
2. Inspecting containers
3. Container process management
4. Resource monitoring
5. Troubleshooting failed containers

Executing Commands in Containers

1. Interactive shell access
2. Process execution
3. File inspection
4. Administrative operations
5. Debugging workflows

7. Docker Networking Fundamentals

Docker Networking Concepts

1. Container communication

2. Bridge networks
3. Host networking
4. Overlay networking overview
5. Network isolation concepts

Managing Docker Networks

1. Creating custom networks
2. Connecting containers
3. Network inspection
4. Service discovery concepts
5. DNS resolution inside containers

Publishing and Exposing Services

1. Port forwarding
2. External access configuration
3. Reverse proxy concepts
4. Multi-container communication
5. Security considerations for networking

8. Docker Storage and Persistent Data

Understanding Container Storage

1. Ephemeral container storage
2. Persistent data requirements
3. Storage lifecycle concepts
4. Data management challenges

Docker Volumes

1. Named volumes
2. Anonymous volumes
3. Volume management
4. Sharing data between containers
5. Backup considerations

Bind Mounts and File Sharing

1. Host-to-container mappings
2. Development workflow integration
3. Configuration management
4. File permission considerations
5. Security implications

9. Multi-Container Applications with Docker Compose

Introduction to Docker Compose

1. Compose architecture
2. YAML configuration structure
3. Multi-container application concepts
4. Service definitions

Docker Compose Operations

1. Starting applications
2. Stopping services
3. Scaling containers
4. Environment variable management
5. Container dependency handling

Building Multi-Service Applications

1. Web application containers
2. Database integration
3. Internal networking
4. Persistent storage integration
5. Service orchestration basics

Compose Operational Best Practices

1. Configuration organization
2. Environment separation

3. Deployment consistency
4. Development workflow optimization

10. Docker Security Fundamentals

Container Security Concepts

1. Container isolation boundaries
2. Runtime security considerations
3. Privileged containers
4. Root versus non-root containers

Image Security

1. Trusted images
2. Official repositories
3. Vulnerability scanning concepts
4. Secure image sourcing
5. Hardened image concepts

Security Best Practices

1. Least privilege principles
2. Secret management overview
3. Limiting attack surfaces
4. Image update strategies
5. Security compliance considerations

Disclaimer

This course outline is intended to serve as a general training framework and guideline only. The instructor reserves the right to modify, reorder, expand, condense, or otherwise amend the course content, sequence of topics, practical exercises, and delivery approach as deemed appropriate based on audience profile, participant experience levels, technological updates, operational requirements, and time constraints. Any such adjustments may be made without prior notice in order to ensure the most effective and relevant learning experience.

© 2026 Faysal Aziz · Kuala Lumpur

https://faysalaziz.com/courses/docker-foundations/