

Docker Container Compact: Basics to Orchestration (DCK)

ID DCK Price 1,990.— €excl. tax) Duration 3 days

This text was automatically translated. Please click <u>here</u> to read the original German version.

Who should attend

- Software developers
- · Software architects
- System administrators
- · DevOps engineers

Prerequisites

- · First experiences with the terminal/bash
- · Working with files and directories in the terminal
- Basic concepts of network technology (IP addresses, ports, DNS)
- Basic understanding of software development and IT infrastructures
- · Familiarity with the concepts of virtualization

Course Objectives

The course teaches the use of Docker in practice and the technical background to it. The use is explained under both Windows and Linux, with the focus on Linux. However, the content can easily be transferred to Windows. The first day covers use cases and the basics of containers. The use cases provide an insight into the situations in which the use of containers makes sense. The basics enable participants to use containers safely. The second day includes Docker Compose, creating your own images and securing containers. With this knowledge, containers can be developed and operated securely. The third day provides a deep insight into debugging with containers, the Docker architecture and the use of containers in development environments, CI/CD systems and clusters. The course is therefore also explicitly aimed at teams with different levels of experience with Docker and containers.

Course Content

Day 1

- Motivation, basic concepts and use cases for containers
- Comparison of development environments with and without containers
- Comparison of deployment scenarios with and without containers
- Differentiation between bare metal, VMs and containers
- Container vs. Docker
- Container-Lifecycle, bestehende Container starten/stoppen
- Container-Images, Tags, Registry
- · Volumes, volume types and persistence
- · Networking and internal/external accessibility

Day 2

- · Mapping service architectures with Docker Compose
- Designing container-based applications
- Create your own images (Dockerfile)
- Baseimage, Layer und Caching, Multistage, Dockerfile Best Practices
- Basics of the Docker architecture (Docker CLI, Docker Daemon)
- Security best practices with securing images, containers, the Docker host, the Docker daemon
- Docker under Windows with Linux/Windows container, WSL2, Docker Desktop

Day 3

- Advanced debugging of containers
- Docker Architektur mit containerd, runc, Kernel (Namespaces, cgroups)
- · Containers without Docker/alternative runtimes
- Rootless Docker
- Container-based development environments
- · CI/CD with and for containers
- Container orchestration (Kubernetes) outlook

About Fast Lane

Fast Lane is a global, award-winning specialist in technology and business training as well as consulting services for digital transformation. As the only global partner of the three cloud hyperscalers- Microsoft, AWS and Google- and partner of 30 other leading IT vendors, Fast Lane offers gualification solutions and professional services that can be scaled as needed. More than 4,000 experienced Fast Lane professionals train and advise customers in organizations of all sizes in 90 countries worldwide in the areas of cloud, artificial intelligence, cyber security, software development, wireless and mobility, modern workplace, as well as management and leadership skills, IT and project management.



VMware

Germany

Fast Lane Institute for Knowledge **Transfer GmbH** Tel. +49 40 25334610

info@flane.de / www.flane.de

Austria

ITLS GmbH (Partner of Fast Lane) Tel. +43 1 6000 8800

info@itls.at / www.itls.at

Switzerland

Fast Lane Institute for Knowledge Transfer (Switzerland) AG Tel. +41 44 8325080

info@flane.ch / www.flane.ch



Fast Lane Services

- ✓ High End Technology Training
- ✓ Business & Soft Skill Training
- ✓ Consulting Services
- ✓ Managed Training Services
- ✓ Digital Learning Solutions
- ✓ Content Development
- Remote Labs
- ✓ Talent Programs
- Event Management Services