

DOCKER

Linux containers are changing the way companies think about service development and deployment. Containers play a vital role in the modern data-center, and Docker is leading the way. This course covers all the core features of Docker including: container creation and management, interacting with Docker hub, using Dockerfile to create and manage custom images, advanced Docker networking (how to safely expose container services to the world, and link containers), the use of Docker volumes to manage persistent data, and DockerCompose to build multi-container applications.

Emphasis is placed on best practices and how to secure Docker installations and containers. The course culminates with comprehensive labs where students use Docker, Git, and a continuous integration server to automate the testing of containerized applications.

Structure:

50% theory 50% hands on lab exercises

Prerequisites:

Proficiency with the Linux CLI. A broad understanding of Linux system administration.

Course Outline:

CONTAINER TECHNOLOGY OVERVIEW

- ✓ Application Management Landscape
- ✓ Application Isolation
- ✓ Container Resource Control & Security
- ✓ Container Types
- ✓ Container Ecosystem

MANAGING CONTAINERS

- ✓ Installing Docker
- ✓ Docker Control Socket
- ✓ Creating a New Container
- ✓ Listing Containers
- ✓ Viewing Container Operational Details
- ✓ Running Commands in an Existing Container
- ✓ Interacting with a Running Container
- ✓ Stopping, Starting, and Removing Containers

MANAGING IMAGES

- ✓ Docker Images
- ✓ Listing and Removing Images
- ✓ Searching for Images
- ✓ Downloading Images
- ✓ Component Soft – Open your mind!
- ✓ Committing Changes
- ✓ Uploading Images
- ✓ Export/Import Images

CREATING IMAGES WITH DOCKERFILE

- ✓ Dockerfile
- ✓ Caching
- ✓ docker build
- ✓ Dockerfile Instructions
- ✓ ENV and WORKDIR
- ✓ Running Commands
- ✓ Getting Files into the Image
- ✓ Defining Container Executable
- ✓ Best Practices

DOCKER NETWORKING

- ✓ Overview
- ✓ Data-Link Layer Details
- ✓ Network Layer Details
- ✓ Hostnames and DNS
- ✓ Local Host <--> Container
- ✓ Container <--> Container (same node)
- ✓ Container <--> Container: Links
- ✓ Container <--> Container: Private Network
- ✓ Managing Private Networks
- ✓ Remote Host <--> Container
- ✓ Multi-host Networks with Overlay Driver

DOCKER VOLUMES

- ✓ Volume Concepts
- ✓ Creating and Using Volumes
- ✓ Managing Volumes (cont.)
- ✓ Changing Data in Volumes
- ✓ Removing Volumes
- ✓ Backing up Volumes
- ✓ SELinux Considerations
- ✓ Mapping Devices

DOCKER COMPOSE/SWARM

- ✓ Concepts
- ✓ Compose CLI
- ✓ Defining a Service Set
- ✓ Docker Swarm

CONTINUOUS INTEGRATION WITH GITLAB, GITLAB CI, AND DOCKER LAB TASKS

- ✓ GitLab and GitLab CI Setup
- ✓ Unit and Functional Tests