

CTR-816 Docker From Basics to Production

Overview

Course Duration: 40 Hours

About This Course

This course takes you from Docker basics all the way to running applications on Kubernetes in production. You'll work with a real microservices app, learning how to containerize it, manage multi-container setups, and deploy it to a Kubernetes cluster you build yourself on AWS. You'll cover Docker fundamentals, networking, security, CI/CD, and Kubernetes — ending with a fully deployed, publicly accessible application.

Audience Profile

- Developers
- Architects
- DevOps
- SRE
- IT Operators

The App

In this course, we'll work with a real AI-powered application consisting of two microservices:

- YoloAPI - A Python FastAPI backend that receives images and uses the YOLOv8 model to detect objects (like cars, people, dogs, etc.)
- YoloUI - A simple web frontend where users can upload images and see the detected objects

You'll start by running this app manually on a single EC2, then package it into Docker containers, and finally deploy it to a Kubernetes cluster that **you'll build yourself** on AWS EC2 instances using Kubeadm. .

Course Outline:

Docker Fundamentals

A short introduction to cloud computing in AWS. Then we'll run the YoloAPI app on a single instance, first as a simple regular Python process (venv, install dependencies, etc), then as a Docker container. You'll learn what containers are and why they're useful.

Intro to cloud computing in AWS

- Amazon Web Services introduction
- Regions and Availability zones
- AWS shared responsibility model
- Different service models - IaaS, PaaS, SaaS

Introduction to Docker and Containerization

- Linux-wise - what are actually containers?
- Docker architecture and components
- Overview of Docker's ecosystem and community (DockerHub)

Basic Docker Commands and Workflow

- Docker commands syntax and structure
- Running your first container
- Docker CLI commands for managing images and containers
- Inspecting containers and viewing logs

Docker Images and Containers

- Docker images - build, layers, caching

- Dockerfile best practices (multistage builds)

Managing Data in Docker with Volumes

- Persistent data and stateful applications
- Volumes, bind mounts, and tmpfs mounts
- Volume management commands and strategies

Docker in Development with Visual Studio Code

- Visual Studio Code Docker setup
- Remote - Containers extension
- Debugging and testing in containers
- Hands-on: Developing inside a container

Advanced Docker Usage and New Features

We'll provision a full multi-containers app (YoloAPI and YoloUI) as well designed Docker compose projects in a single instance (custom networks and volumes, security considerations). Finally, we'll build a simple CI/CD pipeline that deployed a new version of the container on every push.

Introduction to Docker Networking

- Container networking basics
- Network types in Docker (none, bridge, host, overlay)
- Port mapping and container linking
- Network troubleshooting and best practices

Docker Compose

- docker-compose.yml configurations
- Multi-container applications with Docker Compose
- Hands-on: Multi-container setup with networking
- Reproducible development environments with Docker Compose

Security in Docker

- Docker in rootless mode
- Vulnerability detection with Docker Scout

- Docker security best practices

CI/CD with Docker: Integration into DevOps Workflows

- CI/CD pipelines with GitHub Actions
- Managing and versioning container images

Introducing K8S

We'll introduce k8s foundations, architecture, control-plane and worker components. We'll manually provision a cluster using kubeadm on a few instances (multiple control-planes, multiple workers). Finally, we'll deploy our Yolo app in the cluster.

- Kubernetes Foundations
- K8S Architecture and design - control-plane and worker nodes components
- Labels, namespaces
- K8S Objects - Pod, replicaset, deployment, service, configmap, secrets

System reliability and data durability

We'll introduce advanced features in k8s cluster - data persistence and stateful applications, cluster monitoring. We'll persist the YoloAPI uploaded images data (using statefulsets), and deploy a monitoring stack to collect logs in cluster nodes (using daemonsets)

- K8S Volumes
- K8S StateLess and Stateful applications
- Daemonsets
- Microservices Design and best practices in the world of K8S - HA, autoscaling, rolling update, resource quota

Networking and advance Kubernetes

We'll expose our app to be publicly available using LB and real registered domain. In addition we'll talk and practice some advanced topics in cluster networking.

- Kubernetes Networking
- Pod to Pod
- Exposing Services - ClusterIP, NodePort, LoadBalancer, External
- IP Per Pod

- Inter Pod Communication
- Intra Pod Communication

Ingress Controllers

- Ingress overview
- L7 cloud LB

Prerequisites

This course is designed for individuals with a foundational knowledge of command-line interfaces and a good understanding of software development and deployment concepts.

We also recommend our Microservices Seminar as all aspects of the course are around the terminology and practices in MSA.

Familiarity with the basics of networking and cloud computing will enhance the learning experience.