Faysal Aziz

Containerization & Deployment

Harnessing Docker and Kubernetes in a Real-World Remote Environment

FA-0141 · 2 days

DevOps, Cloud & Infrastructure

Introduction

Duration - 2 days

In today's rapidly evolving tech landscape, containerization and orchestration have become crucial skills for developers, DevOps engineers, and IT professionals. Docker and Kubernetes stand at the forefront of this movement, enabling organizations to deploy, manage, and scale applications more efficiently than ever before.

This 2-day intensive course is designed to provide hands-on experience with Docker and Kubernetes in a realistic industrial setting. Using remote Ubuntu servers accessed through Visual Studio Code (VSCode) with the SSH plugin, participants will gain practical skills that mirror real-world industry environments.

This approach ensures that you are not just learning concepts, but are also able to apply them directly in a production-like setting.

Learning outcomes

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

- Understand the core concepts of containerization and the role of Docker in modern application deployment.
- Build, manage, and deploy Docker containers in a remote server environment.
- Utilize Kubernetes for orchestrating containerized applications, ensuring scalability, resilience, and efficient resource management.

- Configure and manage a Kubernetes cluster on remote Ubuntu servers.
- Integrate VSCode with SSH to streamline development and deployment workflows.
- Apply best practices in container security, networking, and storage in a Kubernetes environment.

Prerequisites

- Good knowledge of Linux command line.
- Familiarity with fundamental programming concepts.
- Understanding of basic networking concepts.
- Some experience with any programming or scripting language (e.g., Python, JavaScript, Bash).
- (Optional) Prior exposure to virtualization or cloud computing concepts.

Training outline

1. Introduction to Containerization

Understanding Containerization

1. Concept and need for containerization.
2. Virtual Machines vs. Containers: Key differences.
3. Benefits of using containers in application development and deployment.

Introduction to Docker

1. What is Docker? A brief history.
2. Docker architecture: Docker Engine, Docker Images, Docker Containers, Docker Hub.
3. Installation of Docker on Ubuntu remote servers.
4. Setting up VSCode with SSH for remote Docker management.

2. Working with Docker

Docker Images and Containers

1. Understanding Docker images: Layers, union filesystem.
2. Pulling images from Docker Hub.
3. Building custom Docker images using Dockerfile.
4. Managing Docker containers: Running, stopping, restarting, and removing containers.

Docker Networking

1. Introduction to Docker networking concepts.
2. Creating and managing Docker networks.

3. Connecting containers with different networking modes (bridge, host, overlay).

Docker Volumes and Storage

1. Understanding Docker volumes and bind mounts.
2. Managing data persistence in Docker containers.
3. Best practices for Docker storage management.

Docker Compose

1. Introduction to Docker Compose: Defining multi-container applications.
2. Writing and managing docker-compose.yml files.
3. Deploying and managing multi-container applications with Docker Compose.

3. Introduction to Kubernetes

Understanding Kubernetes

1. Kubernetes overview: Why Kubernetes for container orchestration?
2. Key components of Kubernetes: Pods, Nodes, Clusters, Services, Deployments.
3. Setting up a Kubernetes cluster on remote Ubuntu server.

Kubernetes Architecture

1. Detailed exploration of Kubernetes architecture.
2. Kubernetes Master and Worker nodes: API Server, Scheduler, Controller Manager, kubelet, kube-proxy.
3. Working with Kubernetes namespaces and contexts.

4. Deploying Applications with Kubernetes

Managing Kubernetes Pods and Deployments

1. Creating and managing Pods: YAML configuration files.
2. Deployments and ReplicaSets: Ensuring application availability and scalability.
3. Rolling updates and rollbacks with Kubernetes Deployments.

Kubernetes Networking

1. Understanding Kubernetes networking: Services, ClusterIP, NodePort, LoadBalancer.
2. Configuring Ingress controllers for external access to services.
3. Networking policies for security and traffic control.

Kubernetes Storage

1. Persistent Volumes and Persistent Volume Claims.
2. Dynamic provisioning of storage in Kubernetes.
3. Best practices for managing storage in Kubernetes environments.

This course, led by an instructor with over 30 years of industry experience, is designed to be hands-on and practical, ensuring that participants leave with skills they can immediately apply in their work environments. By using remote Ubuntu servers and VSCode with SSH, we simulate an industrial setting, making this learning experience as close to the real-world as possible.

© 2026 Faysal Aziz · Kuala Lumpur