Course: Practical Oriented Docker & Kubernetes

Course Objectives

In this course you will learn to automate the deployment, scaling, and management of applications, Orchestration, in the software industry. The course primarily focuses on containerization technology, its implementation and uses in various application environments.

Prerequisites

Basic Knowledge of Linux

Day 1: Installation and Configuration

Concepts:

- Docker Architecture
- Docker Components(Images, Containers, Volume, Dockerfile)
- Docker Repository & Git Repository
- Connecting Servers Docker with Git & Its Basics (Areas, Repo Commands)
- Setup of repository, storage driver selection, Docker engine installation
- Port Binding
- Default container commands and Docker CLI usage

Hands-On Labs:

- Installing & Configuration in Docker
- Attached and Detached Modes
- Removing Docker Containers
- Docker Container Exec
- Importance of IT Flags
- Overriding Default Container Commands
- Restart Policies in Docker
- Removing Docker Container Images
- Disk Usage Metrics for Docker Components
- Automatically Delete Containers on Exit

Exercises:

- Test restart policies and IT flag combinations
- Override entry commands and measure resource usage
- Use CLI to monitor and clean Docker disk space

Day 2: Dockerfile, Compose, Storage & Volume

Concepts:

- Writing efficient Dockerfiles: COPY vs ADD, EXPOSE, ENTRYPOINT, WORKDIR
- Image lifecycle management: tagging, commit, inspect, prune
- Docker Compose fundamentals for orchestrating multi-container apps
- Docker Volumes: types (bind, named, tmpfs), lifecycle, usage patterns
- Container storage: storage drivers, persistent storage strategies

Hands-On Labs:

- Author Dockerfile with ENV, ENTRYPOINT, WORKDIR and build images
- Tag, commit, inspect, and prune Docker images with CLI
- Define and run multi-service Docker Compose YAML file
- Create/manage named volumes and bind mounts
- Enable automatic volume removal and custom volume drivers

Exercises:

- Build and launch a full-stack app with Docker Compose
- Compare storage drivers with disk usage analysis
- Volume lifecycle simulation with container deletion

Day 3: Docker Networking Essentials

Concepts:

- Docker networking overview: bridge, host, none networks
- Port mapping and exposed ports in Docker
- Legacy container linking and communication strategies

Hands-On Labs:

- Create user-defined bridge networks
- Use none network and host network modes
- Expose/publish ports with CLI options
- Link legacy containers and verify connectivity

Exercises:

- Build a custom network for inter-container communication
- Test multiple network modes and simulate isolation scenarios

Day 4: Kubernetes Orchestration and Object Management

Concepts:

- Limitations of Docker Swarm and why Kubernetes is preferred
- Kubernetes Architecture and Core Objects (Pod, ReplicaSet, Deployment)
- Services (ClusterIP, NodePort, LoadBalancer), Ingress overview
- Probes (Liveness & Readiness), Labels/Selectors, Namespaces

Hands-On Labs:

- Install Minikube/kubectl and create your first Pod
- Deploy and manage ReplicaSet and Deployments
- Configure Services and test networking between Pods
- Implement Liveness and Readiness probes in YAML specs
- Use labels/selectors to organize and target Kubernetes objects

Exercises:

- Build a deployment with multiple Pods using probes and service exposure
- Simulate blue-green deployment using labels and selectors
- Explore Namespaces and isolate resources between environments

Day 5: Security & Advanced Storage

Concepts:

- Storage in Kubernetes: PV, PVC, storage classes
- Block vs object storage: real-world use cases
- Kubernetes RBAC and role-based access restrictions
- Network Policies: securing inter-pod communication
- Linux Capabilities in containers and Docker security context

Hands-On Labs:

- Define and mount Persistent Volumes (PV) and Claims (PVC)
- Configure RoleBindings and apply RBAC rules to restrict access
- Write and enforce Kubernetes Network Policies
- Set container security contexts and drop capabilities
- Use logging drivers and secure bind mounts

Exercises:

- Deploy app with PVC-backed persistent storage and test data retention
- Apply fine-grained RBAC roles and simulate user access scenarios
- Secure multi-container app using Network Policies and Linux capabilities