

# **Docker, Kubernetes and Openshift Orchestration (5 days)**

**By Dr. Vishwanath Rao**

## **What you'll learn**

- Learn the fundamental uses of container technology and the architectural principles of how to apply containers to technical challenges.
- Create containers with both packaged solutions and custom software.
- Use the Docker and Kubernetes CLI environments to deploy solutions in local (Docker) and cluster (Kubernetes) environments.
- Learn how stateful services – such as databases, streaming solutions , and other specialty applications – can be deployed in highly-available configurations with redundancy and service-discovery/load-balancing.
- Show how other components – can be used to integrate services within the Kubernetes platform.
- Explore how Docker and technologies such as Jenkins can be used to continuously test and build containers based on changes to source code.

## **Who should attend**

Developers, system architects, technical team leads, systems administrators, and DevOps professionals who want to understand and use Kubernetes in cloud environments.

## **Prerequisites**

Proficiency with the Linux Command Line Interface (CLI)

Broad understanding of Linux system administration: Linux file system, networking, and bash scripting.

Computer programming concepts and methodologies. Full code examples for the course are provided in the Python and Java programming languages.

Day 1

## **Containerization using Docker**

- Docker Basics

- Where Can I Run Docker?
- Installing Docker Container Engine
- Docker Toolbox
- What is Docker?
- Docker Architecture
- Docker Architecture Diagram
- Docker Images
- Docker Containers
- Docker Integration
- Docker Services
- Docker Application Container Public Repository
- Docker Run Command
- Starting, Inspecting, and Stopping Docker Containers
- Docker Volume
- Dockerfile
- Docker Compose
- Using Docker Compose
- Dissecting docker-compose.yml
- Specifying services
- Dependencies between containers
- Injecting Environment Variables
- Summary

## **Managing Docker State**

- State and Data in Docker
- Volumes
- More About Volumes
- Uses for Volumes
- Working With Volumes
- Create Volume
- Use Volumes with Containers
- Bind Mounts
- Using Bind Mounts
- tmpfs Mounts
- Storing Data in the Container
- Storage Drivers
- Remote Data Storage
- Networking
- The Default Bridge Network
- User-Defined Bridge Networks
- Docker Network Commands
- Creating a User-Defined Bridge Network
- Summary

## Day 2

### **Open Container Initiative and Container Runtime Interface**

- Open Container Initiative (OCI)
- Docker
- Docker Engine Architecture
- runC
- containerd
- containerd Benefits
- CRI-O
- CRI-O Components
- Kubernetes and CRI-O
- Using Container Runtimes with Minikube
- Docker Runtime and Kubernetes
- Putting Things Together
- Summary

### **Kubernetes Architecture**

- Kubernetes Basics
- What is Kubernetes?
- Container Orchestration
- Architecture Diagram
- Components
- Kubernetes Cluster
- Master Node
- Kube-Control-Manager
- Nodes
- Pod
- Using Pods to Group Containers
- Label
- Label Syntax
- Label Selector
- Annotation
- Persistent Storage
- Resource Quota
- Interacting with Kubernetes
- Summary

### **Working with Kubernetes**

- Installation

- Startup
- Kubernetes Tools
- kubectl Command Line Interface
- API Proxy
- Dashboard
- Kubernetes Component Hierarchy
- Deployments
- Deployment Commands
- Updating Deployments
- Network Considerations
- Services
- Namespaces
- Labels
- Annotations
- Other Useful Commands
- Summary

## Day 3

### **Kubernetes Workload**

- Kubernetes Workload
- Kubernetes Workload (contd.)
- Managing Workloads
- Imperative commands
- Imperative Object Configuration
- Declarative Object Configuration
- Configuration File Schema
- Understanding API Version
- Obtaining API Versions
- Obtaining API Versions (contd.)
- Stateless Applications
- Sample Deployment Manifest File
- Working with Deployments
- Stateful Applications
- Sample Stateful Manifest File
- Sample Stateful Manifest File (Contd.)
- Working with StatefulSet
- Jobs
- Sample Job Manifest File
- Sample Job Manifest File (Contd.)
- Working with Batch Job
- DaemonSets
- DaemonSets (contd.)
- Sample Daemon Manifest File
- Rolling Updates
- Rolling Updates (Contd.)

- Summary

Day 4

## **Scheduling and Node Management**

- Kubernetes Scheduler
- Kubernetes Scheduler Overview (contd.)
- Skip Kubernetes Scheduler
- Scheduling Process
- Scheduling Process - Predicates
- Scheduling Process - Priorities
- Scheduling Algorithm
- Kubernetes Scheduling Algorithm
- Scheduling Conflicts
- Controlling Scheduling
- Label Selectors
- Label Selectors (contd.)
- Label Selectors (Contd.)
- Node Affinity and Anti-affinity
- Node Affinity Example
- Node Antiaffinity Example
- Taints and Tolerations
- Taints and Tolerations (Contd.)
- Taints and Tolerations - Example
- Summary

## **Managing Persistent Storage**

- Storage Methods
- Container OS file system storage
- Docker Volumes
- Kubernetes Volumes
- K8S Volume Types
- Cloud Resource Types
- configMaps
- Creating configMaps from Literals
- Creating configMaps from files
- Using configMaps
- emptyDir
- Using an emptyDir Volume
- Other Volume Types
- Persistent Volumes
- Creating a Volume
- Persistent Volume Claim

- Persistent Volume
- Pod that uses Persistent Volume
- Secrets
- Creating Secrets from Files
- Creating Secrets from Literals
- Using Secrets
- Security Context
- Security Context Usage
- Summary
- 

Day 5

## **Openshift Orchestration**

Container Orchestration : OpenShift Part1

Control access to resources on OpenShift

Monitor and collect metrics on OpenShift

Deploy applications on OpenShift Platform Container using S

Manage storage on OpenShift

Container Orchestration : OpenShift Part2

Source to Image (S2I)

Statefull and stateless Application

Manage Openshift with DevOps

View and expose app

Services

Replication controller, Replica Set,

Deployments, Volumes, Secrets.

Network, API

Create an App, Deploy app

Auto scaling

Dashboard

monitoring

Deploying YAML Files