Architecting MicroServices with Kubernetes, Docker, and Continuous Integration (2 Days)

This two-day DevOps training class is loaded with practical real-world information. You will leave this course loaded with knowledge on the usage of this stack for Microservices. This class combines instructor-led and hands-on lab exercises.

TOPICS

- Microservices
- CI/CD
- Kubernetes
- Docker

WHAT YOU WILL LEARN

After completing this course, the student should be able to:

- Confidently use the stack outlined in the course.
- Understand the various key components.
- Apply the knowledge to migrate applications to microservice architected solutions on Docker, Kubernetes, and Jenkins with OpenShift
- Understand the various components in an OpenShift environment for CI/CD

AUDIENCE

This course will be particularly useful for architects, technology managers, and development leaders implementing or considering Microservices and DevOps for their organization including the following:

- Architects
- Software Reliability Engineers
- Engineers
- App Dev Managers
- Lead Application Developers

PREREQUISITES

A desire to learn how this Microservices toolchain can improve your organization effectiveness, build & release processes, application architecture & development, and business continuity for greenfield and application modernization

Course Outline

Chapter 1. Introduction to Kubernetes

- What is Kubernetes
- What is a Container
- Container – Uses
- Container – Pros
- Container – Cons
- Composition of a Container
- Control Groups
- Namespaces
- Union Filesystems
- Popular Containerization Software
- Microservices
- Microservices and Containers / Clusters
- Microservices and Orchestration
- Microservices and Infrastructure-as-Code
- Kubernetes Container Networking
- Kubernetes Networking Options
- Kubernetes Networking – Balanced Design

Chapter 2. Kubernetes – From the Firehose

- What is Kubernetes?
- Container Orchestration
- Kubernetes Basic Architecture
- Kubernetes Detailed Architecture
- Kubernetes Concepts
- Cluster and Namespace
- Node
- Master
- Pod
- Label
- Annotation
- Label Selector
- Replication Controller and Replica Set
- Service
- Storage Volume
- Secret
- Resource Quota
- Authentication and Authorization
- Routing
- Registry
- Using Docker Registry

Chapter 3. Docker Introduction

- What is Docker
- Where Can I Ran Docker?
- Docker and Containerization on Linux
- Linux Kernel Features: cgroups and namespaces
- The Docker-Linux Kernel Interfaces
- Docker Containers vs Traditional Virtualization
- Docker as Platform-as-a-Service
- Docker Integration
- Docker Services
- Docker Application Container Public Repository
- Competing Systems
- Docker Command-line
- Starting, Inspecting, and Stopping Docker Containers

Chapter 4. CI/CD with OpenShift, Jenkins, and Blue Ocean

- What is OpenShift
- OpenShift Online
- OpenShift Origin
- OpenShift Architecture
- OpenShift Origin Installation
- OpenShift CLI
- OpenShift CLI (Contd.)
- Jenkins Continuous Integration
- Jenkins Features
- Running Jenkins
- Downloading and Installing Jenkins
- Running Jenkins as a Stand-Alone Application
- Running Jenkins on an Application Server
- Installing Jenkins as a Windows Service
- Different types of Jenkins job
- Configuring Source Code Management(SCM)
- Working with Subversion
- Working with Subversion (cont'd)
- Working with Git
- Build Triggers
- Schedule Build Jobs
- Polling the SCM
- Maven Build Steps
- Jenkins / OpenShift Pipeline
- Jenkins / OpenShift Pipeline Output
- Installing Jenkins Plugins
- The Blue Ocean Plugin
- Blue Ocean Plugin Features
- New modern user experience
- Advanced Pipeline visualizations with built-in failure diagnosis

- Branch and Pull Request awareness
- Personalized View
- OpenShift Pipeline Output
- Creating OpenShift Blue Ocean Pipeline

Chapter 5. Operational Readiness

- What is Operational Readiness
- Telemetry
- End-to-end Requirements Traceability
- Log Strategy
- Monitoring Strategy
- Runbooks

Chapter 6. Application Modernization

- What is Application Modernization
- Typical App Modernization Projects
- Why Modernization?
- Goals for Application Modernization
- Modernization Process
- Modernization in a Nutshell
- Modernization in a Nutshell - Analyze
- Modernization in a Nutshell - Rationalize
- Modernization in a Nutshell - Modernize
- Modernization in a Nutshell – Supervise
- Twelve-factor Applications
- Twelve Factors, Microservices, and App Modernization
- 12-Factor Microservice Codebase
- 12-Factor Microservice Dependencies
- 12-Factor Microservice Config
- 12-Factor Microservice Backing Services
- 12-Factor Microservice Continuous Delivery
- 12-Factor Microservice Processes
- 12-Factor Microservice Data Isolation
- 12-Factor Microservice Concurrency
- 12-Factor Microservice Disposability
- 12-Factor Microservice Environment Parity
- 12-Factor Microservice Logs
- 12-Factor Microservice Admin Processes
- Monolithic revisited
- Monolithic vs. Microservices
- Maintaining State in App Modernization
- Cloud Service Fabric

Chapter 7. Security in Microservices

- Why Microservice Security?
- Security Testing in Microservices
- Security Topology
- Authorization and Authentication
- J2EE Security Refresh
- Role-based Access Control in a Nutshell
- Claim-based Access Control in a Nutshell
- Sharing Sessions
- Session Cookie
- JSON Web Token (JWT)
- Spring Security
- Summary

Lab Exercises

Lab 1. Getting Started with Docker
Lab 2. Getting Started with Kubernetes
Lab 3. Getting Started with OpenShift
Lab 4. Managing Secrets with OpenShift
Lab 5. CI/CD with Jenkins, Docker, and OpenShift
Lab 6. CI/CD with Jenkins, Blue Ocean, Docker, and OpenShift