

**Architecting Microservices with Kubernetes, Docker, and Continuous Integration**

**Course Number:** MSV-100
**Duration:** 2 days

**Overview**

This Architecting Microservices with Kubernetes, Docker, and Continuous Integration training teaches how to use this stack for Microservices and use the various components in an OpenShift environment for CI/CD.

**Prerequisites**

No prior experience is presumed.

**Materials**

All Microservices training students receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computer for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will be fine.

**Objectives**

* 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

**Outline**

* Introduction
* Docker
	+ What is Docker
	+ Where Can I Run Docker?
	+ Installing Docker Container Engine
	+ Docker Machine
	+ 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
	+ Docker Volume
	+ Dockerfile
	+ Docker Compose
	+ Using Docker Compose
	+ Dissecting docker-compose.yml
	+ Specifying services
	+ Dependencies between containers
	+ Injecting Environment Variables
* 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
* 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
* Getting Started with OpenShift
	+ What is OpenShift/OKD
	+ Differences between OpenShift and Kubernetes
	+ Where OpenShift Fits in the IT Landscape?
	+ OpenShift Releases
	+ OpenShift Architecture
	+ OpenShift - Infrastructure
	+ OpenShift - Nodes
	+ OpenShift - Pods
	+ OpenShift – Registry
	+ OpenShift - Service layer
	+ OpenShift Origin Installation
	+ Firewall Configuration
	+ OpenShift CLI
	+ OpenShift CLI (Contd.)
	+ OpenShift – Volumes
	+ OpenShift – Secrets
	+ OpenShift – Secrets (Contd.)
* CI/CD with OpenShift, Jenkins, and Blue Ocean
	+ 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
	+ Configuring Jenkins to Access OpenShift/Kubernetes
	+ 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
* Operational Readiness
	+ What is Operational Readiness
	+ Telemetry
	+ End-to-end Requirements Traceability
	+ Log Strategy
	+ Monitoring Strategy
	+ Runbooks
* Application Modernization
	+ Next Generation Methodologies, Approaches, Tools, and Applications
	+ 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
	+ What Can Be Done to Modernize Applications?
	+ So, How Can Microservices Help Me?
	+ The Data Exchange Interoperability Consideration
	+ Microservices in Their Purest Form: AWS Lambdas
	+ The Microservices Architecture Design Principles
	+ Decentralized Processing
	+ Crossing Process Boundary is Expensive!
	+ Managing Microservices
	+ Traditional Enterprise Application Architecture (Simplified)
	+ Monolithic revisited
	+ Monolithic vs. Microservices
	+ Microservices Architecture Example (Simplified)
	+ Maintaining State in App Modernization
	+ 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
	+ Design for Failure
	+ Fault Injection During System Testing
	+ Messaging Architectures – Messaging Models
	+ What is Kafka?
	+ Kafka Architecture
	+ Need for Kafka
* 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
* Conclusion