

**Microservices Development with Spring Boot and Docker**

**Course Number:** MSV-110WA  
**Duration:** 5 days

**Overview**

This Microservices Development training course teaches attendees how to use Spring Boot, Linux Containers, and Docker to deploy resilient and scalable microservices. Participants also learn DevOps basics and how to incorporate Git Flow, implement twelve-factor applications, and more.

**Prerequisites**

All attendees must have basic knowledge of Java and Git.

**Materials**

All Microservices training attendees receive comprehensive courseware covering all topics in the course.

**Software Needed on Each Student PC**

For in-person deliveries, attendees require computers with the Java Development Kit version 8 and an internet connection. Students also require a Java Integrated Development Environment like Eclipse or IntelliJ. We will provide full classroom setup instructions that will include seating in small groups, with supplies such as flipcharts, sticky notes, markers, and pens for the attendees and a projector and Internet connection for the instructor's laptop.

Online deliveries for this interactive training will use an online meeting platform (such as Zoom, WebEx, GoTo, or Teams) to have face-to-face contact online, including use of breakout rooms for group activities.

**Objectives**

* Understand DevOps practices and GitFlow
* Break up monoliths into microservices
* Create a build using Gradle
* Create RESTful services with Spring Boot
* Use databases and JPA in Spring Boot
* Incorporate security patterns and best practices in Spring Boot
* Deploy resilient and scalable services
* Understand traffic routing patterns
* Apply metrics and tracing of Microservices

**Outline**

* Introduction
* DevOps Fundamentals
  + Why DevOps?
  + What is DevOps?
  + Collaborative, Matrixed, and Cross-Functional Teams
  + Key Components of Successful DevOps Teams
  + DevOps-ification
  + DevOps Vocabulary
  + DevOps Goals
  + Not DevOps - Crush Buzzwords
  + Driving Business Outcomes with DevOps
  + Technology-Enabled Business
  + DevOps Key Enabler for Digital Transformation
  + Core Values and Mission
  + Communication
  + Collaboration
  + Value Stream Mapping
  + Behavioral Patterns for Success
  + DevOps Org Structures
  + DevOps Team - Separate
  + DevOps Merged Organization
  + DevOps Overlapped Organization
  + Organizational Structure Leadership
  + What Does Continuous Delivery Mean?
  + Deployment Pipelines
  + Your Organization is Doing CD if …
  + Pipelining for CD
  + Continuous Integration
  + CI Pipeline
  + CD & CI Methodologies
  + Key Tool Categories for CI/CD
* Introduction to Git Flow
  + Why Use an SCM Workflow?
  + What is Git Flow?
  + The Main Branch
  + How Git Flow works
  + Git Flow Extension
  + Initializing Git Flow
  + Features
  + Release
  + Hotfixes
  + Git Flow and Continuous Integration
  + Trunk-based Development
  + GitHub Flow
* Breaking Up Monoliths: Pros and Cons
  + Traditional Monolithic Applications and Their Place
  + Disadvantages of Monoliths
  + Developer's Woes
  + Architecture Modernization
  + Architecture Modernization Challenges
  + Microservices Architecture is Not a Silver Bullet!
  + What May Help?
* Twelve-factor Applications
  + Twelve Factors, Microservices, and App Modernization
  + Categorizing the 12 Factors
  + 12-Factor Microservice
  + Kubernetes and the Twelve Factors
* Microservice Development
  + What are Microservices?
  + Microservices vs. Classic SOA
  + Principles of Microservices Architecture Design
  + Domain-Driven Design
  + Microservices and Domain-Driven Design
  + Designing for failure
  + Microservices Architecture: Pros and Cons
  + Docker and Microservices
  + Microservice Deployment with Docker: Workflow
  + Writing Dockerfile
  + Kubernetes
  + What is OpenShift
  + OpenShift Architecture
  + Microservices and Various Applications
  + Web Applications?
  + Single Page Applications
  + Traditional Enterprise Application Architecture
  + Sample Microservices Architecture
  + Serverless & Event-driven Microservice:AWS Lambda
* gRPC
  + What is gRPC?
  + Protocol Buffers
  + REST vs. gRPC
  + Protobuf vs. JSON
  + HTTP/2 vs. HTTP 1.1
  + Messages vs. Resources and Verbs
  + Streaming vs. Request-Response
  + Strong Typing vs. Serialization
  + Web Browser Support
  + REST vs. gRPC – In a Nutshell
* Introduction to Spring Boot
  + What is Spring Boot?
  + Spring Framework
  + How is Spring Boot Related to Spring Framework?
  + Spring Boot 2
  + Spring Boot Main Features
  + Spring Boot on the PaaS
  + Understanding Java Annotations
  + Spring MVC and REST Annotations
  + Example of Spring MVC-based RESTful Web Service
  + Spring Booting Your RESTful Web Service
  + Spring Boot Skeletal Application Example
  + Starters
  + Maven - The 'pom.xml' File
  + Spring Boot Maven Plugin
  + Gradle - The 'build.gradle' File
  + Spring Boot Gradle Plugin
  + HOWTO: Create a Spring Boot Application
  + Spring Initializr
* Overview of Spring Boot Database Integration
  + DAO Support in Spring
  + Spring Data Access Modules
  + Spring JDBC Module
  + Spring ORM Module
  + DataAccessException
  + @Repository Annotation
  + Using DataSources
  + DAO Templates
  + DAO Templates and Callbacks
  + ORM Tool Support in Spring
* Using Spring with JPA
  + Spring JPA
  + Benefits of Using Spring with ORM
  + Spring @Repository
  + Using JPA with Spring
  + Configure Spring Boot JPA EntityManagerFactory
  + Application JPA Code
  + Spring Boot Considerations
  + Spring Data JPA Repositories
  + Database Schema Migration
  + Database Schema Migration for CI/CD using Liquibase
  + How Liquibase Works?
  + Changelogs in Liquibase
  + Preconditions in Changelogs
  + Sample Empty Changelog
  + Sample Precondition in Changelog
  + Sample Changeset in Changelog
  + Running Liquibase
  + Liquibase Commands
* Spring REST Services
  + REpresentational State Transfer
  + Principles of RESTful Services
  + Understanding REST
  + @RestController Annotation
  + HTTP Request Method Mapping
  + Path Variables and Query Parameters
  + RequestBody and ResponseBody
  + JAX-RS vs. Spring
  + Java Clients Using RestTemplate
  + RestTemplate Methods
  + RestTemplate Example
  + Testing with Postman
* Scripting in Postman
  + Postman Scripts
  + Organization
  + Execution Order
  + Pre-Request Scripts
  + Post-Request Scripts
  + Console Output
  + Tests
  + Sample Assertions
  + Multiple Test Example
  + Test Results
  + Variables
  + Variable Scope
  + Defining Global Variables
  + Defining GlobalVariables
  + Using Environment Variables
  + Dynamic Variables
* Spring Security
  + Securing Web Applications with Spring Boot 2
  + Spring Security
  + Authentication and Authorization
  + Programmatic vs Declarative Security
  + Adding Spring Security to a Project
  + Spring Security Configuration
  + Spring Security Configuration Example
  + Authentication Manager
  + Using Database User Authentication
  + LDAP Authentication
  + What is Security Assertion Markup Language (SAML)?
  + What is a SAML Provider?
  + Spring SAML2.0 Web SSO Authentication
  + Setting Up an SSO Provider
  + Adding SAML Dependencies to a Project
  + SAML vs. OAuth2
  + OAuth2 Overview
  + Authenticating Against an OAuth2 API
  + OAuth2 using Spring Boot
  + JSON Web Tokens
  + How JWT Works
  + Adding JWT to HTTP Header
  + How The Server Makes Use of JWT Tokens
  + What are “Scopes”?
  + JWT with Spring Boot – Dependencies and Main Class
* Docker Introduction
  + 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 Integration and 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
  + runC
  + Running a Container using runC
* Introduction to Kubernetes
  + What is Kubernetes?
  + What is a Container?
  + Container – Uses, Pros, and Cons
  + Composition of a Container
  + Control Groups
  + Namespaces
  + Union Filesystems
  + Popular Containerization Software
  + Microservices and Containers / Clusters
  + Microservices and Orchestration
  + Microservices and Infrastructure-as-Code
  + Kubernetes Container Networking
  + Kubernetes Networking Options
  + Kubernetes Networking – Balanced Design
* Traffic Routing Patterns
  + Edge Proxy Server
  + Request Handling
  + Filters
  + Filter Architecture
  + API Gateway for Routing Requests
  + Rate Limiting
  + Configuring Rate Limiting in NGINX
  + Circuit Breaker
  + Design Principles
  + Cascading Failures
  + Bulkhead Pattern
  + Circuit Breaker Pattern
  + Thread Pooling
  + Request Caching
  + Request Collapsing
  + Fail-Fast
  + Fallback
  + Circuit Breaker Solutions
  + Load Balancing in Microservices
  + Server-side load balance
  + Client-side Load Balance
  + Architecture
  + Service Mesh
  + Service Mesh Solutions
  + Content Delivery Network (CDN)
  + How does a CDN Work?
  + Benefits of using a CDN
  + CDN Solutions
* Conclusion