**Microservices Development**

**With Spring Boot and Spring Cloud and Docker**

**DURATION:** 5 Days

Rapid advancements in cloud-based software delivery and virtualization have caused many developers to question the wisdom of a “Big Server” approach to deploying web applications. Spring Boot is a technology stack that builds on the popular Spring Framework to allow deployment of Spring-based applications as stand-alone jar files that host their own web servers.  This approach works nicely with deployment automation and rapid scaling.

Cloud-based, highly-distributed applications face additional challenges in supporting a dynamic environment – especially around configuration, service discovery, service resilience, and monitoring. Spring Cloud embraces and extends the popular suite of open source cloud tools published by Netflix (Eureka, Hystrix, Ribbon, etc.).

This Spring Microservices training introduces Spring Boot, Spring Cloud, and the Netflix OSS suite as a way to deploy highly resilient and scalable RESTful services and web applications.

**SPRING MICROSERVICES TRAINING OBJECTIVES**

All students will learn how to:

* Use Spring Boot to build standalone web applications and RESTful services
* Implement asynchronous messaging with JMS
* Build Microservices
* Use Netflix OSS to implement patterns for service discovery, load balancing, fault tolerance, and other key concerns for scalable distributed systems
* Filter requests to your microservices using Zuul

**SPRING MICROSERVICES TRAINING PREREQUISITES**

All attendees should be experienced Java SE developers and have familiarity with Core Spring.

**SOFTWARE NEEDED FOR EACH PC:**

* 64-bit Windows, Mac, or Linux environment with at least 8GB Free RAM
* Java 1.8 or later
* Spring Tool Suite (STS) Latest Version
* Related lab files that the instructor provides

**SPRING MICROSERVICES TRAINING OUTLINE**

Day 1:

* Introduction
* Microservices Fundamentals
  + How did we get here
  + What’s in a name
  + What is a Microservice Architecture
  + What is a Microservice
  + Defining Microservices in term of their Principles, Benefits and Challenges
  + Understand and explore the forces (stressors) of change on software
  + Shift in Thought Process: Monolith to Microservices
  + Benefits of Microservices
  + Downsides of Microservices
  + Use Cases for Microservices
* Spring MVC Primer (30 Mins)
  + Spring MVC
  + Spring Web Modules
  + Spring MVC Components
  + The DispatcherServlet
  + Spring MVC Annotated Controllers
  + @RequestMapping
  + Controller Handler Method Parameters
  + Controller Handler Method Return Types
* Introduction to Spring Boot
  + What is Spring Boot?
  + Spring Boot Main Features
  + Using Spring CLI
  + Example of Spring MVC-based RESTful Web Service
  + Spring Booting Basic RESTful Web Service
  + Spring Boot Skeletal Application Example
  + Externalized Configuration
  + Logging

Day2:

* Spring REST Services
  + What is REST?
  + Restful Controllers
  + Content Negotiation
  + JSON and XML Data Exchange
  + Restful Clients using RestTemplate
  + JPA Repositories with Hibernate Provider
  + Spring Data Rest, Exposing Endpoints
* Bilding MicroServices Using Spring Boot
  + Service Discovery in Microservices Architecture
  + Event-Driven Data Management for Microservices
  + Designing and Implementing Microservice Failover and Auto-Recovery Strategies using the Circuit Breaker Pattern

Day3:

* Introduction to Spring Cloud
  + Cloud Computing
  + What are the modules of Spring Cloud?
  + Spring Cloud
  + Service and Client Discovery
  + Netflix OSS
* Netflix OSS
  + Service Discovery
  + Eureka Servers and Clients
  + Load Balancing with Ribbon using Eureka

Day 4:

* Circuit Breakers
  + When services fail
  + Circuit Breakers - Hystrix
* Filtering And Routing with Zuul
  + Filtering requests and Zuul
  + Feign clients
* Logging and Monitoring Microservices (Theory)
  + Logging Solutions for Microservices
  + Monitoring Microservices With Turbine and Hystrix

Day 5:

* Micro Services Deployment
  + Introduction to Docker
  + Docker Image
  + Docker Container
  + Docker Commands
  + Running Docker Images
  + Docker Port Mappings
  + Docker Hub
  + Download Docker Images from Docker Hub
  + Build a docker image for a Spring boot Based Microservice
  + Upload an Image to docker hub
  + Running the built Docker Image
  + Scaling your app with Docker Containers
* Q&A
* Conclusion

**Note**:

1. The courseware and lab documents will be provided by the instructor
2. The instructor will provide a pre-built VMware/Virtual Box Image for Docker Lab
3. Please make sure to help the instructor to transfer the required files and images to every trainee’s computer.