MICROSERVICES Using Spring Boot and Java

By Dr. Vishwanath Rao

Prerequisites

- Knowledge of Java
- Git

Duration

Five days

Chapter 1. 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
- Core Values - Culture
- Core Values - Automation
- Core Values - Measurement
- Core Values - Sharing

- Communication
- Collaboration
- Value Stream Mapping
- Behavioral Patterns for Success
-

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?
- In-Class Discussion
- Summary

Twelve-factor Applications

- Twelve-factor Applications
- Twelve Factors, Microservices, and App Modernization
- The Twelve Factors
- Categorizing the 12 Factors
- 12-Factor Microservice Codebase
- 12-Factor Microservice Dependencies
- 12-Factor Microservice Config
- 12-Factor Microservice Backing Services
- 12-Factor Microservice Build, Release, Run
- 12-Factor Microservice Processes
- 12-Factor Microservice Port Binding
- 12-Factor Microservice Concurrency
- 12-Factor Microservice Disposability
- 12-Factor Microservice Dev/Prod Parity

- 12-Factor Microservice Logs
- 12-Factor Microservice Admin Processes
- Kubernetes and the Twelve Factors - 1 Codebase
- Kubernetes and the Twelve Factors - 2 Dependencies
- Kubernetes and the Twelve Factors - 3 Config
- Kubernetes and the Twelve Factors - 4 Backing Services
- Kubernetes and the Twelve Factors - 5 Build, Release, Run
- Kubernetes and the Twelve Factors - 6 Processes
- Kubernetes and the Twelve Factors - 7 Port Binding
- Kubernetes and the Twelve Factors - 8 Concurrency
- Kubernetes and the Twelve Factors - 9 Disposability
- Kubernetes and the Twelve Factors - 10 Dev/Prod Parity
- Kubernetes and the Twelve Factors - 11 Logs
- Kubernetes and the Twelve Factors - 12 Admin Processes
- Summary

- **Microservice Development**
- What are Microservices?
- Microservices vs Classic SOA
- Principles of Microservices Architecture Design
- Domain-Driven Design
- Domain-Driven Design - Benefits
- Microservices and Domain-Driven Design
- Designing for failure
- Microservices Architecture – Pros
- Microservices Architecture – Cons
- Docker and Microservices
- Microservice Deployment with Docker – Workflow
- Writing Dockerfile
- Kubernetes
- What is OpenShift
- OpenShift Architecture

- Microservices and Various Applications
- Web Applications
- Web Applications – Reference Architecture
- Web Applications – When to use?
- Single Page Applications
- Single Page Applications – Benefits
- Traditional Enterprise Application Architecture
- Sample Microservices Architecture
- Serverless & Event-driven Microservice – AWS Lambda
- Summary

gRPC

- What is gRPC?
- Protocol Buffers
- REST vs. gRPC
- Protobuf vs. JSON
- HTTP/2 vs. HTTP 1.1
- HTTP/2 vs. HTTP 1.1 (Contd.)
- 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
- Summary

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
- Summary

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

Spring REST Services

- REpresentational State Transfer
- Principles of RESTful Services
- Understanding REST
- REST Example – Create
- REST Example – Retrieve
- REST Example – Update
- REST Example – Delete
- REST Example – Client Generated ID
- REST Example – JSON
- @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
- Summary

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
- OAuth2 Overview
- OAuth – Facebook Sample Flow
- OAuth Versions
- OAuth2 Components
- OAuth2 – End Points
- OAuth2 – Tokens
- OAuth – Grants
- Authenticating Against an OAuth2 API
- OAuth2 using Spring Boot – Dependencies
- OAuth2 using Spring Boot – application.yml
- OAuth2 using Spring Boot – Main Class
- OAuth2 using Spring Boot – Single Page Application Client
- JSON Web Tokens
- JSON Web Token Architecture
- How JWT Works
- JWT Header
- JWT Payload
- JWT Example Payload
- JWT Example Signature
- How JWT Tokens are Used
- Adding JWT to HTTP Header
- How The Server Makes Use of JWT Tokens
- What are “Scopes”?
- JWT with Spring Boot – Dependencies
- JWT with Spring Boot – Main Class
- Summary
- Microservices and Orchestration
- Microservices and Infrastructure-as-Code
- Kubernetes Container Networking
- Kubernetes Networking Options

- Kubernetes Networking – Balanced Design
- Summary

- Edge Proxy Server
- Request Handling
- Filters
- Filter Architecture
- API Gateway for Routing Requests
- API Gateway for Routing Requests (Contd.)
- API Gateway – Example
- Rate Limiting
- Rate Limiting – Business Cases
- Configuring Rate Limiting in NGINX
- Circuit Breaker
- Design Principles
- Design Principles (continued)
- 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 (Contd.)

- Service Mesh Solutions
- Content Delivery Network (CDN)
- How does a CDN Work?
- Benefits of using a CDN
- CDN Solutions
- Summary