

Course duration

- 5 days

Course Benefits

- 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 a containerized microservice environment for CI/CD

Course Outline

1. Microservice Development
 1. What are Microservices?
 2. Microservices vs Classic SOA
 3. Principles of Microservices Architecture Design
 4. Business Domain-Centric Design
 5. Designing for failure
 6. Microservices Architecture – Pros
 7. Microservices Architecture – Cons
 8. Docker and Microservices
 9. Microservice Deployment with Docker – Workflow
 10. Writing Dockerfile
 11. Kubernetes
 12. What is OpenShift
 13. OpenShift Architecture
 14. Microservices and Various Applications
 15. Web Applications
 16. Web Applications – Reference Architecture
 17. Web Applications – When to use?
 18. Rich Client Applications
 19. Rich Client Applications – Reference Architecture
 20. Rich Client Applications – When to use?
 21. Rich Internet Applications
 22. Rich Internet Applications – Reference Architecture
 23. Rich Internet Applications – When to use?
 24. Mobile Applications
 25. Mobile Applications – Reference Architecture
 26. Mobile Applications – When to use?

27. Service Applications
28. Service Applications – Reference Architecture
29. Service Applications – When to use?
30. Single Page Applications
31. Single Page Applications – Benefits
32. Traditional Enterprise Application Architecture
33. Sample Microservices Architecture
34. Serverless & Event-driven Microservice – AWS Lambda
35. Summary
2. REST Services
 1. Many Flavors of Services
 2. Understanding REST
 3. Principles of RESTful Services
 4. REST Example – Create
 5. REST Example – Retrieve
 6. REST Example – Update
 7. REST Example – Delete
 8. REST Example – Client Generated ID
 9. SOAP Equivalent Examples
 10. REST Example – JSON
 11. REST vs SOAP Communication
 12. More REST vs SOAP
 13. REST vs SOAP Summary
 14. RESTful Services Usage
 15. Additional Resources
 16. Summary
3. Advanced Objects and Functionality in JavaScript
 1. JavaScript Evolution
 2. Basic Objects
 3. Constructor Function
 4. More on the Constructor Function
 5. Object Properties
 6. Deleting a Property
 7. The instanceof Operator
 8. Object Properties
 9. Constructor and Instance Objects
 10. Constructor Level Properties
 11. Namespace
 12. Functions Are First-Class Objects
 13. Closures
 14. Closure Examples
 15. Private Variables with Closures
 16. Immediately Invoked Function Expression (IIFE)
 17. The Module Pattern
 18. Module Pattern Example
 19. Prototype
 20. Inheritance in JavaScript

21. The Prototype Chain
22. Traversing Prototype Property Hierarchy
23. Prototype Chain
24. Inheritance Using Prototype
25. Extending Inherited Behavior
26. Enhancing Constructors
27. Improving Constructor Performance
28. Inheritance with Object.create
29. The hasOwnProperty Method
30. Summary
4. React Overview
 1. What is React?
 2. What's in a Name?
 3. React Component Model
 4. What React Is Not
 5. What You Will Not Find in React
 6. Motivation for Creating React
 7. A React JavaScript Example
 8. One-Way Data Flow
 9. JSX
 10. A JSX Example
 11. The Virtual (Mock) DOM
 12. Only Sub-components that Actually Change are Re-Rendered
 13. React Libraries
 14. Summary
5. Programming with React API
 1. React Programming Options
 2. Components vs Elements
 3. Three Ways to Create a React UI Component
 4. React API On-Line Documentation
 5. Setting Up the Libraries
 6. The ReactDOM Object
 7. The ReactDOM Object (Cont'd)
 8. The React Object
 9. The React.createElement Method
 10. The ReactElement Object
 11. The ReactElement Structure
 12. The React.DOM Object
 13. The React.PropTypes Object
 14. The React.Children Object
 15. The propTypes Object
 16. Lifecycle Methods (Applied only to ES6 Classes)
 17. Summary
6. Basic Components and JSX
 1. What is JSX?
 2. JSX Transpilation to React Code Example
 3. Running the Transpiled Code

4. Babel
5. Babel JavaScript Library
6. Script Import Skeleton Code
7. Playing Around in CodePen
8. React Components and Properties (Props)
9. Ways to Create UI Components
10. Creating a Functional Component Example
11. Component Names Must Be Capitalized
12. Creating a UI Component with React.createClass()
13. The render Method Object
14. Creating a UI Component Using ES6 Class Notation
15. Using ES6 Classes with React
16. Which UI Component Creation Syntax Should I Use?
17. Components vs Elements
18. Elements Are Immutable
19. Properties
20. Property Naming Convention
21. Properties Default to 'True'
22. Spread Attributes (an ES6 Feature)
23. Expressions
24. Summary
7. Introduction to Node.js
 1. What Is Node.js?
 2. Application of Node.js
 3. Installing Node.js and NPM
 4. "Hello, Node World!"
 5. How It Works
 6. Built on JavaScript: Benefits
 7. Traditional Server-Side I/O Model
 8. Disadvantages of the Traditional Approach
 9. Event-Driven, Non-Blocking I/O
 10. Concurrency
 11. Using Node Package Manager (NPM)
 12. Express
 13. Microservices with Node.js
 14. The Express Package
 15. Installing and Using Express
 16. Defining Routing Rules in Express
 17. Route Path
 18. The Response Object
 19. A Simple Web Service with Express Example
 20. Composite Services
 21. Example - Call an API Using a Promise
 22. Using the callApi() Function
 23. Summary
8. Extending React
 1. The Need to Extend React

2. Redux
3. Redux Design Ideas
4. React Router
5. React Router Code Examples
6. Issues With Manual Module Management
7. Webpack
8. Testing React Apps: ReactTestUtils
9. Testing React Apps: Jest
10. Testing with Jest and Enzyme
11. Summary
9. React Component Concepts
 1. Nesting JSX Elements
 2. Example of JSX Nesting
 3. Comments in JSX Code
 4. JSX Escapes Values
 5. Event Handling
 6. Event Handler Example
 7. Working with Lists of Items
 8. Keys in Lists
 9. Example List With Key
 10. Container vs. Presentational Components
 11. State
 12. Types of State Data
 13. State Hierarchy
 14. Lift State Up
 15. Props vs. State
 16. Pass Down a Function
 17. Immutability
 18. Immutability – Why?
 19. Virtual DOM and State
 20. Setting state
 21. Updating Input fields
 22. Passing Props to Components
 23. Passing Functions to Components
 24. Event Binding - DOs
 25. Event Binding – Don'ts
 26. Passing Parameters to Event Handlers
 27. App Development Workflow – 1/3
 28. App Development Workflow – 2/3
 29. App Development Workflow – 3/3
 30. Summary
10. Introduction to Spring Boot
 1. What is Spring Boot?
 2. Spring Boot Main Features
 3. Spring Boot on the PaaS
 4. Understanding Java Annotations
 5. Spring MVC Annotations

6. Example of Spring MVC-based RESTful Web Service
7. Spring Booting Your RESTful Web Service
8. Spring Boot Skeletal Application Example
9. Converting a Spring Boot Application to a WAR File
10. Externalized Configuration
11. Starters
12. The 'pom.xml' File
13. Spring Boot Maven Plugin
14. HOWTO: Create a Spring Boot Application
15. Summary
11. Spring MVC
 1. Spring MVC
 2. Spring Web Modules
 3. Spring MVC Components
 4. DispatcherServlet
 5. Template Engines
 6. Spring Boot MVC Example
 7. Spring MVC Mapping of Requests
 8. Advanced @RequestMapping
 9. Composed Request Mappings
 10. Spring MVC Annotation Controllers
 11. Controller Handler Method Parameters
 12. Controller Handler Method Return Types
 13. View Resolution
 14. Spring Boot Considerations
 15. Summary
12. Overview of Spring Database Integration
 1. DAO Support in Spring
 2. Spring Data Access Modules
 3. Spring JDBC Module
 4. Spring ORM Module
 5. DataAccessException
 6. @Repository Annotation
 7. Using DataSources
 8. DAO Templates
 9. DAO Templates and Callbacks
 10. ORM Tool Support in Spring
 11. Summary
13. Using Spring with JPA or Hibernate
 1. Spring JPA
 2. Benefits of Using Spring with ORM
 3. Spring @Repository
 4. Using JPA with Spring
 5. Configure Spring Boot JPA EntityManagerFactory
 6. Application JPA Code
 7. "Classic" Spring ORM Usage
 8. Spring JpaTemplate

9. Spring JpaCallback
10. JpaTemplate Convenience Features
11. Spring Boot Considerations
12. Spring Data JPA Repositories
13. Summary
14. Spring REST Services
 1. REST Services With Spring MVC
 2. Spring MVC Components
 3. Spring MVC @RequestMapping with REST
 4. Working With the Request Body and Response Body
 5. @RestController Annotation
 6. Implementing JAX-RS Services and Spring
 7. JAX-RS Annotations
 8. Spring Security
 9. Spring Security Options
 10. Spring Security Features
 11. Java Clients Using RestTemplate
 12. RestTemplate Methods
 13. Summary
15. Spring Security
 1. Securing Web Applications with Spring Security 3.0
 2. Spring Security 3.0
 3. Authentication and Authorization
 4. Programmatic v Declarative Security
 5. Getting Spring Security from Maven
 6. Spring Security Configuration
 7. Spring Security Configuration Example
 8. Authentication Manager
 9. Using Database User Authentication
 10. LDAP Authentication
 11. Summary
16. Spring JMS
 1. Spring JMS
 2. JmsTemplate
 3. Connection and Destination
 4. JmsTemplate Configuration
 5. Transaction Management
 6. Example Transaction Configuration
 7. Producer Example
 8. Consumer Example
 9. Converting Messages
 10. Message Listener Containers
 11. Message-Driven POJO's Async Receiver Example
 12. Message-Driven POJO's Async Receiver Configuration
 13. Spring Boot Considerations
 14. Summary
17. Introduction to Couchbase

1. What is Couchbase?
2. Key Components of Couchbase
3. Benefits of Couchbase
4. Basics of Data Modeling
5. Modeling One-to-many Relationship
6. Modeling Many-to-many
7. Doing a Query
8. About Query Index
9. Example MapReduce View
10. Summary
18. Introduction to Couchbase Programming Using Java
 1. Getting Started
 2. Opening a Connection
 3. Creating Index
 4. Doing a Query Using MapReduce View
 5. Doing an N1QL Query
 6. Retrieve a Document by ID
 7. Adding a Document
 8. Updating a Document
 9. Deleting a Document
 10. Summary
19. Introduction to KAFKA
 1. Messaging Architectures – What is Messaging?
 2. Messaging Architectures – Steps to Messaging
 3. Messaging Architectures – Messaging Models
 4. What is Kafka?
 5. What is Kafka? (Contd.)
 6. Kafka Overview
 7. Kafka Overview (Contd.)
 8. Need for Kafka
 9. Kafka Partitions
 10. Kafka Architecture
 11. Core concepts in Kafka
 12. Kafka Topic
 13. Kafka Producer
 14. Kafka Consumer
 15. Kafka Broker
 16. Kafka Cluster
 17. Why Kafka Cluster?
 18. Sample Multi-Broker Cluster
 19. Overview of ZooKeeper
 20. Kafka Cluster & ZooKeeper
 21. Who Uses Kafka?
 22. Summary
20. Using Apache Kafka
 1. Installing Apache Kafka
 2. Configuration Files

3. Starting Kafka
4. Using Kafka Command Line Client Tools
5. Setting up a Multi-Broker Cluster
6. Using Multi-Broker Cluster
7. Kafka Connect
8. Kafka Connect – Configuration Files
9. Using Kafka Connect to Import/Export Data
10. Creating a Spring Boot Producer
11. Adding Kafka dependency to pom.xml
12. Defining a Spring Boot Service to Send Message(s)
13. Defining a Spring Boot Controller
14. Testing the Spring Boot Producer
15. Creating a Nodejs Consumer
16. Summary

21. Introduction to Kubernetes

1. What is Kubernetes
2. What is a Container
3. Container – Uses
4. Container – Pros
5. Container – Cons
6. Composition of a Container
7. Control Groups
8. Namespaces
9. Union Filesystems
10. Popular Containerization Software
11. Microservices
12. Microservices and Containers / Clusters
13. Microservices and Orchestration
14. Microservices and Infrastructure-as-Code
15. Kubernetes Container Networking
16. Kubernetes Networking Options
17. Kubernetes Networking – Balanced Design
18. Summary

22. Kubernetes – From the Firehose

1. What is Kubernetes?
2. Container Orchestration
3. Kubernetes Basic Architecture
4. Kubernetes Detailed Architecture
5. Kubernetes Concepts
6. Cluster and Namespace
7. Node
8. Master
9. Pod
10. Label
11. Annotation
12. Label Selector
13. Replication Controller and Replica Set

14. Service
15. Storage Volume
16. Secret
17. Resource Quota
18. Authentication and Authorization
19. Routing
20. Registry
21. Using Docker Registry
22. Summary
23. Docker Introduction
 1. What is Docker
 2. Where Can I Run Docker?
 3. Docker and Containerization on Linux
 4. Linux Kernel Features: cgroups and namespaces
 5. The Docker-Linux Kernel Interfaces
 6. Docker Containers vs Traditional Virtualization
 7. Docker as Platform-as-a-Service
 8. Docker Integration
 9. Docker Services
 10. Docker Application Container Public Repository
 11. Competing Systems
 12. Docker Command-line
 13. Starting, Inspecting, and Stopping Docker Containers
 14. Summary
24. CI/CD with OpenShift, Jenkins, and Blue Ocean
 1. What is OpenShift
 2. OpenShift Online
 3. OpenShift Origin
 4. OpenShift Architecture
 5. OpenShift Origin Installation
 6. OpenShift CLI
 7. OpenShift CLI (Contd.)
 8. Jenkins Continuous Integration
 9. Jenkins Features
 10. Running Jenkins
 11. Downloading and Installing Jenkins
 12. Running Jenkins as a Stand-Alone Application
 13. Running Jenkins on an Application Server
 14. Installing Jenkins as a Windows Service
 15. Different types of Jenkins job
 16. Configuring Source Code Management(SCM)
 17. Working with Subversion
 18. Working with Subversion (cont'd)
 19. Working with Git
 20. Build Triggers
 21. Schedule Build Jobs
 22. Polling the SCM

- 23. Maven Build Steps
- 24. Jenkins / OpenShift Pipeline
- 25. Jenkins / OpenShift Pipeline Output
- 26. Installing Jenkins Plugins
- 27. The Blue Ocean Plugin
- 28. Blue Ocean Plugin Features
- 29. New modern user experience
- 30. Advanced Pipeline visualizations with built-in failure diagnosis
- 31. Branch and Pull Request awareness
- 32. Personalized View
- 33. OpenShift Pipeline Output
- 34. Creating OpenShift Blue Ocean Pipeline
- 35. Summary
- 25. Operational Readiness
 - 1. What is Operational Readiness
 - 2. Telemetry
 - 3. End-to-end Requirements Traceability
 - 4. Log Strategy
 - 5. Monitoring Strategy
 - 6. Runbooks
 - 7. Summary
- 26. Application Modernization
 - 1. What is Application Modernization
 - 2. Typical App Modernization Projects
 - 3. Why Modernization?
 - 4. Goals for Application Modernization
 - 5. Modernization Process
 - 6. Modernization in a Nutshell
 - 7. Modernization in a Nutshell - Analyze
 - 8. Modernization in a Nutshell - Rationalize
 - 9. Modernization in a Nutshell - Modernize
 - 10. Modernization in a Nutshell – Supervise
 - 11. Twelve-factor Applications
 - 12. Twelve Factors, Microservices, and App Modernization
 - 13. 12-Factor Microservice Codebase
 - 14. 12-Factor Microservice Dependencies
 - 15. 12-Factor Microservice Config
 - 16. 12-Factor Microservice Backing Services
 - 17. 12-Factor Microservice Continuous Delivery
 - 18. 12-Factor Microservice Processes
 - 19. 12-Factor Microservice Data Isolation
 - 20. 12-Factor Microservice Concurrency
 - 21. 12-Factor Microservice Disposability
 - 22. 12-Factor Microservice Environment Parity
 - 23. 12-Factor Microservice Logs
 - 24. 12-Factor Microservice Admin Processes
 - 25. Monolithic revisited

- 26. Monolithic vs. Microservices
- 27. Maintaining State in App Modernization
- 28. Cloud Service Fabric
- 29. Summary
- 27. Introduction to Feign
 - 1. What is Feign
 - 2. Feign - Annotations
 - 3. Creating a REST client with Feign
 - 4. Benefits of using Feign
 - 5. Feign – Default Beans
 - 6. FeignFeign – Simple Example
 - 7. Multiple Interfaces
 - 8. Ribbon
 - 9. Ribbon Load Balancing Architecture
 - 10. Using Ribbon
 - 11. Ribbon and Feign
 - 12. Hystrix
 - 13. Hystrix Dependency
 - 14. Using Hystrix
 - 15. Summary
- 28. Activiti Workflow
 - 1. Business Process Management
 - 2. Business Process Model and Notation
 - 3. BPMN (Contd.)
 - 4. BPMN – Elements
 - 5. BPMN 2.0
 - 6. What is Activiti
 - 7. Activiti – Components
 - 8. Activiti – Alternative Modeling GUI
 - 9. Activiti – Sample workflow in Activiti Process Modeler
 - 10. Using Spring Boot with Activiti
 - 11. Spring Boot with Activiti – Getting Started
 - 12. Spring Boot with Activiti – Simple Application
 - 13. Spring Boot with Activiti – Add BPMN 2.0 process definition
 - 14. Spring Boot with Activiti – Create a CommandLineRunner
 - 15. Spring Boot with Activiti – Create a Bean
 - 16. Spring Boot with Activiti – Start the Application
 - 17. Summary

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Microservices class:

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