Course duration

- 2 days

Course Benefits

- Gain a fundamental understanding of microservices and practical experience in implementing microservices using different technology stacks.

Course Outline

1. Microservices
    1. What is a "Microservice"?
    2. SOA - Microservices Relationship
    3. ESB - Microservices Relationship
    4. One Helpful Analogy
    5. The Driving Forces Behind Microservices
    6. How Can Microservices Help You?
    7. The Microservices Architecture
    8. Utility Microservices at AWS
    9. Microservices Inter-connectivity
    10. The Data Exchange Interoperability Consideration
    11. Managing Microservices
    12. Implementing Microservices
    13. Embedding Databases in Java
    14. Microservice-Oriented Application Frameworks and Platforms
    15. Summary
2. Microservices with Node.js
    1. What is Node.js?
    2. Node's Value Proposition
    3. Example of a Node.js App: a Simple Web Server
    4. Node.js Project Types
    5. Managing Large Applications
    6. Core Modules
    7. Why Node.js uses JavaScript?
    8. The Traditional Concurrency Support Model
    9. Disadvantages of the Traditional Approach
    10. Event-Driven, Non-Blocking I/O
    11. The Success Callback Function
    12. Using Node Package Manager (NPM)
    13. NPM Registry (Repository)
    14. NPM Enterprise

15. Package Life-Cycle Management
16. Local and Global Package Installation Options
17. Listing and Using Module Versions
18. The Express Package
19. Installing and Using Express
20. Defining Routing Rules in Express
21. Route Path
22. The Response Object
23. A Simple Web Service with Express Example
24. The MEAN Stack
25. Summary

3. 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. Famous RESTful Services
   15. Additional Resources
   16. Summary
4. Introduction to Angular
   1. What is Angular?
   2. Central Features of the Angular Framework
   3. Why Angular?
   4. Building Blocks of an Angular Application
   5. Basic Architecture of an Angular Application
   6. Angular vs. AngularJS
   7. Angular Semantic Versioning
   8. Installing and Using Angular
   9. A Basic Angular Application
   10. Anatomy of a Basic Application
   11. The Main Component File
   12. The Application Module File
   13. The index.html File
   14. The Bootstrap File
   15. Running the Application
   16. Building the Application
   17. Summary
5. Docker Introduction

    1. What is Docker
    2. Where Can I Ran 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
6. Apache ZooKeeper
    1. What is Apache ZooKeeper?
    2. Apache ZooKeeper Logo
    3. ZooKeeper Design
    4. ZooKeeper High Availability
    5. ZooKeeper - Client Interaction
    6. Leader Election
    7. The Big Picture
    8. znodes: A Closer Look
    9. Ephemeral znodes
    10. ZooKeeper Java API
    11. Watches
    12. Summary
7. Introduction to Spring Boot for Non-Java Developers
    1. What is Spring Boot?
    2. Spring Boot Main Features
    3. Spring Boot vs DropWizard
    4. Spring Boot on the PaaS
    5. Understanding Java Annotations
    6. Spring MVC Annotations
    7. Example of Spring MVC-based RESTful Web Service
    8. Spring Booting Your RESTful Web Service
    9. Spring Boot Skeletal Application Example
    10. Converting a Spring Boot Application to a WAR File
    11. Summary
8. Defining the Cloud
    1. A Bit of History
    2. Wikipedia Entry
    3. Cloud Computing at a Glance
    4. Gartner Research on Cloud
    5. Electrical Power Grid Service Analogy
    6. The NIST Perspective
    7. Five Characteristics

   8. On-demand Self-Service (NIST Characteristic)
   9. Broad Network Access (NIST Characteristic)
   10. Resource Pooling (NIST Characteristic)
   11. Rapid Elasticity (NIST Characteristic)
   12. Measured Service (NIST Characteristic)
   13. The Three Cloud Service Models (NIST)
   14. The Cloud Computing Spectrum: IaaS, PaaS and SaaS
   15. The Four Cloud Deployment Models (NIST)
   16. The NIST Cloud Definition Framework
   17. A Hybrid Cloud Diagram
   18. Cloud Deployment Model Dynamics
   19. Summary

9. Cloud Services
   1. Defining Cloud Services
   2. User-Cloud Interaction
   3. Cloud Service Characteristics
   4. The Typical Cloud Services
   5. Application Services
   6. Messaging Application Service
   7. Email Application Service
   8. Cache Application Service
   9. Specialized Application Services
   10. AWS Analytics Systems
   11. Google App Engine (GAE) MapReduce Service
   12. Use Cases for MapReduce Jobs
   13. Integration Platform as a Service (IPaaS)
   14. Storage Services
   15. Object Storage
   16. Archive Storage
   17. Relational Storage
   18. NoSQL Storage
   19. Some AWS Storage Services
   20. Data Warehouses in the Cloud
   21. Cloud Utility Services
   22. Scalability and HA of Your Applications in the Cloud
   23. The Auto-scaling Service
   24. Monitoring Services
   25. Configuring Instance Health Check in AWS
   26. Amazon Web Services Integration Diagram
   27. Google App Engine (GAE) Services Integration Diagram
   28. Microsoft Azure Services
   29. Comparing Cloud Service Stacks
   30. Summary
10. Introduction to Continuous Integration and Jenkins-CI
    1. Agile Development
    2. What is Continuous Integration
    3. Typical Setup for Continuous Integration

4. Jenkins Continuous Integration
5. Jenkins Features
6. Running Jenkins
7. 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:

- Foundational knowledge of programming and software design principles.