**Microservices TOC**

**Prerequisites**:

Participants should be comfortable with the below technologies to make the most of this workshop

1. Java 11
2. Git and GitHub
3. Spring Boot 2

**Duration**: 5 days

**Lab Setup**:

1. JDK 11
2. Docker
3. Apache Kafka
4. Gitlab account
5. CMDR (https://cmdr.net)
6. IntelliJ Idea / Spring Tool Suite
7. MySQL, MySQL Workbench
8. Postman client

**Course Outline:**

1. **Microservices – Introduction**

* Why and When to use
* SOA versus Microservices
* Benefits of using Microservices
* Challenges in using Microservice Architecture
* Breaking down a monolithic app to microservice app
* Case study of organizations who have moved to Microservice Architecture
* Frameworks used to build Microservices
* Design Patterns to be used when using a Microservice architecture

1. **Microservices and Cloud**

* Microservices and Spring Cloud
* Microservices and NETFLIX OSS
* The Twelve-Factor App
* CAP Theorem, Murphy’s law
* Cloud - IAAS, PAAS, SAAS, Cloud Computing Design Patterns - : Sharing, Scaling Elasticity, Reliability, Resiliency and Recovery, Monitoring, Provisioning and Administration Patterns

1. **Controlling your configuration using Spring Cloud configuration server**
   * GitHub integration
   * Securing sensitive information
   * Automatic updates using @RefreshScope
2. **Service Discovery using Eureka**
   * Eureka Discovery
   * Eureka Clustering with peer to peer to increase resiliency
3. **Client Resiliency Patterns using Reslience4J**

* Circuit Breaker using Resilience4j
* Fallback pattern
* Bulkhead pattern

1. **Service Gateway using API Gateway**

* Spring Cloud Gateway
* A/B Routes Configuration

1. **Securing Microservices**

* OpenID
* OAUTH 2 with Okta
  + Authorization Server
  + 4 Grant types
  + OAuth2RestTemplate

1. **Event sourcing using Saga**

* Event driven Architecture with Spring Cloud Stream and Apache Kafka / RabbitMQ
* Orchestrator pattern using Saga
* Choreography
* Compensating actions

1. **Log aggregation in Microservices**

* ELK Stack - Elasticsearch, Logstash, Kibana

1. **Testing and Deploying Microservices**

* Unit testing with Mock objects
* Integration testing using MockMVC
* Creating and using CI/CD pipeline
* Jenkins 2
* Deployment using docker compose
* Leveraging AWS Services like RDS, ECS for production deployment
* Dockererizing Spring Boot Applications
* Passing Environmental variables to Docker images

**Mode of delivery**:

The training will be based on a real-world use case and will be built from ground up in a iterative manner. All the sessions will be workshop based, where participants will be coding along with the instructor for the entire session. The final artifact will be deployed to AWS Cloud environment.

**Deliverables**:

1. Codebase developed during workshop will be shared to participants
2. Reference materials and Slides used for the workshop
3. Offline support for 30 days after training.