**Microservices TOC**

**Prerequisites**:

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

1. Java 8 +
2. Maven 3
3. Git and GitHub
4. Spring Boot 2
5. Docker
6. Kubernetes

**Duration**: 5 days

**Lab Setup**:

1. JDK 8
2. Maven 3
3. Docker
4. Apache Kafka
5. Gitlab account
6. CMDR (https://cmdr.net)
7. IntelliJ Idea / Spring Tool Suite
8. Kubernetes
9. 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**

* 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. **Docker** 
   * Containers
   * Images
   * Dockerfile
   * Multi stage builds
   * Docker compose
2. **Managing K8s Pods**
   * Pods
   * Replica Sets
3. **Centralized configurations**
   * ConfigMaps
   * Secrets
4. **Ingress Traffic using Ingress**
   * K8s Ingress controller
   * Path based and header based routing
   * Load balancing
   * Liveness Probe
   * Readiness Probe
5. **Istio Service Mesh**

* Setting up Istio Service Mesh
* Rate limit
* mTLS authentication
* Microservices security
* Telemetry

1. **Traffic Management**
   * Istio Gateway
   * Traffic splitting
   * Dark traffic
   * Sidecar proxy
   * Advance traffic splitting
2. **Service Resiliency**
   * Observability using Kiali
   * Circuit breaker
   * Distributed tracing
   * Istio resiliency
3. **Advance topics in Istio**
   * Sticky session
   * gRPC with Istio
   * Injecting Delays
   * Injecting Aborts
4. **Microservices Security**

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

1. **Inter Process Communication in Microservice Architecture**

* Event driven Architecture with Spring Cloud Stream and Apache Kafka / RabbitMQ
* Event sourcing
* Transactions using Sagas

1. **Performance Monitoring**

* ELK Stack - Elasticsearch, Logstash, Kibana
* NewRelic

1. **CI-CD**

* 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

1. **Deployment**

* Deployment using K8s
* Rolling update
* Blue-Green deployment
* Canary Deployment
* Setting up SSL

**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