Java Microservices

Program Duration: 64 hours (16 day – 4 hour each day)

Program Overview:
Understanding of Micro service with spring boot. Hand on experience with exercise Current software development trend with micro service.

Pre-requisite

- Basics Java 8 features
- Oop
- Java collection framework

Who should attend: Fresher and intermediate level java Developer

On Completion of this program :

- Understand Architecture of micro service
- Able to create micro service from scratch
- Debug to spring boot and understand different library
- Inter communication between micro service
- SQL and No SQL integration with spring boot
- Security in micro service.
- Application and uses in real world use case.
- Understanding of current tools uses to create Micro service.

Course Contents:
Day 1 & 2 : Prerequisite for micro service training

- Java essential for micro service
- Build tools maven
- Docker basic introduction
- Tools for code – STS, Postman
- Environment set up
- Hand-on lab on tools.

Day 3 & 4: Introduction to Micro services:

- What is MICROSERVICES
- Micro Services Architecture
- Microservice design pattern and its implementation
- Service Distribution
- Security
- REST API
- Uses
- Containerization
- Benefit
- Micro service in cloud world
- Tools and technology

Day 5 & 6: Spring Boot :

- Way to create Microservices
- Intro to Spring Boot
- Why Spring Boot
- DI with Spring Boot
- Spring REST
- Lab With Spring boot Start and Up

Day 7 & 8: REST API:

- Spring Boot REST Controller
- GET API
- POST/PUT API
- DELETE API
- Path Parameters
- Header Parameters
- Query Parameters

Lab to create Full controller Rest Api Microservice

Day 9 & 10 : Data Base With SpringBoot :

- SQL database
- Nosql database
- Spring Data Library
- MongoDB
- Database Configuration
- Lab With Spring boot and Database

Day 11 & 12 : Micro service Inter Communication:

- Design patterns : Service discovery, Load balancer, Circuit breaker, API Gateway • Inter communication between micro service - with security and handshaking mechanism examples
- Micro service Messaging
- ActiveMQ Producers
- ActiveMQ Consumers
- Eureka service Discovery load balancing
- Integration of microservices with API gateway
- Lab with message communication

Day 13: Micro service Testing:

- Automated unit testing
- Component test
- Integration test
- Spring Security(using jwt)

Day 14 : Docker

- Introduction to Docker
- Architecture
- Dockerhub and registry
- Dockerfile
- Docker compose
- Images and containers

Day 15: Deployment

- · Deployments of microservices over cloud environments (AWS or Azure, any one)

Day 16: Mini project (Please refer to attached project)

Project : Pension Management system
Language: java
Tools: Maven , Spring Tool Suite, JDK 11, postman, Docker Desktop Data Base: Mongo DB, MySQL
Messaging: ActiveMQ
Containerization: Docker Basic to run the Application