**Java Microservices Boot Camp**

**Intermediate+ Advanced Curriculum**

**(40 Hours)**

Virtual instructor led training sessions will be conducted.

Hands-on exercises/assignments to be completed daily outside of business hours.

**Curriculum Outline:**

**Prerequisite: Experience in Spring Boot 2.x**

Day 1 Microservices API Best Practices

* Banking POC discussion
* API design Best Practices
* Swagger Documentation (Hands on)
* Naming conventions
* Filtering Flexibility using Graphql, Squiggly, RSQL Visitor (Hands on)
* Paging and Sorting (Hands on)
* Versioning (Hands on)
* HTTP Status Codes (Hands on)
* Stability and Consistency (Hands on)
* Security using OAUth2 and JWT (Hands on)

Day 2 Microservices Design

• Microservice Architecture Decisions

• Microservice Design Patterns

o High Cohesion

o Loosely coupled

o Adapter pattern

o Chain Pattern

o Shared Resource Pattern

• Microservices Composition Pattern

• Microservice Security Principles/Techniques

• Ownership and Versioning

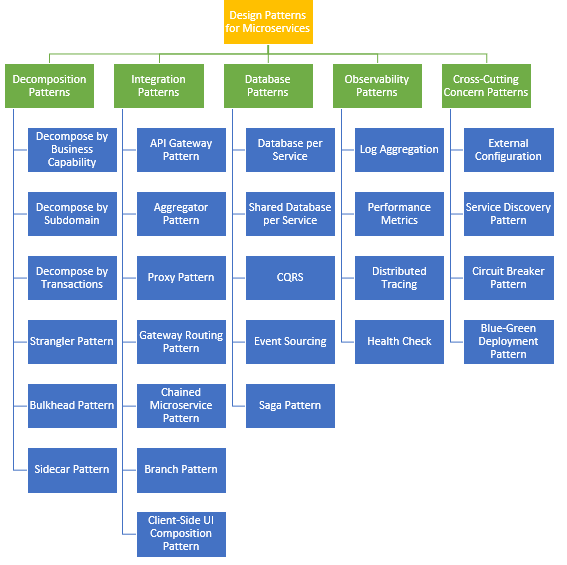
• Domain Centric

• Centralized Monitoring

Domain Driven Design

* Strategic Design Tools
* Tactical Design Tools
* Problem Space
* Solution Space
* Sub Domains
* Bounded Context
* Ubiquitous Language
* Context Map
* Model Driven Design
* Entities
* Repositories
* Services
* Aggregators
* Factory
* Value Objects
* Best Practices in Code Implementation

Design Patterns in Microservices



Day 3: Advanced Microservices Hands-on

**Moving Forward with Microservices**

**Migration to Microservices**

**Microservices Transaction Management**

**Microservices Database Migration**

**Microservices Reporting**

**Integrate with Application Performance Metrics tools such as**

**o AppDynamics**

**o Splunk and**

**o Other tools such as Kibana, ELK search.**

**Devops CI/CD**

Day 4 Continued:

**Docker Overview**

* + Understanding Docker
  + The differences between dedicated hosts, virtual machines, and Docker
  + Docker installation
  + The Docker command-line client
  + Docker and the container ecosystem

**Building Container Images**

* + Introducing the Dockerfile
  + Building container images

**Storing and Distributing Images**

* + Technical requirements
  + Docker Hub
  + Docker Registry
  + Docker Trusted Registry

**Managing Containers**

* + Technical requirements
  + Docker container commands
  + Docker networking and volumes

**Containerizing Microservices with Docker**

* + Microservices and containers
  + Deploying microservices in Docker
  + Creating a Docker container for Spring boot microservice

Week 4 Continued:

**Kubernetes Architecture**

* Master Components of Kubernetes
  + kube-apiserver
  + etcd key-value store
  + kube-scheduler
  + kube-controller-manager
  + cloud-controller-manager
* Node Components of Kubernetes
  + Docker
  + kubelet
  + kube-proxy
  + kubectl
* Deploying simple applications
* Show case load balancing, Self-healing, DNS allocation
* Labels and Selectors node selection
* Understanding Kubernetes namespaces
* Services
  + NodePort
  + ClusertIP
* Replication Sets
* Deployments
* Daemon Sets
* Jobs
* CronJobs
* Volumes
  + Persistent Volumes
  + Persistent Volume Claims
* ConfigMaps & Secrets
* Kubernetes Deployments Advanced
  + Scaling
  + Rolling Updates
  + Rollback
  + Probes
* Spring Boot Kubernetes [Hands on]
  + ConfigMap integration
  + Service Discovery
  + Spring Boot Probes integration with Actuator
  + Metrics with Prometheus, Grafana and spring cloud sleuth
  + Distributed Tracing with Jager