MICROSERVICES Deployments and Maintenance
By Dr. Vishwanath Rao

Prerequisites

- Knowledge on Java and Spring Boot
- Understanding on REST based services.
- Completion of Module 1 : MicroServices using Spring Boot training.

Objectives

- Able to see deployment strategies of Micro services.
- Able to use different types of components in Micro Services.
- Able to use Kafka for messaging and its relationship with Micro Services.
- Practical usage of Deployments strategy.
- Woriking with Micro services Logging and its best practices.

Day 1

Implementing MicroServices With SpringBoot

- Setting up a development environment
- Best Practices and Common Principles
- NovelHealthCare Project Overview
- Implementing super admin, admin, doctor and patient microservices

Controller Components Service Components Entity/Domain Components
Repository Components

Service discovery (consul/kubernetes service discovery)
Blue-green, canary, rolling deployments

The Kafka Architecture
The main components of Kafka Use cases for Kafka
The contents of Kafka's /bin directory How to start and stop Kafka
How to create new topics

Day 2

How to use Kafka command line tools to produce and consume messages
Kafka Streams
Relying on Kafka Topics for Storage Relying on Kafka for System State
Kafka Event-Driven Microservice Architecture

Rate Limiting
Rate Limiting – Business Cases Configuring Rate Limiting in NGINX Circuit Breaker

Design Principles
Design Principles (continued) Cascading Failures

Bulkhead Pattern Circuit Breaker Pattern Thread Pooling Request Caching

Request Collapsing Fail-Fast

Fallback
Circuit Breaker Solutions
Load Balancing in Microservices Server-side load balance Client-side Load Balance Architecture
Service Mesh
Service Mesh (Contd.) Service Mesh Solutions
Content Delivery Network (CDN) How does a CDN Work? Benefits of using a CDN
CDN Solutions

Day 3

JWT
Intrduction to JSON Web Token Authorization
Information Exchange JWT Structure
Header Payload Signature
Microservices communication using secured JWT

Distributed transaction
Isolate user actions for concurrent requests Transaction atomic
Two-phase commit (2pc) pattern Saga Pattern
Eventual Consistency and Compensation

Leading Practices for Microservice Logging Logging Challenges
Leading Practices
Correlate Requests with a Unique ID Include a Unique ID in the Response Send Logs to a Central Location Structure Your Log Data
Add Context to Every Record Examples of Content
Write Logs to Local Storage Collecting Logs with Fluentd