Spring Boot with Microservices

Duration 5 days

|  |
| --- |
| **Outline**  **Day-1**   * Introduction * Introduction to spring Core * Introduction to spring DI * Important Annotations   + What is Spring Boot?   + Spring starter Maven Dependencies   + Understanding @SpringBootApplication   + Example of Spring MVC-based RESTful Web Service   + Project Structure   + Externalized Configuration application.properties and YAML   + @ConfigurationPropetrties     - Type Safe Properties     - JSR303 Validation of Properties     - Spring Environment Object     - @Value vs @ConfigurationProperties   + Interceptors and Filters   + External Tomcat /web logic Deployment (war)   + Logging * Actuators   + Exposing Information about your services   + Customize Health and Info Endpoints   + Custom Metrics and Custom Actuator   + Prometheus Monitoring   Day-2   * Persistence with JPA Repositories   + JPA, EntityManagers, Entity Classes, Annotation mappings   + Spring JPA Data, CrudRepository, PagingAndSortingRepository   + Spring Data Rest, Exposing Databases as Endpoints   + Criteria query (rsql, squiggly and graphql)   + HATEOAS JSON   + @Projections and Excerpts     - Restrict the data sent back to the client   + Spring Unit Testing   Day-3   * Introduction to Microservices   + What are Microservices?   + Decentralized Governance, Scalability, Fault Tolerance   + Cloud Computing   + Spring Cloud   + Service and Client Discovery   + Netflix OSS * Core Microservice Patterns using Spring Cloud and Netflix OSS   + Where are my Services?     - Using Service Discovery,Eureka Servers and Clients     - External Configuration(cloud config and vault)   **Day-4**   * + Scale Services     - Load Balancing with Ribbon using Service Discovery     - A LoadBalanced RestTemplate     - Load Balancing with Ribbon without Service Discovery   + Circuit Breaker, when services fail     - Hystrix     - Callbacks   + Gateway/Edge Services – Providing an API     - Zuul services     - Filtering the Request and Response   + Create Feign Clients to your Services   + Microservice event sourcing   + Event Driven Services and Sagas   + Managing transactions,   + Two-phase commit (2pc) pattern, implementing Saga pattern     - Keeping Services synchronized with each other     - Creating Message Queues     - Project development and deployment on Spring   Day-5   * + - Deploying to Docker container using AWS     - Spring Security     - Unit testing and TDD approach in microservice using Mockito and Junit     - Deploying Spring microservice as lambda on AWS.  Best Practice.     - Architecture Overview ( Lambda , EC2/KMS/Env Variable , S3 , RDS , Cloud watch )     - Microservices design patterns |
|  |