**Topic 1**

***Introduction to Microservices***

1. The evolution of Microservices.
   1. Business demand as a catalyst for Microservices evolution
   2. Technology as a catalyst for the Microservices evolution
   3. What are microservices?
   4. Principles of microservices
   5. Characteristics of microservices.
   6. Microservices benefits.
      1. Polyglot, selective scalability, reduce technology debt, coexistence of versions, support event-driven architecture.
   7. Relationship with other architecture styles.
      1. Relations with SOA.
      2. Relations with Twelve-Factor apps.
2. Applying Microservices concepts
   1. Design decisions
   2. Establishing Microservice boundaries
   3. Designing Communication styles
   4. Orchestration of Microservices
   5. Multiple VM’s or One microservice per VM.
   6. Shared datasource?
   7. Logging and Monitoring.
3. Microservices capability model.
   1. Core capabilities
   2. Infrastructure capabilities
   3. Support capabilities
   4. Process & Governance capabilities

**Topic 2**

***Building Microservices with Spring Boot***

1. Introduction to Spring Boot
2. Your First Spring Boot Application with CLI, Spring Initializr and STS
3. Overview of opinionated architecture
4. Spring Boot Auto-Conﬁguration and Feature customizations
   1. Spring Boot lifecycle
   2. In-depth coverage of Start-up failures, Banner configuration, SpringApplicationBuilder, Events and different Runners.
5. Web Development and Data Access with Spring Boot
   1. In-depth coverage of different annotations in Spring.
   2. Deep dive on Rest API and Spring Boot Restful implementation
   3. API Design - RAML
   4. Embedded servers & Integration of external web-servers
   5. Spring Data in-depth (JDBC, JPA, MongoDB)
6. Testing with Spring Boot
   1. Testing RestAPI’s with Hamcrest & Mockito
7. Messaging with Spring Boot
   1. JMS & AMQP (ActiveMQ & RabbitMQ)
8. Security with Spring Boot
   1. Simple Security for Spring Boot
   2. Security Using the application.properties file
   3. In-Memory Security
   4. Security Using a Database
   5. Securing Resources
   6. Spring Boot with OAuth2
9. Spring Boot Actuator
   1. Actuator API’s in depth
   2. Use of GuageService / CounterService and HealthIndicator to develop our own metrics.

**Topic 3**

***Spring Cloud***

1. What is Spring Cloud
2. Components of Spring Cloud.
3. Spring Cloud and NetflixOSS
4. Details on Config, Eureka, Feign, Ribbon, Zuul and Streams.
   1. Explain the components with a solid usecase with code.
5. Logging and monitoring Microservices
   1. Log Streams, Log shippers, Log store, Log stream processors, Log dashboard.
   2. Custom logging implementation using **ELK**.
   3. Distributed tracing using Spring Cloud Sleuth.
6. Keep promises in mind. Spring Cloud Hystrix with Turbine.

**Topic 4**

***Containerization***

1. What are containers?
2. Components of Docker.
3. Docker concepts.
   1. Images, Container, Registry
4. Deploying Microservices with Docker.

**Topic 5**

***Microservices on the cloud***

1. Why cluster management is important/ what does cluster management do?
2. Managing Dockerized Microservices with Mesos and Marathon
3. The new mantra of leanIT – DevOps
4. Meeting the trio – microservices, DevOps, and cloud

**Topic 6**

***Openshift Origin***

1. What’s Kubernetis?
2. Pods, Labels, Replication Controllers, Services.
3. Working with Kubernetis with an example.

***Schedule:***

Topic 1 and 2 shall be covered in 3 days.

Topic 3, 4, 5 & 6 shall be covered in 2 days.

## **Prerequisites:**

* + - Thorough Java 8 knowledge
    - Thorough Spring framework knowledge (Spring core, Spring MVC, Spring data, Spring security). At least from Spring version 3.
    - Knowledge of Maven.
    - Understanding of JPA. JDBC, MongoDB, RabbitMQ, Redis etc.
    - Basic knowledge of RestAPI's (what Restful application is)
    - Knowledge of design patterns will help them understand the concepts better, especially Strategy, Decorator, Command and Template patterns.

***Softwares required:***

Windows 10 preferred

Java 8: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Eclipse oxygen with STS plugin: <https://www.eclipse.org/downloads/> (After installing eclipse, install the STS plugin from the market place)

MySQL server & workbench: <https://dev.mysql.com/downloads/file/?id=474496>, <https://dev.mysql.com/downloads/file/?id=474210>

MongoDB: <https://www.mongodb.com/dr/fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-3.6.2-signed.msi/download>

ActiveMQ: <http://www.apache.org/dyn/closer.cgi?filename=/activemq/5.15.2/apache-activemq-5.15.2-bin.zip&action=download>

RabbitMQ (need ErlangVM installed): <https://www.rabbitmq.com/install-windows.html>, <https://packages.erlang-solutions.com/erlang/>

Elasticsearch/ Logstash/ Kibana: <https://www.elastic.co/downloads/elasticsearch>, <https://www.elastic.co/downloads/logstash>, https://www.elastic.co/downloads/kibana

Docker for windows (need OracleVM VirtualBox): <https://download.docker.com/win/stable/DockerToolbox.exe>, <http://download.virtualbox.org/virtualbox/5.1.24/VirtualBox-5.1.24-117012-Win.exe>

Minishift (RedHat CDK), OC client: <https://github.com/minishift/minishift/releases/download/v1.12.0/minishift-1.12.0-windows-amd64.zip>, https://github.com/openshift/origin/releases/download/v3.7.1/openshift-origin-client-tools-v3.7.1-ab0f056-windows.zip

***Usecases used:***

1. A Journal entry application for Spring Boot concepts.
2. An Airline industry application for explaining Microservices concepts and used in explaining Spring Cloud concepts.
3. Online Table Reservation System (For evaluation)

An online table reservation system project that allows user to check for various restaurants and book available tables by taking the table payment and also provides additional optional things such as menu selection.

The project modules are listed as below:

 Restaurants have tables. And they have some menu to serve.

Booking service provides users the facility to search restaurants and tables and reserve a table.

Users can browse the different restaurants through the booking system, check the table varieties, reserve a table and optionally select a menu.

 Users may register themselves with the system for repeated visits.

 After this the total bill is calculated and user shall pay through the system.

 The user is notified about his booking by sending an email.

Registered users should have OAuth2 security.

Advantages:

• It saves user time in search of restaurants.

• The system generates online bill for requested days and even sends an email.

• User can pay online on the spot.

• It saves organization resources and expenses.

Applications:

• This system can be applied in restaurants.

• It can also be implemented in resorts.

Challenge

This application is to be developed as Microservices using Spring Boot with Angular4(optional) as a front-end and MySql/ MongoDb/ In-memory db as the back-end.

Developers are requested to develop an MSA for the above application. The challenge is to identify bounded context domain models and segregate as Microservices with own data source. Angular UI is optional and developers are free to have a minimal UI as this is to learn the usage of Spring Boot/cloud and its different modules and how to develop microservices.

No restrictions on thinking beyond. For eg: Will AMQP be of any use?