Comp1 – Develop Stock Market app with Microservice architecture

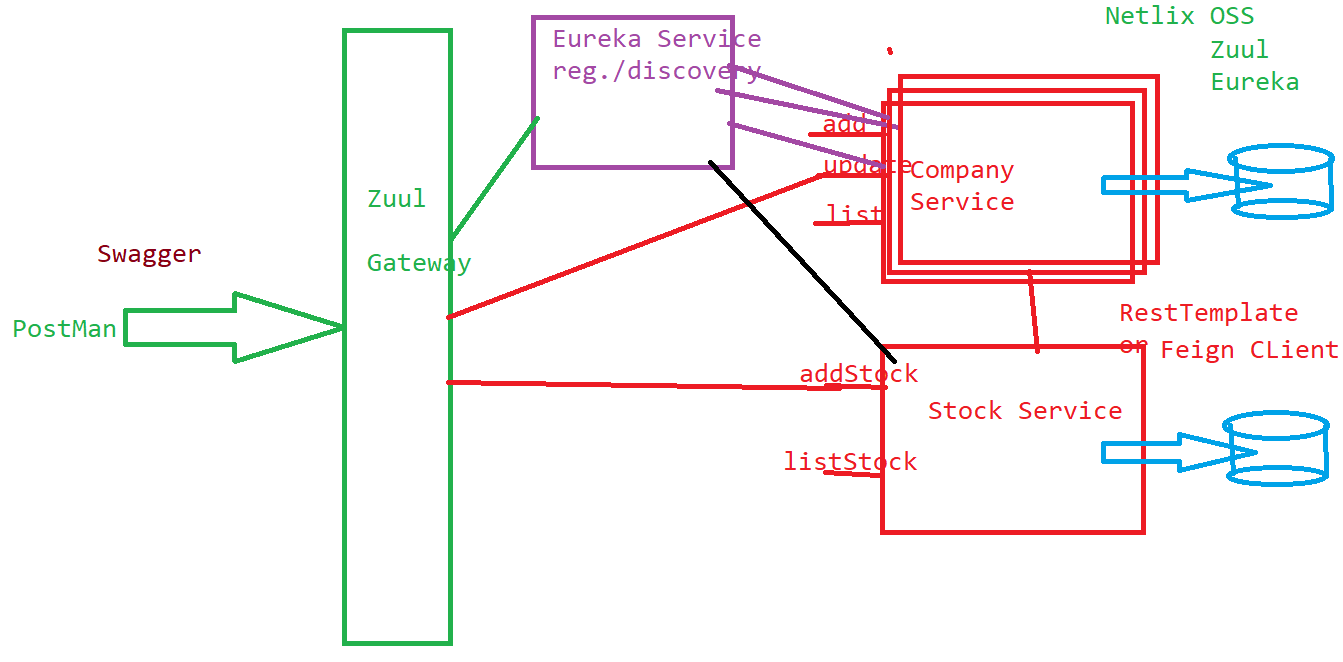
* Use Microservice Architecture(with Zuul & Eureka) with minimum two Services and separate Database(preferably one RDBMS and other Mongo DB)
* Swagger

**Panel Expectation:** End to End Scenario like adding Company, adding Stock price. Good confidence in Spring Boot, Zuul/spring cloud gateway, Eureka, Swagger

Below URL has sample Microservice Application

<https://drive.google.com/drive/folders/1txKYlZdwl6slWq3eZTq446BvEYInYjX8?usp=sharing>

Folder structure, good coding practices, Input data validations/Exception handling

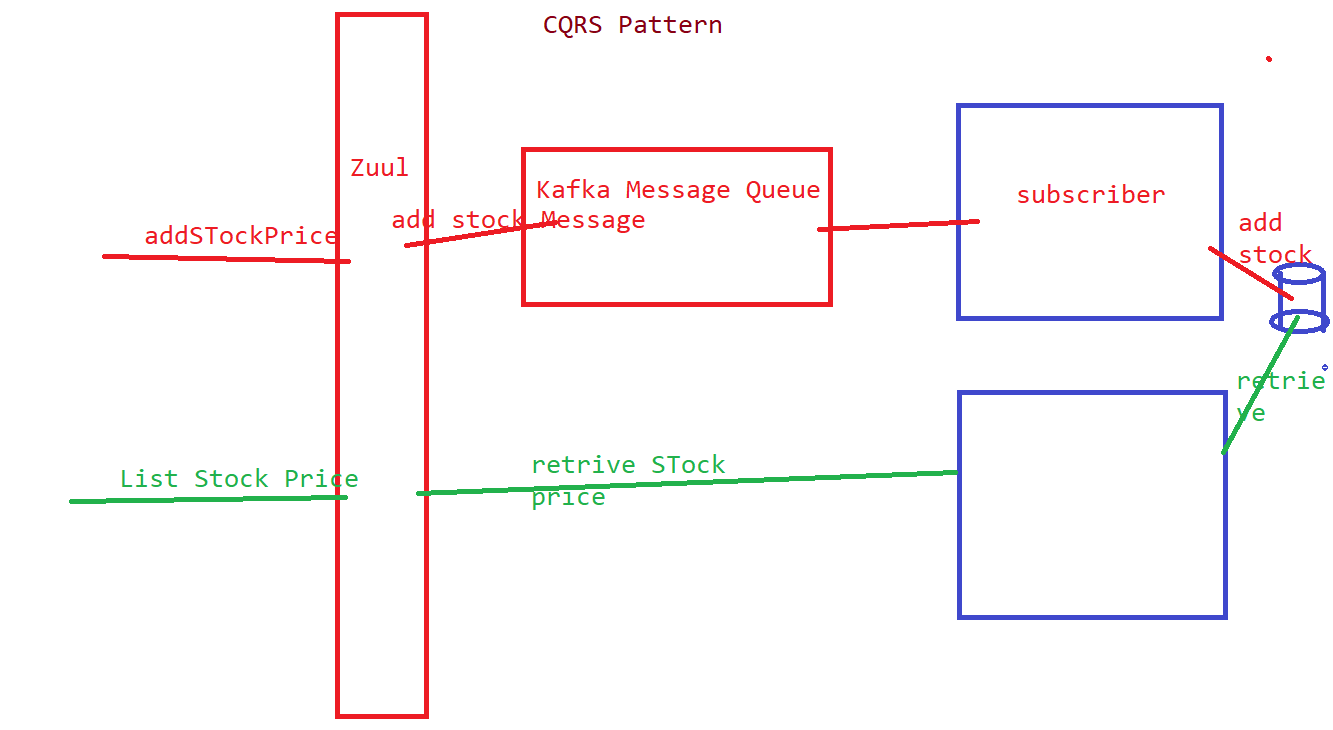
**Spring Boot concepts:** Explore Spring Boot concepts in similar lines as below

* All annotations used in Spring Boot applications and purpose of each
* Core Java 8 features(Fucntional interface, streams, Method reference, date/time, etc…)
* @SpringBootApplication, @RestController & @Controller diff, CRUD operations - Http Methods used. Http Headers/Http Status code, @Configuration, @CrossOrigin, @ControllerAdvice, Spring Core(@Bean, @Component, @ComponentScan, Scope), JPARepository/CrudRepository diff, @Query, MongoTemplate, MongoRepository, [etc... @Valid](mailto:etc...@Valid) for validations
* Advantages of Microservice architecture
* Purpose of Zuul/Spring Cloud Gateway, Eureka Service Registry/Discovery related annotations, configurations
* Does Microservices interact with each other, RestTemplate

Comp2 – Add Below, to Stock Market application

* Elastic Search – database to index & store logs in JSON format, LogStash, Kibana – UI Tool with dashboards to view statistics(ELK) for Log aggregation
* Kafka or Rabbit MQ or any Message Queue,
* CQRS(Command Query Responsibility Segregation) Pattern
* SonarQube
* Veracode – Security related vulnerabilities
* SSL Certificate
* Security(basic or jwt or oauth) with 2Factor Auth(optional),
* Actuator(Spring boot utility) with Prometheus and Grafana

Front End can be done either in Comp2 or 3

****

**Panel Expectation:** During Demo atleast one Scenario such as, adding stock price of a company from Front end, and showing new stock price in Database, is required.

ELK Stack – Elastic Search(NoSQL database), Log Stash(reads file converts into JSON and inserts into Elastic Search DB), Kibana(UI Tool to view graphically), Filebeat(optional).

Actuator dependency can be added to Spring Boot application(atleast one Microservice) and use Prometheus, and Grafana to visually monitor the Microservice(locally or on Cloud)

(If possible ELK, Prometheus & Grafana can be run as Docker Containers)

Angular/React(whichever Front end is opted)

**Angular Concepts:** Explore Angular Concepts in similar lines as below

Component(View), Service

Data binding – One way binding(interpolation), two way binding( [(ngModel)], event binding

Directives used-ngIf, ngFor, ngClass, ngStyle

Decorators(like annotations in Java) used-@Component, @Injectable, @NgModule, @Input, @Output, @ViewChild, @ViewChildren, etc…

How to display one component in another:

@Input – send data from Parent to Child component,

@Output- send data from Child to Parent component

Component LifeCycle: ngOnInit(), ngOnDestory, ngOnChange( ), ngAfterViewInit( ), etc…

Routing & AuthGuard

How to add Bootstrap or any styling at Component or Application Level

Observable(is used in Service to interact with Rest APIs)

angular.json, Angular CLI Commands

**React Concepts:** Explore React components in similar lines as below

Component

Component Life Cycle

JSX & JS differences

ES5, 6 features, arrow functions, de structuring assignments, let, etc…

Difference between State and Props of a Component

How to change the state of a component

How to embed one component in another

What is promise

What is arrow Function

Npm commands, package.json

How to interact with REST end points, which library you are using

Redux – state management

**Database Concepts:**

Characteristics of Mongo DB/Dynamo DB/RDBMS

Sample Queries

**Kafka or any Message Queue:**

Why Message Queue? How does it help in Scalability, loosely coupling Producer, Consumer Applications, etc…

Why we need Topic in Kafka?

Dependencies you added(in pom.xml) to interact with Kafka or any other Message Queue

**Spring Security:**

Security related configurations you have done in Project

Http Basic Authentication, jwt, oauth

Input data validations in FrontEnd/MidTier

Microservice architecture advantages

Above few concepts specified are only for reference you need to explore all commonly used features/concepts

Comp3 - Deploy Front End, Mid Tier(Spring Boot Microservices –atleast two) and Database on AWS or Azure, whichever you are approved with.

**Panel Expectation:** Demo on one end to end scenario like adding a Company or Stock price, and showing added data on Cloud database.

How Deployment is done on AWS or Azure. Good confidence in Java 8 features, Spring Boot, Angular/React, Database used. AWS/Azure services used. Coding Guidelines. Validations done.

**Deployment on AWS:**

Frontend on S3 bucket or ECS Fargate/EC2(after Dockerization)

Spring Boot app -> Dockerize -> push to Dockerhub or ECR -> deploy on ECS Fargate or Elastic Beanstalk(doesn't need Dockerization)

Docker registry can be Dockerhub or ECR

Database -> Dynamo db or RDS or Mongo DB running as Container or use Mongo Atlas cloud

Spring Boot REST, creation of docker image(using mvn command), upload to docker hub,

1. create ECS
2. Create new Task Definition with Fargate
3. Create Cluster using above Task
4. Run it

Functionality of One Service may be deployed on AWS Lambda – Serverless architecture

Use Load Balancer & API Gateway from AWS/Azure

2 Factor Authentication can be used either with Email or SMS based. AWS Cognito(User Pool) can be used for this.

Https using SSL Certificate

In Azure, Spring Boot Microservices and Front end application can be deployed on Azure’s App Services.

**References:**

**How to use JWT:**

<https://www.bezkoder.com/spring-boot-jwt-authentication/>

JWT Project source code: <https://github.com/bezkoder/spring-boot-spring-security-jwt-authentication>

**How to deploy Spring Boot application AWS ECS(Check Description for source code):**

https://www.youtube.com/watch?v=z7\_LdCVnCRU&t=610s

**(Dockerization:** [**https://www.youtube.com/watch?v=Tg2krHXHzBc&list=PLVz2XdJiJQxzMiFDnwxUDxmuZQU3igcBb&t=0s**](https://www.youtube.com/watch?v=Tg2krHXHzBc&list=PLVz2XdJiJQxzMiFDnwxUDxmuZQU3igcBb&t=0s) **)**

**Spring Boot REST with Dynamo DB with Elastic BeanStalk**

https://www.youtube.com/watch?v=E0-P978Yqdw

**SonarQube:**

[**https://medium.com/trendyol-tech/spring-boot-2-2-6-code-quality-with-sonarqube-8-2-community-70a76634bf75**](https://medium.com/trendyol-tech/spring-boot-2-2-6-code-quality-with-sonarqube-8-2-community-70a76634bf75)

[**https://www.youtube.com/watch?v=GZ2JWVUvDhw**](https://www.youtube.com/watch?v=GZ2JWVUvDhw)

**Actuator, Prometheus & Grafana:**

https://www.callicoder.com/spring-boot-actuator-metrics-monitoring-dashboard-prometheus-grafana/

<https://www.youtube.com/watch?v=2wr9njNdywk>

**Two Factor Authentication using AWS Cognito(with User Pool):**

<https://www.javacodegeeks.com/2019/04/integrate-spring-boot-application-cognito.html>

<https://www.youtube.com/watch?v=oFSU6rhFETk>

<https://www.youtube.com/watch?v=wjF8flLnLDk>

<https://www.youtube.com/watch?v=o7OHogUcRmI>

**How to deploy Angular app on S3:**

<https://www.youtube.com/watch?v=TLwMumqyGEQ>

**How to deploy ReactJS app on S3:**

https://www.youtube.com/watch?v=G2tgWGge-uQ

**Azure:**

**Deploy spring boot application onto Azure App Service**

https://www.youtube.com/watch?v=NNjgt\_8w9V4

https://www.youtube.com/watch?v=CPsI1KnVSOM

**Deploy Angular onto Azure**

https://www.youtube.com/watch?v=MAYIwTl66FU

**Kafka Message Queue(Install & Setup on Windows):**

https://www.youtube.com/watch?v=-fWznY5nBSg

**Setting up ELK Stack on Windows 10:**

https://www.youtube.com/watch?v=8iXZTS7f\_hY