Name:Shweta Gund

Theory QA Week 5

Q1.What does one mean by Load Balancing ? How is it implemented in Spring Cloud?

-> To use the Spring Cloud Load Balancer, we need to have a service registry up and running. A service registry makes it trivial to programmatically query for the location of a given service in a system. There are several popular implementations, including Apache Zookeeper, Netflix’s Eureka, Hashicorp Consul, and others.Spring Cloud Loadbalancer is a generic abstraction that can do the work that we used to do with Netflix’s Ribbon project.

Q2.In which business scenario to use Netflix Hystrix?

-> Netflix Hystrix is a design and implementation of circuit breaker pattern to isolate failure points and stop cascading them. It uses circuit breaker and fall back mechanisms to detect failures. This can be done with annotation @HystrixCommand where classes are annotated with @Component or @Sevice. When we use @HystrixCommand, Hystrix monitors failures till threshold point then subsequent calls are failed and redirected to fallback method.

Q3. What is Spring Cloud Gateway? What are its advantages over Netflix Zuul?

-> Spring Cloud Gateway is API Gateway implementation by Spring Cloud team on top of Spring reactive ecosystem. It provides a simple and effective way to route incoming requests to the appropriate destination using Gateway Handler Mapping.Zuul 1.x is not reactive. It is blocking. Now if you are moving to Reactive pattern to get better performance from your microservices and you are using Spring boot 2 with Reactor — you can use Zuul 2.0 which does have Reactive non blocking support with netty.

Q4. What is Spring Cloud Bus? Need for it?

-> Spring Cloud Bus links nodes of a distributed system with a lightweight message broker. This can then be used to broadcast state changes (e.g. configuration changes) or other management instructions. AMQP and Kafka broker implementations are included with the project.

Q5. What is Spring Cloud Data Flow? Need for it?

-> Spring Cloud Data Flow is a cloud-native toolkit for building real-time data pipelines and batch processes. Spring Cloud Data Flow is ready to be used for a range of data processing use cases like simple import/export, ETL processing, event streaming, and predictive analytics.

Q6. What is Docker? How to deploy Spring Boot Microservices to Docker?

-> Docker allows you containerize your microservices and simplify the delivery and management of those microservices.Containerization provides individual microservices with their own isolated workload environments, making them independently deployable and scalable. First, we need to create an Image file for our application. Docker images are an important component for working with the Docker engine. Docker provides a Docker Hub, which is a library and community for container images. We can use Docker Hub to get the most common images. Here we have used a few commands as per our needs. FROM - Must be the first non-comment instruction in the Dockerfile. This command creates a layer from the Docker image. In our cas, we have used java:8 which means this application will run on Java 8.EXPOSE - Exposing the port for the endpoint. In this example, we have configured 8080.ADD - This command helps to take a source and destination. Normally, the source is your local copy. The COPY command also does same thing, but there is a small difference between the COPY and ADD commands. ENTRYPOINT - It's similar to CMD, where our command/jar file will be executed. FROM - Must be the first non-comment instruction in the Dockerfile. This command creates layer from the Docker image. In our case we have used java:8, which means this application will run on Java 8.

Q7. How to deploy multiple microservices to docker?

-> we have to do the following steps manually-Create custom docker network named consumer-producer networkStart Container named producer using image employee-producer and the custom network consumer-producerStart Container named consumer using image employee-consumer and the custom network consumer-produce.The above steps can be automated using docker compose.Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration. Using docker-compose we will be creating the custom network named consumer-producer and then starting the containers employee-producer and employee consumer.

Q8. How to impliment feign client using springboot?

-> Create Microservice #1(Eureka Server)

Step #1: Create a Spring Boot Project

Step #2: Apply Annotation @EnableEurekaServer at the main class

Step #3: Modify application.properties file

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

Create Microservice #2(Producer Service)

Step #1: Create a Spring Boot Project

Step #2: Apply Annotation @EnableEurekaClient at the main class

Step #3: Modify application.properties file

Step #4: Create Model class as Book.java

Step #5: Create a RestContoller class as BookRestController.java

Create Microservice #3(Consumer Service)

Step #1: Create a Spring Boot Project

Step #2: Apply Annotation @EnableEurekaClient and @EnableFeignClients at the main class

Step #3: Modify application.properties file

Step #4: Create Model class as Book.java

Step #5: Create an interface as BookRestConsumer.java

Step #6: Create a RestController as StudentRestController.java

Q9. How to implement security for microservices?

-> JWT token based authentication flow:

1.Get the JWT based token from the authentication endpoint, eg /login.

2. Extract token from the authentication result.

3.Set the HTTP header as Authorization and value as Bearer jwt\_token.

4. Then send a request to access the protected resources.

5. If the requested resource is protected, Spring Security will use our custom Filter to validate the JWT token, and build an Authentication object and set it in Spring Security specific SecurityContextHolder to complete the authentication progress.If the JWT token is valid it will return the requested resource to client.

Q10. How to implement distributed logging for microservices?

-> Logging in microservices comes with other advantages that can be shipped with the architecture, it also comes with its own set of complexities

– the reason being a single request in this architecture could span across multiple services, and it might even travel back and forth. To trace

the end-to-end flow and identify the source of originating errors of a request through our systems, we need a logging and monitoring system in place.