@KLWKS_BOT THANOS

DEPARTMENT OF CSE
COURSE CODE: 23SDCS12A / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:______to______**

#LAB - 13 ➔ Implementing Spring Cloud Integration

Prerequisites:

Basic Idea on Spring Boot

Basic Idea on Spring Cloud

Exercise:

Now, due to the implementation of microservices in market, there is a requirement to have common values for the same variables available in different spring boot app. In this case, there should be similar 2 spring boot app which uses same variable and the value for the variable need to be picked form the github repo based on the "development" or "production" environment. Assign the picked value to the variable and print it to show the output.

❖ Watch The Video And Do In Eclipse Workspace

13 https://youtu.be/8uxqh4BhFRQ?si=NliciRYeE3k7GnKH

EX-SC-CLIENT

AppController.java

```
package com.klu;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class AppController {

   @Value("${msg:Not Connected to Server}")
   String msg;

   @GetMapping("/clientfun")
   public String fun1() {
      return msg;
   }
}
```

applications.properties

```
spring.application.name=EX-SC-CLIENT

server.port=8082

#spring.profiles.active=production
spring.config.import=configserver:http://localhost:8081
```

EX-SC-SERVER

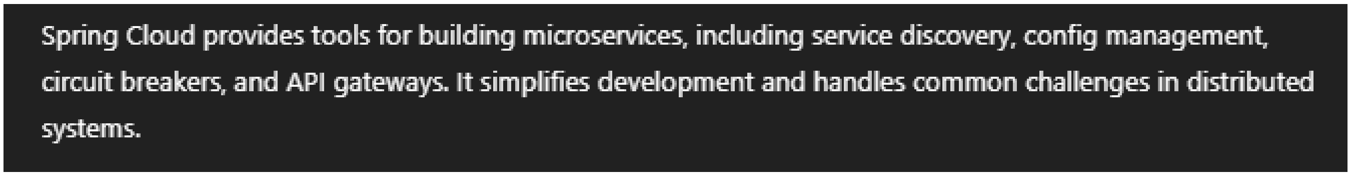
applications.properties

```
spring.application.name=EX-SC-SERVER
server.port=8081
spring.cloud.config.server.git.uri=https://github.com/balajee-rm/JFSD-CLOUD.git
spring.cloud.config.server.git.username=balajee.rm@gmail.com
spring.cloud.config.server.git.password=ghp_ZlDf37DsDLtjmkJxtKfdpZbFkj3Dvh22WV2B
spring.cloud.config.server.git.default-label=master
spring.cloud.config.server.git.clone-on-start=true

management.security.enabled=false
```

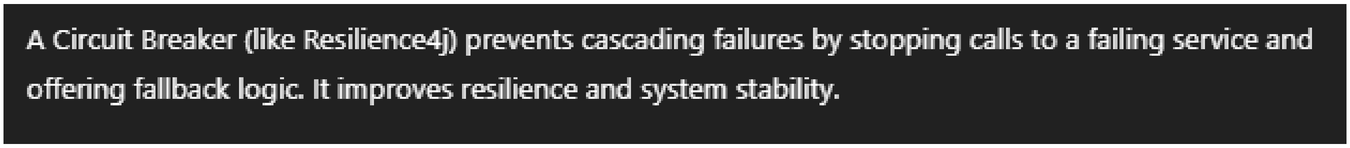
VIVA QUESTIONS:

1. What is Spring Cloud, and why is it useful in a microservices architecture?



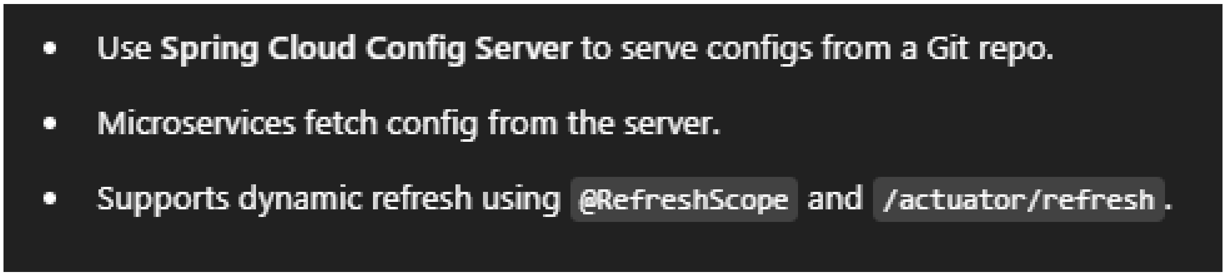
Spring Cloud provides tools for building microservices, including service discovery, config management, circuit breakers, and API gateways. It simplifies development and handles common challenges in distributed systems.

2. What is a Circuit Breaker in Spring Cloud, and how does it help improve system resilience?



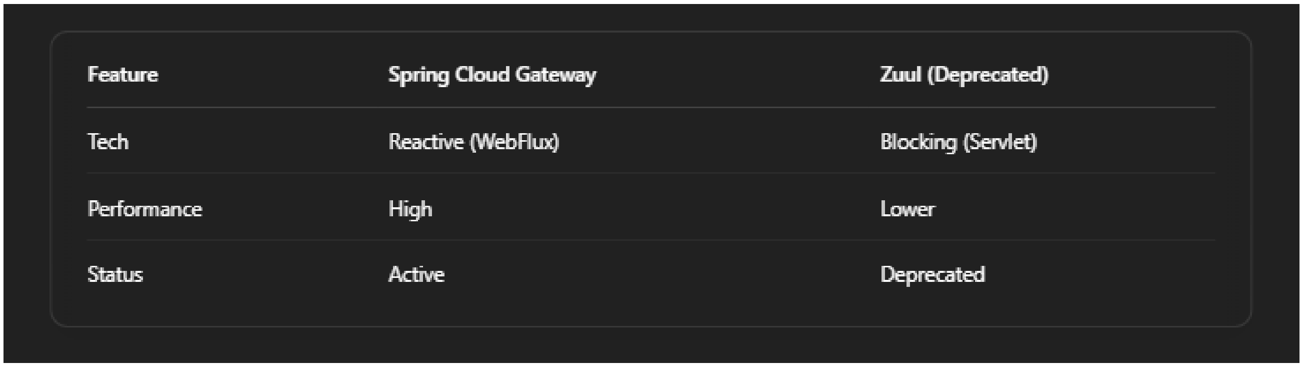
A Circuit Breaker (like Resilience4j) prevents cascading failures by stopping calls to a failing service and offering fallback logic. It improves resilience and system stability.

3. How do you implement centralized configuration management with Spring Cloud Config?



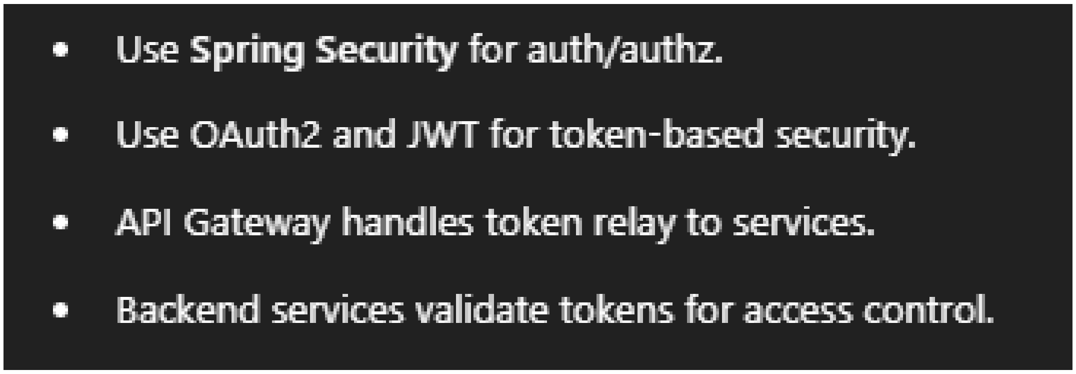
- Use **Spring Cloud Config Server** to serve configs from a Git repo.
- Microservices fetch config from the server.
- Supports dynamic refresh using `@RefreshScope` and `/actuator/refresh`.

4. What is Spring Cloud Gateway, and how does it differ from Zuul as an API Gateway?



Feature	Spring Cloud Gateway	Zuul (Deprecated)
Tech	Reactive (WebFlux)	Blocking (Servlet)
Performance	High	Lower
Status	Active	Deprecated

5. How can you secure microservices in Spring Cloud, and what role does Spring Security play?



- Use **Spring Security** for auth/authz.
- Use OAuth2 and JWT for token-based security.
- API Gateway handles token relay to services.
- Backend services validate tokens for access control.

(For Evaluator's use only)

Comment of the Evaluator (if Any)

Evaluator's Observation

Marks Secured ________out of 50

Full Name of the Evaluator:

Signature of the Evaluator Date of Evaluation: