Liveliness and readiness probe.

Spring boot actuator provides by default liveliness and readiness probes.

readinessProbe:

httpGet:

path: /actuator/health/readiness

port: 8080

initialDelaySeconds: 5

periodSeconds: 10

timeoutSeconds: 2

successThreshold: 1

failureThreshold: 3

livenessProbe:

httpGet:

path: /actuator/health/liveness

port: 8080

initialDelaySeconds: 15

periodSeconds: 20

timeoutSeconds: 2

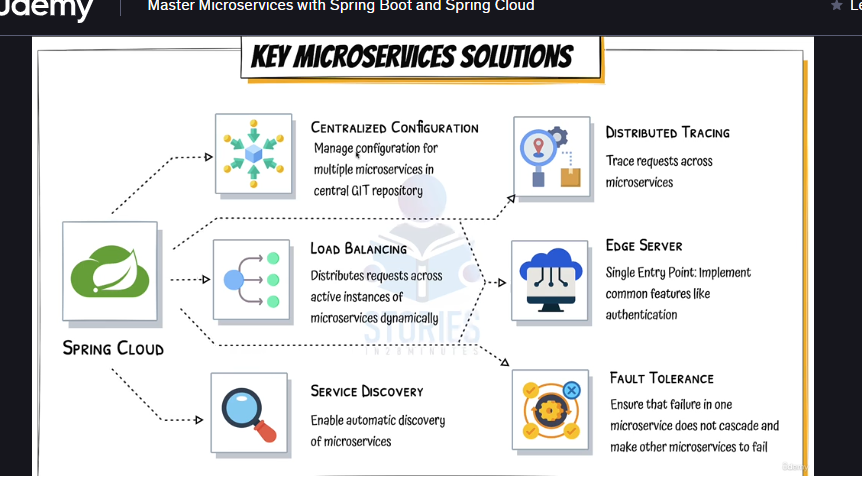
successThreshold: 1

failureThreshold: 3

application.yml

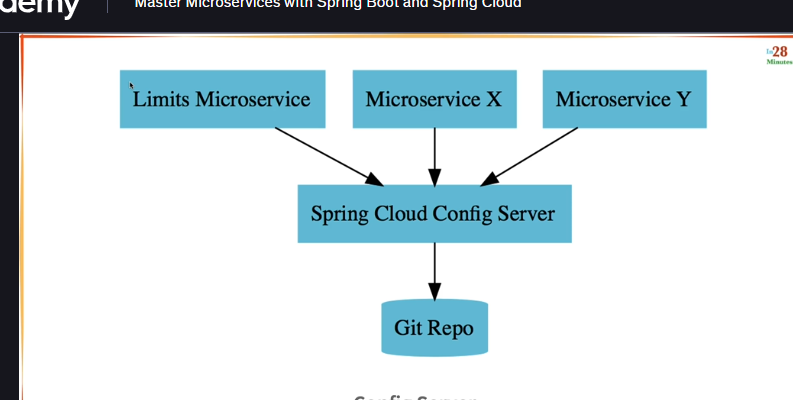
management.endpoint.health.probes.enabled=true

management.endpoints.web.exposure.include=health

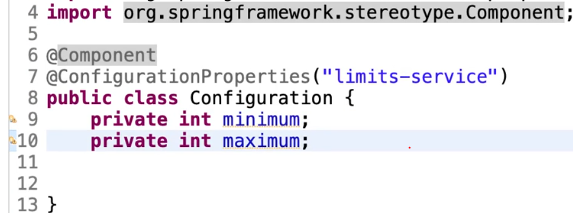


Centralized configurations:

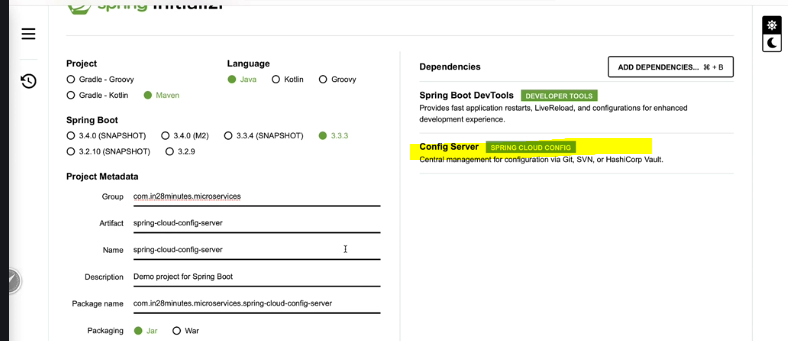
Spring config server.



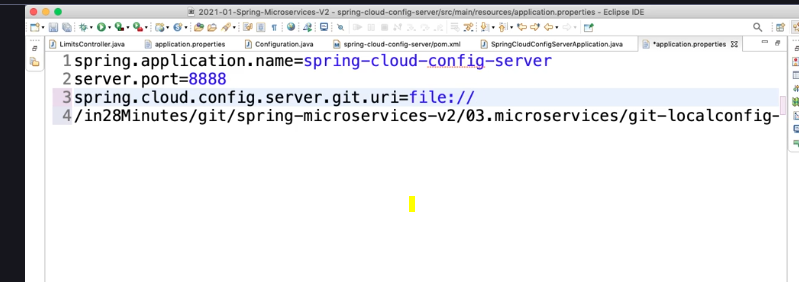
How to get values from properties file. By using @Value at property level and @ConfigurationProperties at the Class level.

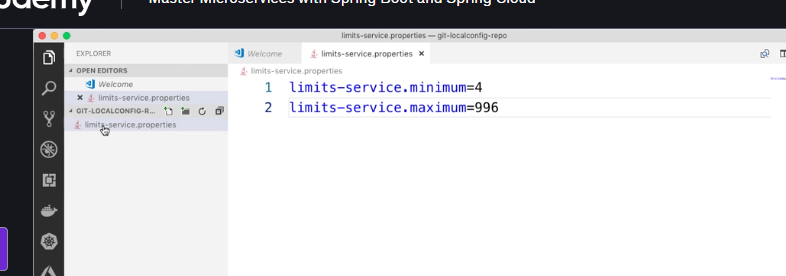


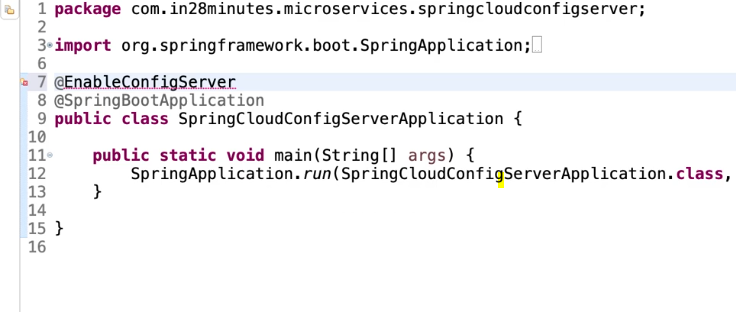
How to get the values from Config server?



In Config server application dependency is Config server while as in the Limit server one of the client is dependency is Config client.







@EnableConfigServer

@SpringBootApplication

public class SpringCloudConfigServerApplication {

public static void main(String[] args) {

SpringApplication.run(SpringCloudConfigServerApplication.class, args);

}

}

spring.application.name=spring-cloud-config-server

server.port=8888

spring.cloud.config.server.git.uri=file:///Ranga/git/01.udemy-course-repos/spring-micro-services/03.microservices/git-localconfig-repo

localhost:8888/limits-service/default

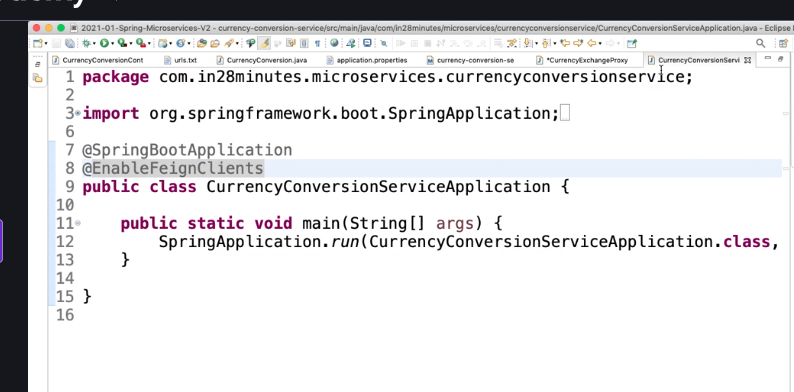
Profiles in Spring config server.

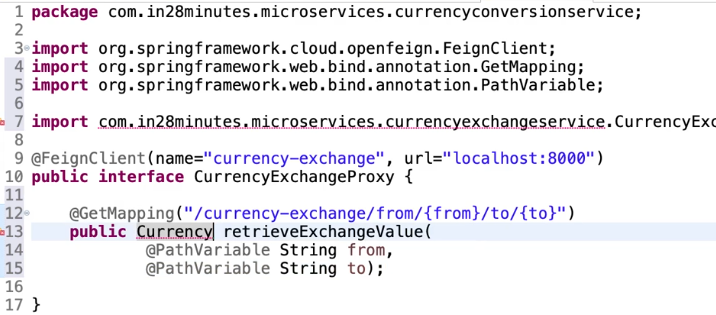
Limits-service-dev.properties

Limits-service.prod.properties

<http://localhost:8888/your-app-name/default>

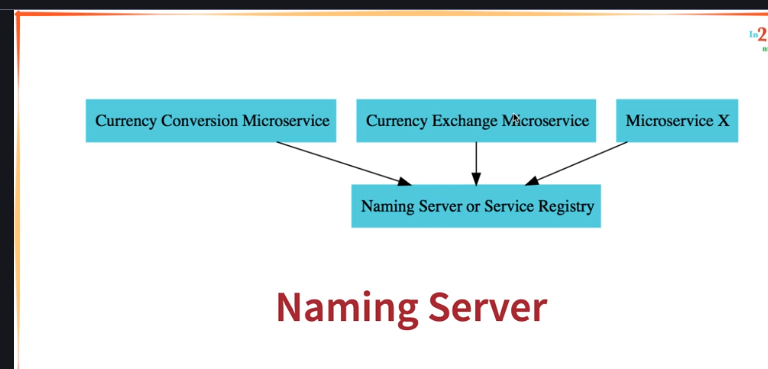
How to enable feign clients?

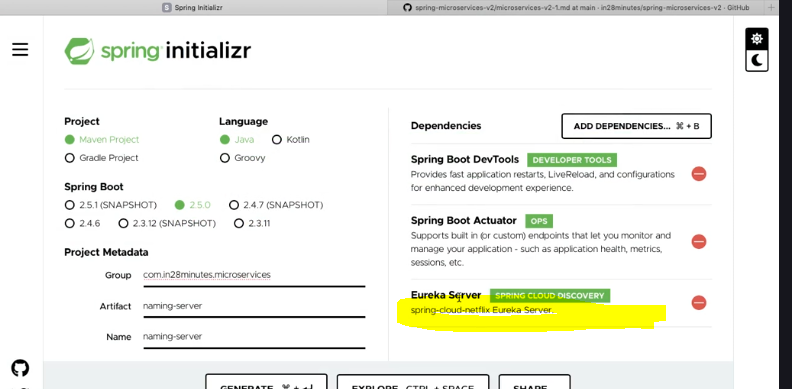


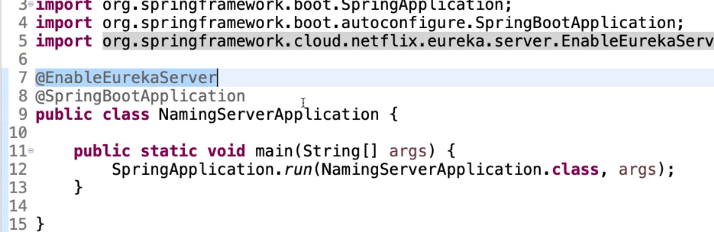


Why do we need Naming server or Service registry

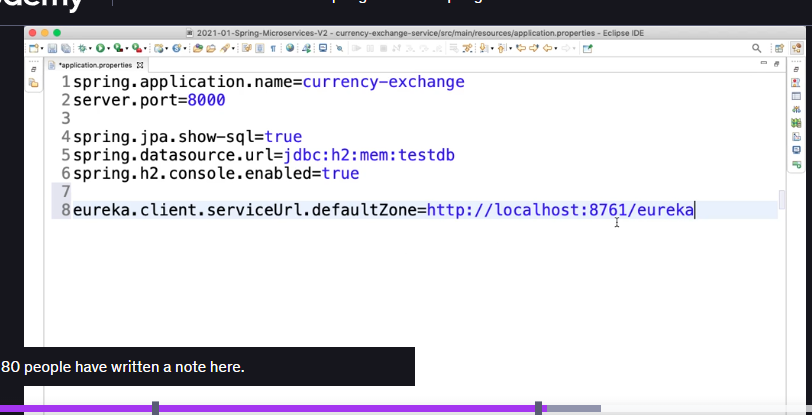
While using open feign in url we mention the host url, when the host url changes we cant keep on modifying the host url so for hat we use service registry.





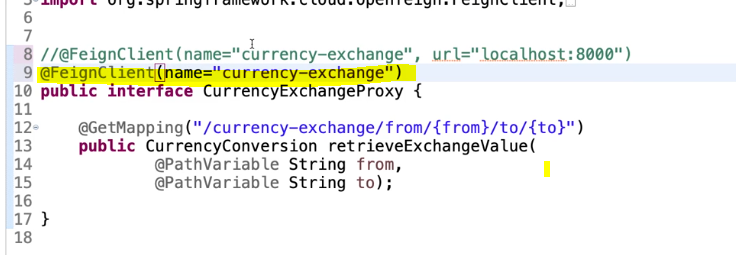


In the currency-exchange:

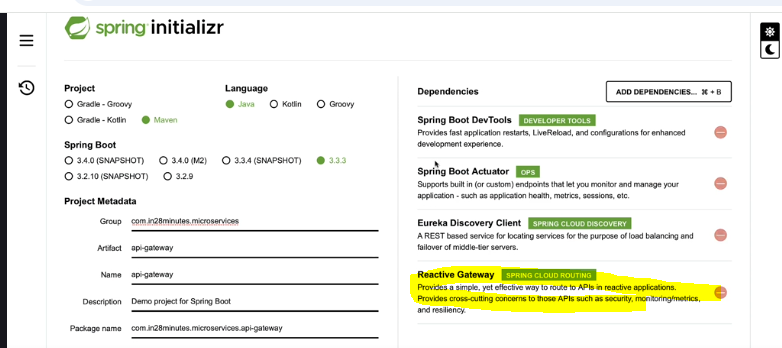


Load Balancing: By default it uses spring cloud load balancing.

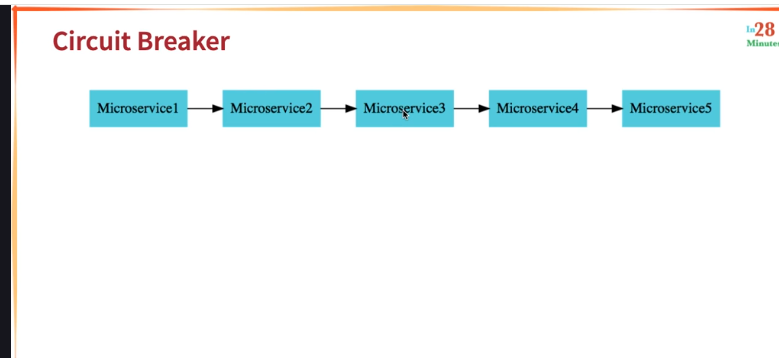
Just by adding name only feign talk to eureka and gets the currency-exchange service instance and sends the request.

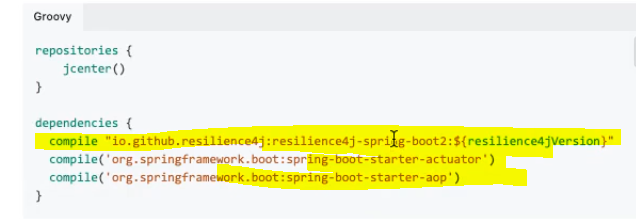


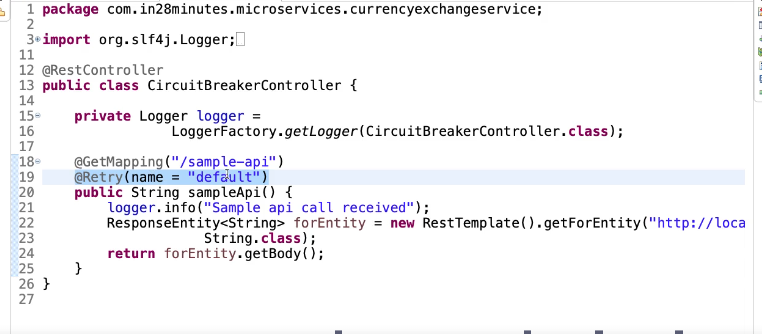
Spring API Gateway: Reactive gateway



Circuit breaker:







**import io.github.resilience4j.retry.annotation.Retry;**

**@RestController**

**public class ProductController {**

**@GetMapping("/products")**

**@Retry(name = "productService", fallbackMethod = "fallbackProducts")**

**public String getProducts() {**

**if (Math.random() < 0.7) {**

**throw new RuntimeException("Remote service failed");**

**}**

**return "Products fetched!";**

**}**

**public String fallbackProducts(Throwable t) {**

**return "Fallback: Cannot fetch products now.";**

**}**

**}**

**resilience4j.retry:**

**instances:**

**productService:**

**maxAttempts: 3**

**waitDuration: 1s**

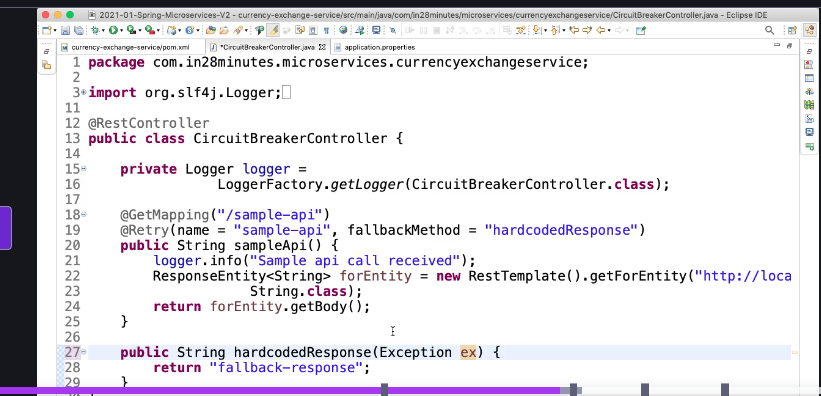
**retryExceptions:**

**- java.io.IOException**

**- java.util.concurrent.TimeoutException**

**ignoreExceptions:**

**- com.example.CustomBusinessException**



**import io.github.resilience4j.circuitbreaker.annotation.CircuitBreaker;**

**@RestController**

**public class OrderController {**

**@GetMapping("/order")**

**@CircuitBreaker(name = "orderService", fallbackMethod = "fallbackOrder")**

**public String getOrder() {**

**// Simulate remote call**

**if (Math.random() < 0.5) {**

**throw new RuntimeException("Remote service failed");**

**}**

**return "Order placed";**

**}**

**public String fallbackOrder(Throwable t) {**

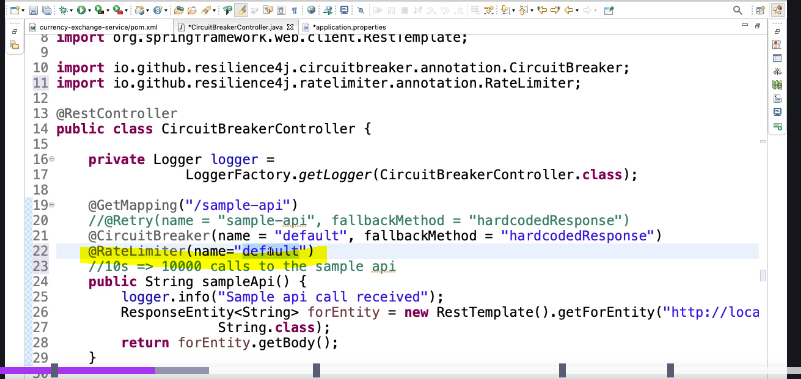
**return "Fallback: Unable to place order at the moment";**

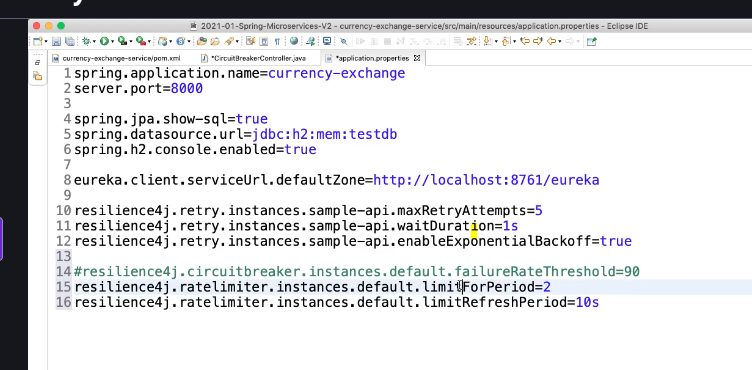
**}**

**}**

curl -X GET <http://localhost:8080/api/data>

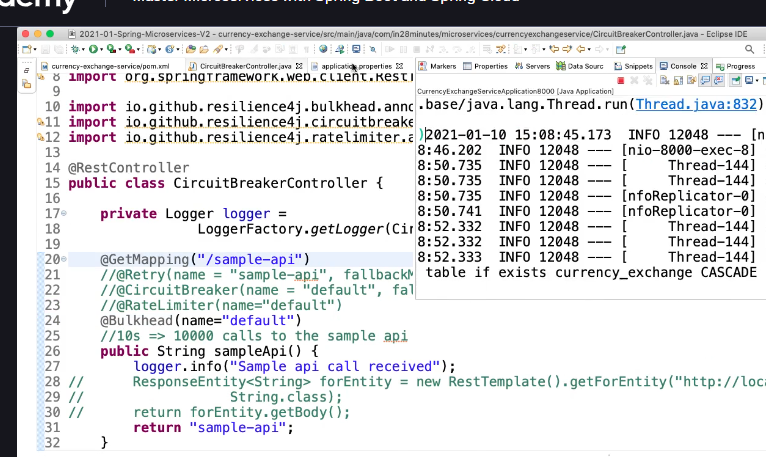
How to configure RateLimit:

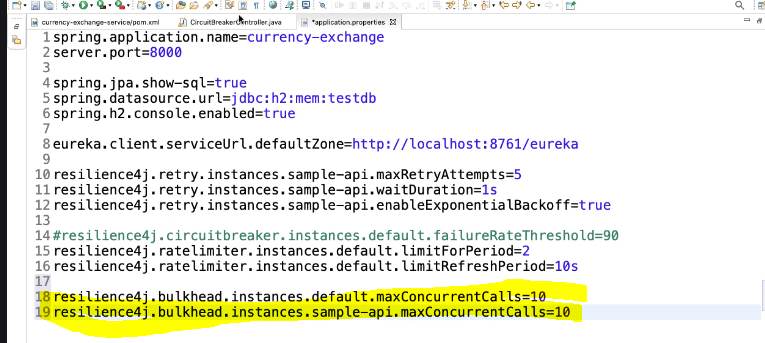




**What is bulk head in micro services?**

**How many concurrent calls we can allow at time for a specific end point.**





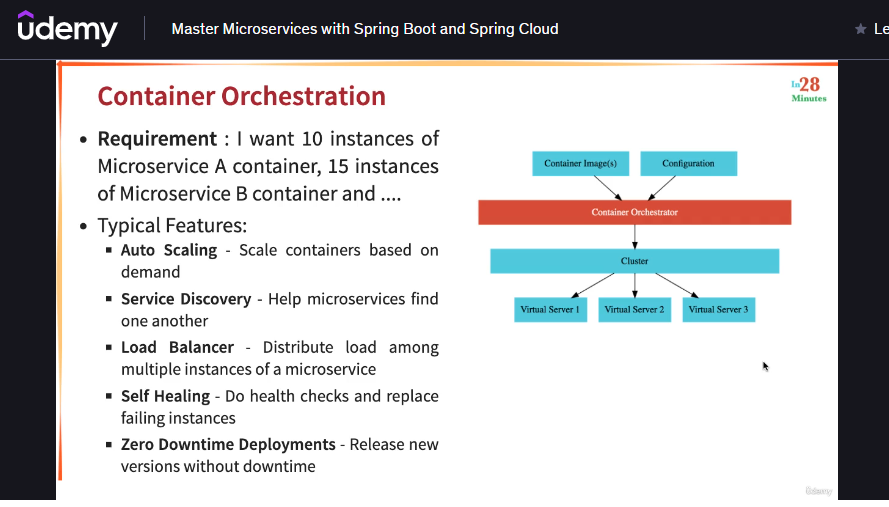
You're referencing Spring Cloud Sleuth — which is used for **distributed tracing** in microservices.

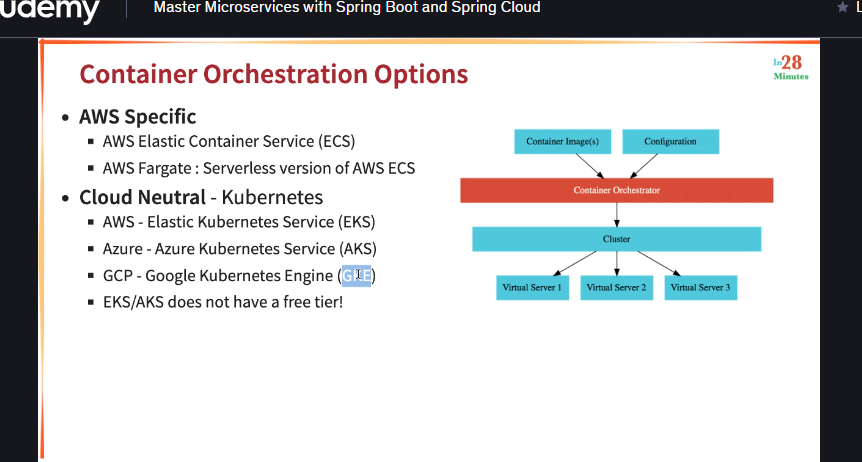
However, **Spring Cloud Sleuth is deprecated as of Spring Cloud 2022.x** and has been replaced by **Micrometer Tracing** + **Brave** or **OpenTelemetry**.

……………………………………………………………………………………..

Micro services with GCP

https://github.com/in28minutes/spring-microservices-v3/tree/main/05.kubernetes





Commands used in this project for AKS

1. gcloud config set project my-kubernetes-project-304910
2. gcloud container clusters get-credentials my-cluster --zone us-central1-c --project my-kubernetes-project-304910
4. kubectl create deployment hello-world-rest-api --image=in28min/hello-world-rest-api:0.0.1.RELEASE
5. kubectl get deployment
7. kubectl expose deployment hello-world-rest-api --type=LoadBalancer --port=8080
8. kubectl get services
9. kubectl get services --watch
11. curl 35.184.204.214:8080/hello-world
13. kubectl scale deployment hello-world-rest-api --replicas=3
15. gcloud container clusters resize my-cluster --node-pool default-pool --num-nodes=2 --zone=us-central1-c
17. kubectl autoscale deployment hello-world-rest-api --max=4 --cpu-percent=70
18. kubectl get hpa
20. kubectl create configmap hello-world-config --from-literal=RDS\_DB\_NAME=todos
21. kubectl get configmap
22. kubectl describe configmap hello-world-config
24. kubectl create secret generic hello-world-secrets-1 --from-literal=RDS\_PASSWORD=dummytodos
25. kubectl get secret
26. kubectl describe secret hello-world-secrets-1
28. kubectl apply -f deployment.yaml
30. gcloud container node-pools list --zone=us-central1-c --cluster=my-cluster
31. kubectl get pods -o wide
33. kubectl set image deployment hello-world-rest-api hello-world-rest-api=in28min/hello-world-rest-api:0.0.2.RELEASE
35. kubectl get services
36. kubectl get replicasets
37. kubectl get pods
39. kubectl delete pod hello-world-rest-api-58dc9d7fcc-8pv7r
41. kubectl scale deployment hello-world-rest-api --replicas=1
43. kubectl get replicasets
45. gcloud projects list
46. kubectl delete service hello-world-rest-api
47. kubectl delete deployment hello-world-rest-api
49. gcloud container clusters delete my-cluster --zone us-central1-c