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

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

LAB - 12 ➔ Implementing Microservices and Load Balancing

Prerequisites:
Implementation skill on Spring Boot App
Basic Idea on Microservices and Load Balancing

Exercise 1:

Now there is a demand to make some mathematical calculation over web. So, you need to create one server application (eureka) and two client applications, in which "client 2" will do the calculation and return the answer to "Client 1". The client 1 will handle the user interaction by getting input from user after then sending the same to "client 2" and then getting answer from "client 2" to return the same to "client 1". The server need to monitor both the clients.

Exercise 2:

Now the number of requests from the end users is raising above the threshold level, so its time to create "client 3" which is similar to "client 2" and have load balancer between them. Attach the load balancer in "client 1", so that, every request come to "client 1" need to be load balanced between "client 2" and "client 3".

❖ Watch The Video And Do In Eclipse Workspace

12 https://youtu.be/6LtQ-wJhju4?si=i3gzPFFRjlRMCGzZ

EX-MS-CLIENT-1

AppController.java

```
package com.klu;

import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

@RestController
@CrossOrigin
public class AppController {


	@GetMapping("/add/{a}/{b}")
	public String add(@PathVariable int a, @PathVariable int b) {
	    return "Client 1: " + (a + b);
	}

}
```

EX-MS-CLIENT-2

AppController.java

```
package com.klu;

import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

@RestController
@CrossOrigin
public class AppController {
     @GetMapping("/add/{a}/{b}")
     public String add(@PathVariable int a, @PathVariable int b) {
       return "Client 2: " + (a + b);
     }

}
```

EX-MS-LB

Calculator.java

```
package com.klu;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.cloud.loadbalancer.annotation.LoadBalancerClient;
import org.springframework.stereotype.Component;
import org.springframework.web.client.RestTemplate;


@Component
@LoadBalancerClient(value = "x", configuration = Addressconfig.class)

public class Calculator {

  @Autowired
  RestTemplate restTemplate;

  public String add(int a, int b) {
    return restTemplate.getForObject("http://x/add/" + a + "/" + b, String.class);
  }
}
```

Addressconfig.java

```
package com.klu;

import org.springframework.cloud.client.DefaultServiceInstance;
import org.springframework.cloud.loadbalancer.core.ServiceInstanceListSupplier;
import org.springframework.cloud.loadbalancer.support.ServiceInstanceListSuppliers;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;




@Configuration
public class Addressconfig {

   @Bean
   ServiceInstanceListSupplier
serviceInstanceListSupplier(ConfigurableApplicationContext context) {
       ServiceInstanceListSupplier serviceInstanceListSupplier =
ServiceInstanceListSuppliers.from("x",
               new DefaultServiceInstance("EX-MS-CLIENT-1", "x", "localhost", 8081, false),
               new DefaultServiceInstance("EX-MS-CLIENT-2", "x", "localhost", 8082, false)
       );

       return ServiceInstanceListSupplier.builder()
               .withBase(serviceInstanceListSupplier)
               .build(context);
   }
}
```

AppController.java

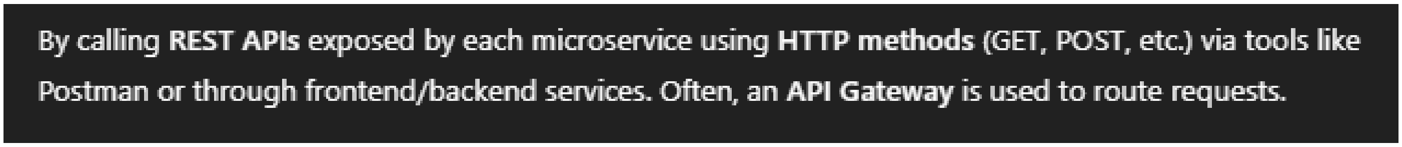
```
package com.klu;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
@CrossOrigin
public class AppController {

   @Autowired
   Calculator obj;

   @GetMapping("/add/{a}/{b}")
   public String add(@PathVariable int a, @PathVariable int b) {
      return obj.add(a, b);
   }
}
```

ApplicationConfiguration.java

```
package com.klu;

import org.springframework.boot.web.client.RestTemplateBuilder;
import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.client.RestTemplate;

@Configuration
public class ApplicationConfiguration {

   @LoadBalanced
   @Bean
   RestTemplate restTemplate(RestTemplateBuilder builder) {
      return builder.build();
   }
}
```

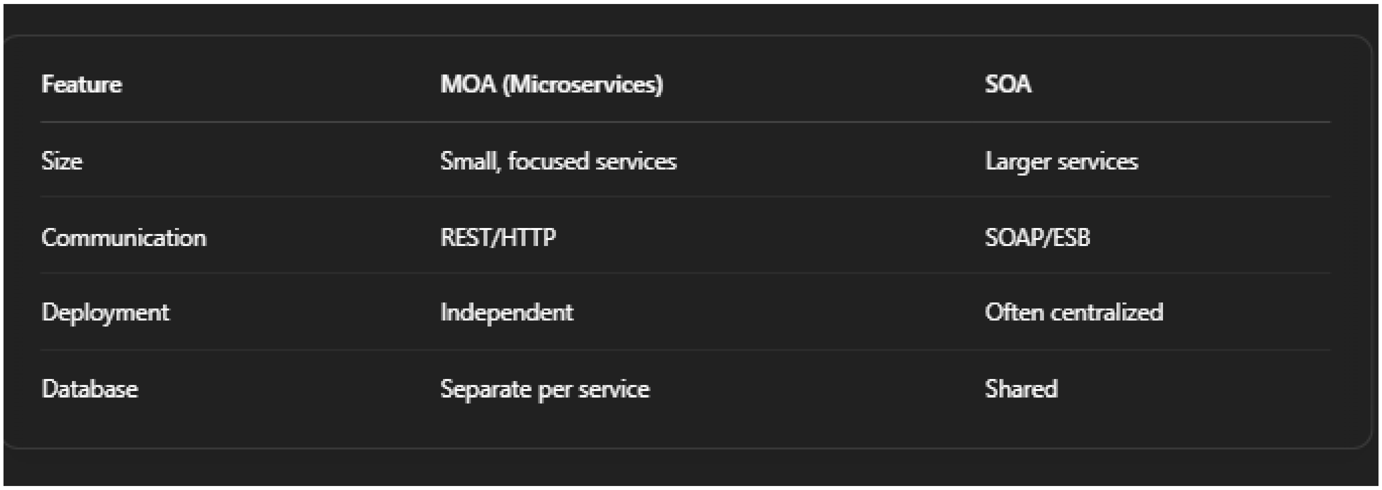
VIVA QUESTIONS:

1. How can we access RESTful in Microservices?



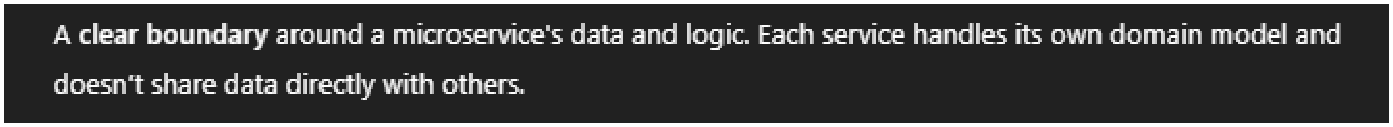
By calling **REST APIs** exposed by each microservice using **HTTP methods** (GET, POST, etc.) via tools like Postman or through frontend/backend services. Often, an **API Gateway** is used to route requests.

2. What's the difference between a microservices-oriented architecture (MOA) and a service-oriented architecture (SOA)?



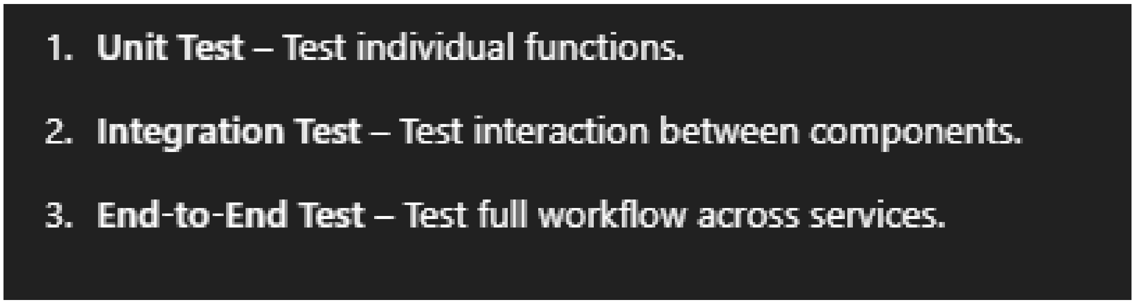
Feature	MOA (Microservices)	SOA
Size	Small, focused services	Larger services
Communication	REST/HTTP	SOAP/ESB
Deployment	Independent	Often centralized
Database	Separate per service	Shared

3. What does the term "bounded context" mean in relation to microservices?



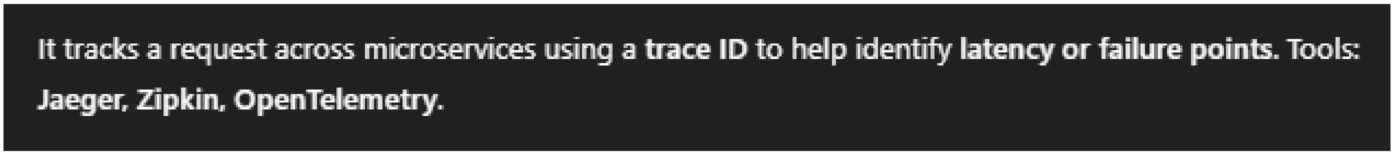
A clear boundary around a microservice's data and logic. Each service handles its own domain model and doesn't share data directly with others.

4. Explain three types of Tests for Microservices?



1. **Unit Test** – Test individual functions.
2. **Integration Test** – Test interaction between components.
3. **End-to-End Test** – Test full workflow across services.

5. How is distributed tracing used in Microservices?



It tracks a request across microservices using a **trace ID** to help identify **latency or failure points**. Tools: **Jaeger, Zipkin, OpenTelemetry.**

(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: