Spring Cloud is a **Spring module** that provides the **RAD** (Rapid Application Development) feature to the Spring framework.

The messaging pattern follows the **Publish-Subscribe** (Pub-Sub) model. In the Pub-Sub model, the sender of the message is called publisher and receiver of the message is called subscribers. **Apache Kafka** and **RabbitMQ** are the popular high throughput messaging system

Service provider: server

Service Requester: client

Actuator – is used to monitor micro-services

**Client side load balancing**: through feign framework load balancing is called client side load balancing

Spring.cloud.loadbalancer is used by feign old load balancer was ribbon

Zuul gateway is old now is not supported by Netflix server new gateway is spring cloud gateway

eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka/

spring.cloud.gateway.discovery.locator.enabled=true

spring.cloud.gateway.discovery.locator.lower-case-service-id=true

spring.application.name=fx-discovery-server

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

Que: how to trace (debug) micro service request and what is distributed tracing?

Ans: all involved microservice send all the information out through distributed tracing server

Server can be in memory or real zipkin is a distributed tracing server

Microservice can be trace by unique id and sleuth is framework which help you handle this.

Q: if distributed server get down then how would you handle

Ans: we can use Rabbit MQ rabbit will get the data from microservice if server is down then it will pass to server