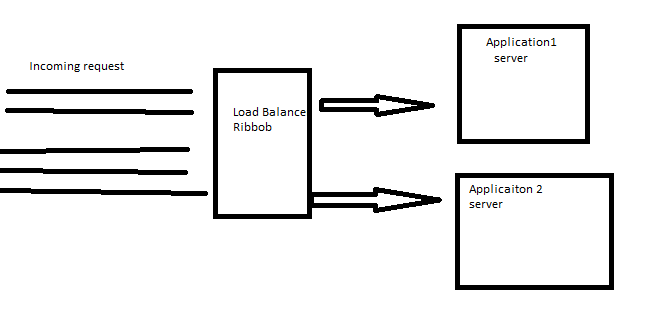
Client side load balancing in microservices.

Spring-cloud Ribbon and Eureka Server

Note: Ribbon Client is by default initialized in eureka server

Load balancing is the process of distributing incoming request to multiple server . Multiple server means we have to deploy our application in multiple server . or we have multiple instances of application in server .

When client send the request then loadbalancer will decide which server will take care of it .



Load balancing algorithm :

Round Ribbon (Default) : Client requests are distributed to application servers in rotation. For example, if you have three application servers: the first client request to the first application server in the list, the second client request to the second application server, the third client request to the third application server, the fourth to the first application server and so on.

Weighted Least Connection : Weighted Least Connection builds on the Least Connection load balancing algorithm to account for differing application server characteristics. The administrator assigns a weight to each application server based on criteria of their choosing to demonstrate the application servers traffic-handling capability. The LoadMaster is making the load balancing criteria based on active connections and application server weighting.

source IP hash : source IP hash load balancing algorithm that combines source and destination IP addresses of the client and server to generate a unique hash key. The key is used to allocate the client to a particular server. As the key can be regenerated if the session is broken, the client request is directed to the same server it was using previously. This is useful if it’s important that a client should connect to a session that is still active after a disconnection.

# What is circuit breaker : Hystrix, Resillience4j API apply circuit breaker pattern .

Note: all the properties to control the circuit breaker pattern are configurable

\_---- > actually circuit breaker is the technique to disconnent the services from heavy loaded or high latency services or shutdown services . eg. Suppose when a high voltage passes , the fuse trips thereby preventing other appliances from burning out . When the fusse trips, circuit is open and no current passes through .when the fuse is reset, the circuit is closed and the current starts passing through it again . this pattern when applied in fault tolerance is called the circuit breaker pattern .

Hystrix : The sort of alternate arrangement when the circuit is open called a fallback pattern . Hystrix allows us to mention any alternate piece of code that you wish to run if a service is down . It does not provide the same result as we expect but provides some form of data instead of error message .

Circuit open = disconnecting particular service from sending request if that service is not working fine .

Fallback execute when : An error occurs, A timeOut occurs, Circuit opens

Dependency :

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-hystrix</artifactId>

<version>1.4.7.RELEASE</version>

</dependency>

Q. What is synchronous and asynchronous call in microservice with hystrix ?