## 服务发现简介

使用服务发现组件后的架构图：

|  |
| --- |
| IMG_256 |

* Eureka Server：服务的注册中心，负责维护注册的服务列表。
* Service Provider：服务提供方，作为一个Eureka Client，向Eureka Server做服务注册、续约和下线等操作，注册的主要数据包括服务名、机器ip、端口号、域名等等。
* Service Consumer：服务消费方，作为一个Eureka Client，向Eureka Server获取Service Provider的注册信息，并通过远程调用与Service Provider进行通信。

Service Provider和Service Consumer不是严格的概念，Service Consumer也可以随时向Eureka Server注册，来让自己变成一个Service Provider。

服务提供者、服务消费者、服务发现组件三者之间的关系：

1. 各个微服务在启动时，将自己的网络地址等信息注册到服务发现组件，服务发现组件会存储这些信息
2. 服务消费者可以从服务发现组件中查询这些服务提供者的网络地址，并使用该地址调用服务提供者的接口
3. 各个微服务与服务发现组件通过发送心跳，服务发现组件如长时间无法与某个微服务实例通信，就会注销该实例
4. 微服务网络地址发生变更，会重新注册到服务发现组件

服务发现组件：

Eureka server配置信息：

|  |
| --- |
| server.port=8080 spring.application.name= eureka-server eureka.instance.hostname=localhost  #避免eureka向自己进行注册(是否注册自己的信息到eureka server 默认是true) eureka.client.register-with-eureka=**false** #避免eureka查找服务列表(是否拉取其他服务的信息，默认是true) eureka.client.fetch-registry=**false** # 关闭自我保护机制 eureka.server.enable-self-preservation=**false** # 每隔10s扫描服务列表，移除失效服务 eureka.server.eviction-interval-timer-in-ms=10000 eureka.client.service-url.defaultZone=http://${eureka.instance.hostname}:${server.port}/eureka |

服务提供者：

配置信息

|  |
| --- |
| server.port=8081 spring.application.name= eureka-user eureka.instance.hostname=localhost #开启健康检查(将健康状态传播到eureka server) eureka.client.healthcheck.enabled=**true** # 每隔10s发送一次心跳（服务续约的间隔，默认为30） eureka.instance.lease-renewal-interval-in-seconds=10 # 告知服务端30秒还未收到心跳的话，就将该服务移除列表（服务失效时间，默认90） eureka.instance.lease-expiration-duration-in-seconds=30   #其他服务获取地址时，提供ip而不是主机名 eureka.instance.prefer-ip-address=**true** eureka.client.service-url.defaultZone=http://localhost:8080/eureka/  spring.datasource.url=jdbc:mysql://127.0.0.1:3306/springcloud?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC spring.datasource.driver-class-name=com.mysql.jdbc.Driver spring.datasource.password=asd3884 spring.datasource.username=root  #mybatis mybatis.mapper-locations=classpath:mapper/user.xml |

服务提供者要向eureka server注册服务，并且完成服务续约等工作

1. 服务注册：

服务提供者在启动时，会检测配置属性

#避免eureka向自己进行注册(是否注册自己的信息到eureka server 默认是true)  
eureka.client.register-with-eureka=**true**

中的参数是否正确（默认为true）

如果确实为true，则会向eureka server 发起一个Rest请求，并携带自己的元数据信息，eureka server会把这些信息储存

1. 服务续约：

注册服务完成后，服务提供者会维持一个心跳（定时向eureka server发起Rest请求）

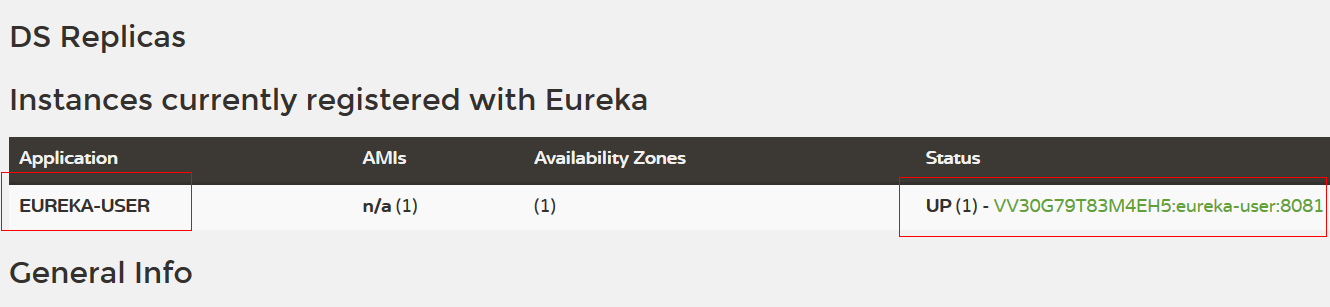
两个参数可以修改服务续约的行为：

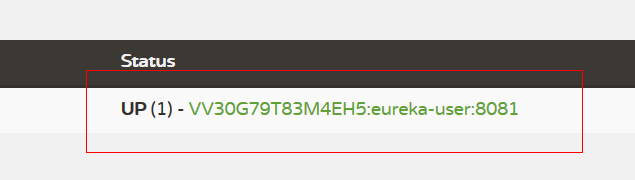
# 每隔10s发送一次心跳(服务续约的间隔，默认30秒)  
eureka.instance.lease-renewal-interval-in-seconds=10  
# 告知服务端30秒还未收到心跳的话，就将该服务移除列表(服务失效时间，默认90秒)  
eureka.instance.lease-expiration-duration-in-seconds=30

默认情况下：每隔30秒服务会向注册中心发送一次心跳，超过90秒没有发送心跳，eureka server会认为服务宕机，会从服务列表中移除

分别启动eureka、user

访问http://localhost:8080





如上：eureka监控页面 status：up(1) 启动了一个实例，没有集群

默认格式：${hostname}:${spring.application.name}:${server.port}

可以通过属性修改格式：

#eureka 监控页面status修改格式  
eureka.instance.instance-id=

${eureka.instance.hostname}:${spring.application.name}:${server.port}

再次访问http://localhost:8080如下：

