# Spring Cloud使用Eureka集群搭建实现高可用服务注册中心

<http://c.biancheng.net/view/5332.html>

前面我们搭建的注册中心只适合本地开发使用，在生产环境中必须搭建一个集群来保证高可用。Eureka 的集群搭建方法很简单：每一台 Eureka 只需要在配置中指定另外多个 Eureka 的地址就可以实现一个集群的搭建了。  
  
下面我们以 2 个节点为例来说明搭建方式。假设我们有 master 和 slaveone 两台机器，需要做的就是：

* 将 master 注册到 slaveone 上面。
* 将 slaveone 注册到 master 上面。

如果是 3 台机器，以此类推：

* 将 master 注册到 slaveone 和 slavetwo 上面。
* 将 slaveone 注册到 master 和 slavetwo 上面。
* 将 slavetwo 注册到 master 和 slaveone 上面。

## 搭建步骤

创建一个新的项目 eureka-server-cluster，配置跟 eureka-server 一样。  
  
首先，我们需要增加 2 个属性文件，在不同的环境下启动不同的实例。增加 application-master.properties：

server.port=8761  
# 指向你的从节点的Eureka  
eureka.client.serviceUrl.defaultZone=http://用户名:密码@localhost:8762/eureka/

增加 application-slaveone.properties：

server.port=8762  
# 指向你的主节点的Eureka  
eureka.client.serviceUrl.defaultZone=http://用户名:密码 @localhost:8761/eureka/

在 application.properties 中添加下面的内容：

spring.application.name=eureka-server-cluster  
# 由于该应用为注册中心, 所以设置为false, 代表不向注册中心注册自己  
eureka.client.register-with-eureka=false  
# 由于注册中心的职责就是维护服务实例, 并不需要检索服务, 所以也设置为 false  
eureka.client.fetch-registry=false  
  
spring.security.user.name=zhangsan  
spring.security.user.password=123456  
  
# 指定不同的环境  
spring.profiles.active=master

在 A 机器上默认用 master 启动，然后在 B 机器上加上 --spring.profiles.active=slaveone 启动即可。  
  
这样就将 master 注册到了 slaveone 中，将 slaveone 注册到了 master 中，无论谁出现问题，应用都能继续使用存活的注册中心。  
  
之前在客户端中我们通过配置 eureka.client.serviceUrl.defaultZone 来指定对应的注册中心，当我们的注册中心有多个节点后，就需要修改 eureka.client.serviceUrl.defaultZone 的配置为多个节点的地址，多个地址用英文逗号隔开即可：

eureka.client.serviceUrl.defaultZone=http://zhangsan:123456@localhost:8761  
                /eureka/,http://zhangsan:123456@localhost:8762/eureka/

# Eureka开发时快速移除失效服务

在实际开发过程中，我们可能会不停地重启服务，由于 Eureka 有自己的保护机制，故节点下线后，服务信息还会一直存在于 Eureka 中。我们可以通过增加一些配置让移除的速度更快一点，当然只在开发环境下使用，生产环境下不推荐使用。  
  
首先在我们的 eureka-server 中增加两个配置，分别是关闭自我保护和清理间隔：

eureka.server.enable-self-preservation=false  
# 默认 60000 毫秒  
eureka.server.eviction-interval-timer-in-ms=5000

然后在具体的客户端服务中配置下面的内容：

eureka.client.healthcheck.enabled=true  
# 默认 30 秒  
eureka.instance.lease-renewal-interval-in-seconds=5  
# 默认 90 秒  
eureka.instance.lease-expiration-duration-in-seconds=5

eureka.client.healthcheck.enabled 用于开启健康检查，需要在 pom.xml 中引入 actuator 的依赖，代码如下所示。

<dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-actuator</artifactId>  
</dependency>

其中：

* eureka.instance.lease-renewal-interval-in-seconds 表示 Eureka Client 发送心跳给 server 端的频率。
* eureka.instance.lease-expiration-duration-in-seconds 表示 Eureka Server 至上一次收到 client 的心跳之后，等待下一次心跳的超时时间，在这个时间内若没收到下一次心跳，则移除该 Instance。

更多的 Instance 配置信息可参考源码中的配置类：org.springframework.cloud.netflix.eureka.EurekaInstanceConfigBean。  
  
更多的 Server 配置信息可参考源码中的配置类：org.springframework.cloud.netflix.eureka.server.EurekaServerConfigBean。