



链滴

# Spring-Cloud 系列第 2 篇：spring-cloud-eureka

作者：[xjtushilei](#)

原文链接：<https://ld246.com/article/1497579524718>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

自学spring-cloud系列，越来越感觉spring-cloud很强大！

主要分为以下几篇：

1. [spring-cloud-config](#): 分布式配置管理
2. [spring-cloud-eureka](#): 服务注册与发现
3. [spring-cloud-eureka-consumer](#): 远程服务调用和及其负载均衡
4. [spring-cloud-Hystrix](#): 熔断器保证服务高可用
5. [spring-cloud-config-eureka-ribbon](#): 分布式配置管理的高可用
6. [spring-cloud-bus](#): 配置信息的实时更新
7. [spring-cloud-zuul](#): 网关基础服务

## 介绍

spring-cloud-eureka，被动式的服务发现，统一监控和管理你的服务列表。

## 什么是服务发现？

服务发现就像聊天室一个,每个用户来的时候去服务器上注册,这样他的好友们就能看到你,你同时也将取好友的上线列表. 在微服务中,服务就相当于聊天室的用户,而服务注册中心就像聊天室服务器一样,目服务发现的解决方案有Eureka,Consul,Etcd,Zookeeper,SmartStack,等等.

## 如何使用

- 创建server端
- 创建client端

### 1. 创建server端

#### 1.1 单机版

pom.xml:

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-eureka-server</artifactId>
</dependency>
```

当然了，我已经在全局加入了一些其他配置文件，因为我使用了模块式的开发，所以这里很简单。

配置文件:

```
server:
  port: 8761
```

```
spring:
```

```
application:
  name: eureka-server

eureka:
  instance:
    lease-expiration-duration-in-seconds: 6
    lease-renewal-interval-in-seconds: 2
  client:
    service-url:
      defaultZone: http://localhost:${server.port}/eureka/
```

一般端口都是8761，可以随意设置。

开发的时候，一般要设置以下两点

**lease-expiration-duration-in-seconds**: 6 意思是6秒不发送心跳检查，就删除该实例，默认90秒

**lease-renewal-interval-in-seconds**: 2 心跳检查的时间，默认30秒

这里报一个 **bug**：我设置6秒还是不管用，依然是90秒才能剔除。可能是我时间设置的太短吗？大家以留言告诉我为什么。

启动：

```
@SpringBootApplication
@EnableConfigServer
public class SpringCloudConfigServerApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringCloudConfigServerApplication.class, args);
    }
}
```

在启动文件里，加入这样一句话就好啦。

## 1.2 多节点版本

在系统的hosts里写入：

```
127.0.0.1 peer1
127.0.0.1 peer2
```

节点1配置文件 **application-peer1.yml**：

```
server:
  port: 8761

spring:
  application:
    name: eureka-server
```

```
eureka:
  instance:
    hostname: peer1
#  lease-expiration-duration-in-seconds: 6
#  lease-renewal-interval-in-seconds: 2
  client:
    service-url:
      defaultZone: http://peer2:8762/eureka/
```

节点2配置文件 `application-peer2.yml` :

```
server:
  port: 8762

spring:
  application:
    name: eureka-server

eureka:
  instance:
#  lease-expiration-duration-in-seconds: 6
#  lease-renewal-interval-in-seconds: 2
    hostname: peer2
  client:
    service-url:
      defaultZone: http://peer1:8761/eureka/
```

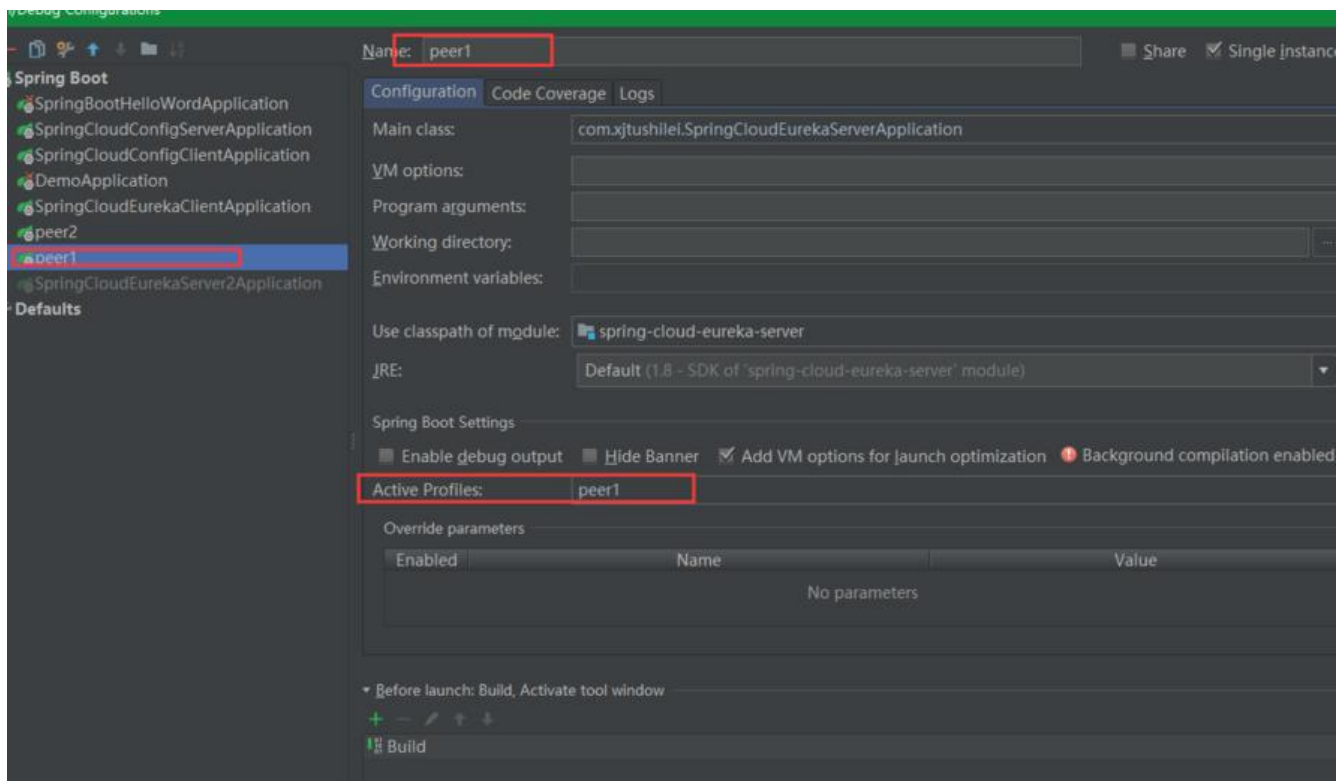
如果有更多个节点，更改端口号即可，并在 `defaultZone`:后面用逗号隔开，增加更多的就好了。

启动方法：

采用不同的配置文件启动：

```
java -jar eureka-server-1.0.0.jar --spring.profiles.active=peer1
java -jar eureka-server-1.0.0.jar --spring.profiles.active=peer2
```

如果是用IDEA环境下运行，直接新配置一个运行环境就好了，如下图：



然后在 dashboard 里查看，可以看到：

DS Replicas

peer2

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
EUREKA-SERVER	n/a (2)	(2)	UP (2) - DESKTOP-5PCKUJ5.eureka-server:8761 , DESKTOP-5PCKUJ5.eureka-server:8762

General Info

Name	Value
total-avail-memory	629mb
environment	test
num-of-cpus	8
current-memory-usage	169mb (26%)
server-uptime	00:00
registered-replicas	http://peer2:8762/eureka/
unavailable-replicas	
available-replicas	http://peer2:8762/eureka/ ,

Instance Info

这里有好多坑，只有你踩过了才能发现真理。其中最主要的是不能用一样的hostname，注册时间有慢和剔除时间有点慢。

## 2. 创建client端

当然了，也很简单。

pom.xml:

```
<!--监控-->
<dependency>
```

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

<!--服务注册-->
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
```

pom需要监控和服务注册，同样，推荐使用模块化开发，直接在顶层配置这两个，所有的文件都不需额外配置。

配置文件：

```
server.port=8083
spring.application.name=eureka-client-1
eureka.client.service-url.defaultZone= http://peer1:8761/eureka/,http://peer2:8761/eureka/
```

这里配置也很简单，告诉我在哪里就好了。如果有多个service-url，直接增加就行了,如上所示。

## 示例源码

所有源码在我的github仓库里，传送门：<https://github.com/xjtushilei/spring-cloud-simples.git>

## 支持

如果你喜欢~ 给个星