

# Spring Cloud Eureka：服务注册与发现

Spring Cloud Eureka是Spring Cloud Netflix 子项目的核心组件之一，主要用于微服务架构中的服务治理。本文将对搭建Eureka注册中心，搭建Eureka客户端，搭建Eureka集群及给Eureka注册中心添加登录认证进行介绍。

## Eureka简介

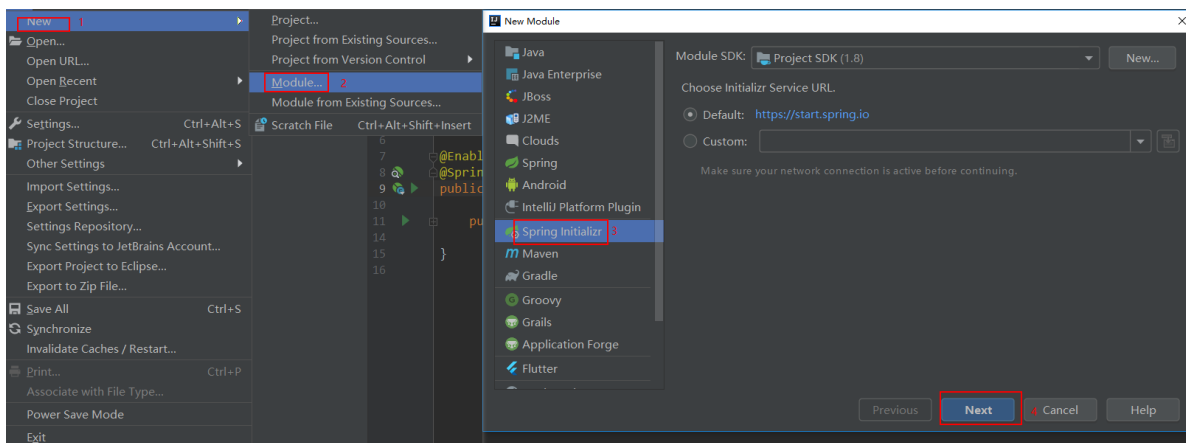
在微服务架构中往往会有一个注册中心，每个微服务都会向注册中心去注册自己的地址及端口信息，注册中心维护着服务名称与服务实例的对应关系。每个微服务都会定时从注册中心获取服务列表，同时汇报自己的运行情况，这样当有的服务需要调用其他服务时，就可以从自己获取到的服务列表中获取实例地址进行调用，Eureka实现了这套服务注册与发现机制。

## 搭建Eureka注册中心

这里我们以创建并运行Eureka注册中心来看看在IDEA中创建并运行SpringCloud应用的正确姿势。

## 使用IDEA来创建SpringCloud应用

- 创建一个eureka-server模块，并使用Spring Initializer初始化一个SpringBoot项目



- 填写应用信息

New Module

Project Metadata

Group:

Artifact:

Type:

Language:

Packaging:

Java Version:

Version:

Name:

Description:

Package:

- 选择你需要的SpringCloud组件进行创建

Dependencies

Developer Tools

Web

Template Engines

Security

SQL

NoSQL

Messaging

I/O

Ops

Testing

Spring Cloud

Spring Cloud Security

Spring Cloud Tools

Spring Cloud Config

☒ Spring Cloud Discovery 1

Spring Cloud Routing

Spring Cloud Circuit Breaker

Spring Cloud Tracing

Spring Cloud Messaging

Pivotal Cloud Foundry

☐ Eureka Discovery Client

☒ Eureka Server 2

☐ Apache Zookeeper Discovery

☐ Cloud Foundry Discovery

☐ Consul Discovery

Eureka Server

spring-cloud-netflix Eureka Server

Service Registration and Discovery

Selected Dependencies

Spring Cloud Discover

Eureka Server X

- 创建完成后会发现pom.xml文件中已经有了eureka-server的依赖

```
1 <dependency>
2   <groupId>org.springframework.cloud</groupId>
3   <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
4 </dependency>Copy to clipboardErrorCopied
```

- 在启动类上添加@EnableEurekaServer注解来启用Euerka注册中心功能

```

1  @EnableEurekaServer
2  @SpringBootApplication
3  public class EurekaServerApplication {
4
5      public static void main(String[] args) {
6          SpringApplication.run(EurekaServerApplication.class, args);
7      }
8
9  }Copy to clipboardErrorCopied

```

- 在配置文件application.yml中添加Eureka注册中心的配置

```

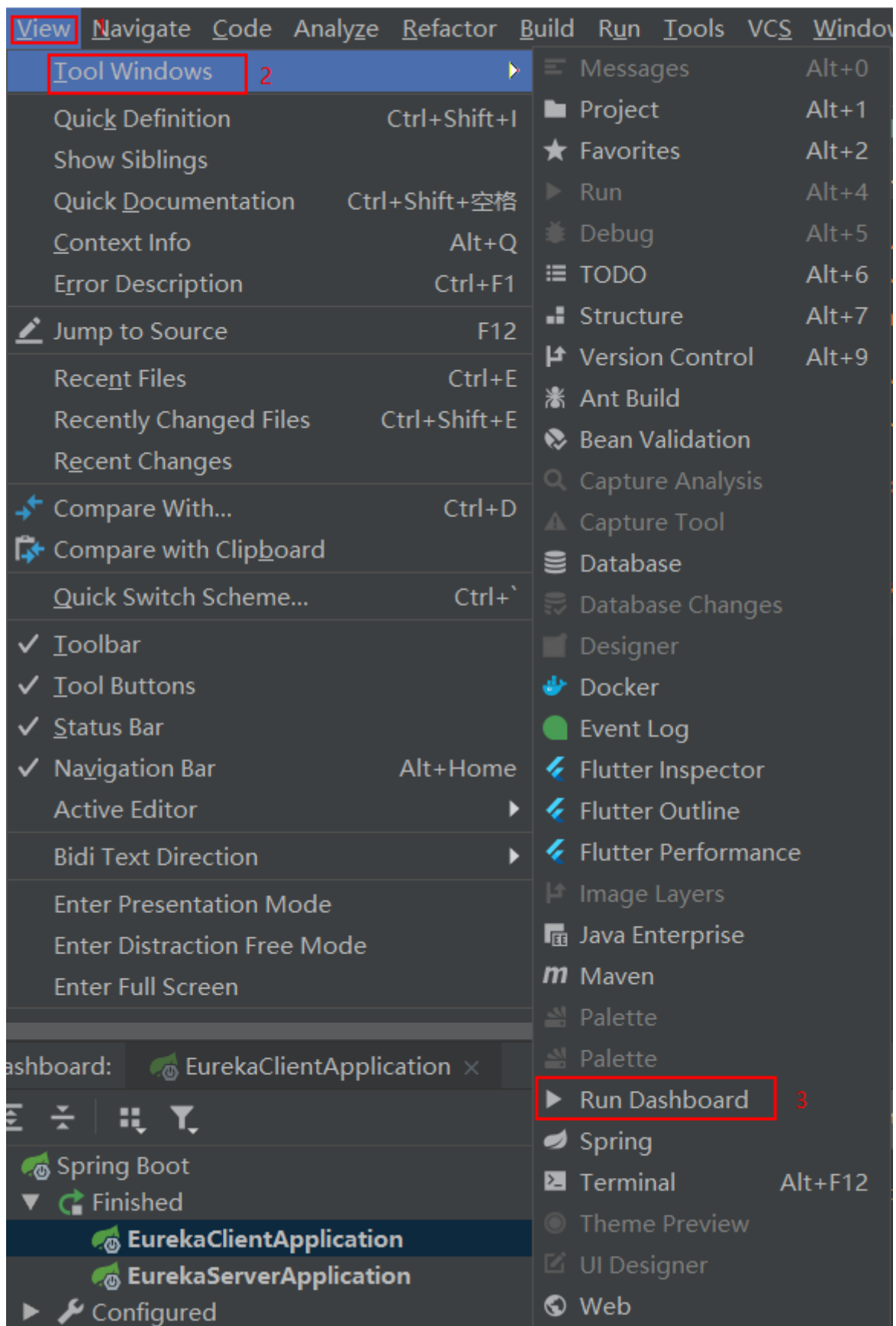
1  server:
2      port: 8001 #指定运行端口
3  spring:
4      application:
5          name: eureka-server #指定服务名称
6  eureka:
7      instance:
8          hostname: localhost #指定主机地址
9      client:
10         fetch-registry: false #指定是否要从注册中心获取服务（注册中心不需要开启）
11         register-with-eureka: false #指定是否要注册到注册中心（注册中心不需要开启）
12     server:
13         enable-self-preservation: false #关闭保护模式Copy to clipboardErrorCopied

```

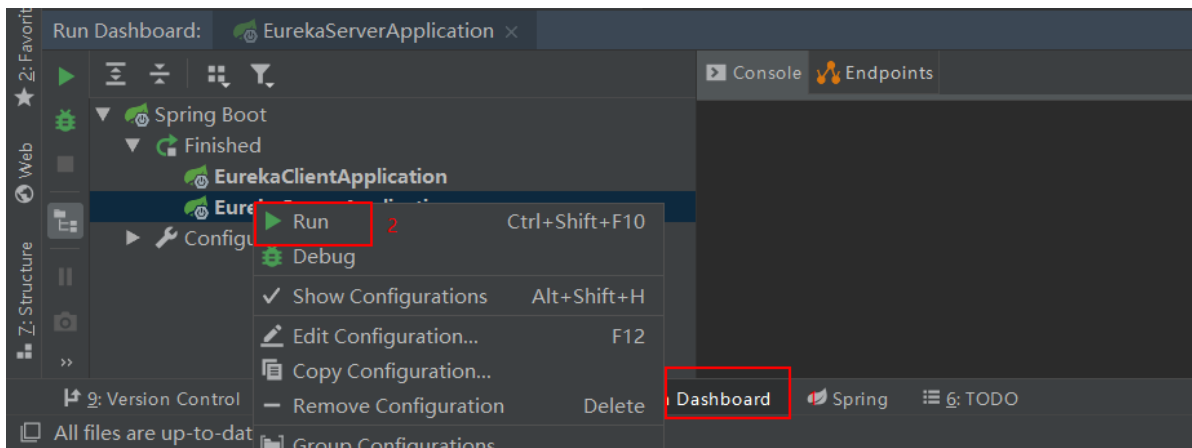
## 使用IDEA的Run Dashboard来运行SpringCloud应用

此时服务已经创建完成，点击启动类的main方法就可以运行了。但是在微服务项目中我们会启动很多服务，为了便于管理，我们使用IDEA的Run Dashboard来启动。


- 打开Run Dashboard，默认情况下，当IDEA检查到你的项目中有SpringBoot应用时，会提示你开启，如果你没开启，可以用以下方法开启。



- 运行SpringCloud应用



- 运行完成后访问地址 <http://localhost:8001/> 可以看到Eureka注册中心的界面

 HOME LAST 1000 SINCE STARTUP

### System Status

Environment	test	Current time	2019-09-07T16:45:46 +0800
Data center	default	Uptime	00:00
		Lease expiration enabled	true
		Renews threshold	1
		Renews (last min)	0

THE SELF PRESERVATION MODE IS TURNED OFF. THIS MAY NOT PROTECT INSTANCE EXPIRY IN CASE OF NETWORK/OTHER PROBLEMS.

### DS Replicas

### Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
No instances available			

## 搭建Eureka客户端

- 新建一个eureka-client模块，并在pom.xml中添加如下依赖

```
1 <dependency>
2   <groupId>org.springframework.cloud</groupId>
3   <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
4 </dependency>
5
6 <dependency>
7   <groupId>org.springframework.boot</groupId>
8   <artifactId>spring-boot-starter-web</artifactId>
9 </dependency>Copy to clipboardErrorCopied
```

- 在启动类上添加@EnableDiscoveryClient注解表明是一个Eureka客户端

```

1  @EnableDiscoveryClient
2  @SpringBootApplication
3  public class EurekaClientApplication {
4
5      public static void main(String[] args) {
6          SpringApplication.run(EurekaClientApplication.class, args);
7      }
8
9  }Copy to clipboardErrorCopied

```

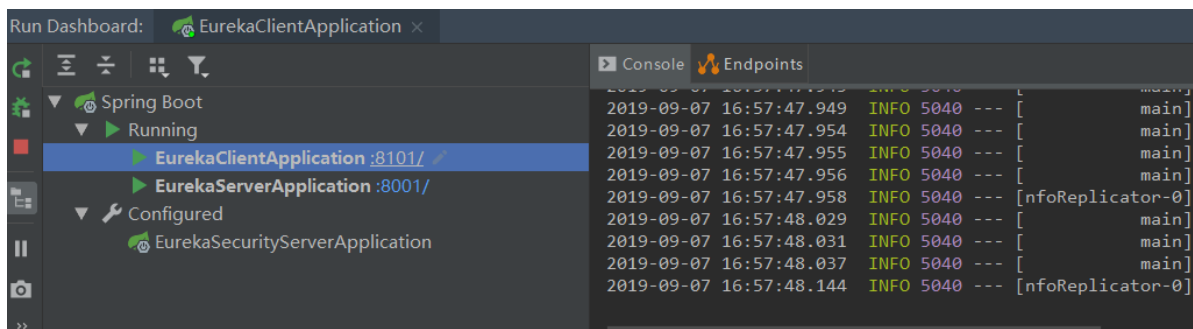
- 在配置文件application.yml中添加Eureka客户端的配置

```

1  server:
2      port: 8101 #运行端口号
3  spring:
4      application:
5          name: eureka-client #服务名称
6  eureka:
7      client:
8          register-with-eureka: true #注册到Eureka的注册中心
9          fetch-registry: true #获取注册实例列表
10         service-url:
11             defaultZone: http://localhost:8001/eureka/ #配置注册中心地址Copy to
              clipboardErrorCopied

```

- 运行eureka-client



- 查看注册中心 <http://localhost:8001/> 发现Eureka客户端已经成功注册

HOME
LAST 1000 SINCE STARTUP

### System Status

Environment	test	Current time	2019-09-07T16:58:28 +0800
Data center	default	Uptime	00:12
		Lease expiration enabled	true
		Renews threshold	3
		Renews (last min)	0

RENEWALS ARE LESSER THAN THE THRESHOLD. THE SELF PRESERVATION MODE IS TURNED OFF. THIS MAY NOT PROTECT INSTANCE EXPIRY IN CASE OF NETWORK/OTHER PROBLEMS.

### DS Replicas

### Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
EUREKA-CLIENT	n/a (1)	(1)	UP (1) - 192.168.56.1:eureka-client:8101

# 搭建Eureka注册中心集群

## 搭建两个注册中心

由于所有服务都会注册到注册中心去，服务之间的调用都是通过从注册中心获取的服务列表来调用，注册中心一旦宕机，所有服务调用都会出现问题。所以我们需要多个注册中心组成集群来提供服务，下面将搭建一个双节点的注册中心集群。

- 给eureka-sever添加配置文件application-replica1.yml配置第一个注册中心

```
1  server:
2    port: 8002
3  spring:
4    application:
5      name: eureka-server
6  eureka:
7    instance:
8      hostname: replica1
9    client:
10     serviceUrl:
11       defaultZone: http://replica2:8003/eureka/ #注册到另一个Eureka注册中心
12     fetch-registry: true
13     register-with-eureka: trueCopy to clipboardErrorCopied
```

- 给eureka-sever添加配置文件application-replica2.yml配置第二个注册中心

```
1  server:
2    port: 8003
3  spring:
4    application:
5      name: eureka-server
6  eureka:
7    instance:
8      hostname: replica2
9    client:
10     serviceUrl:
11       defaultZone: http://replica1:8002/eureka/ #注册到另一个Eureka注册中心
12     fetch-registry: true
13     register-with-eureka: trueCopy to clipboardErrorCopied
```

这里我们通过两个注册中心互相注册，搭建了注册中心的双节点集群，由于defaultZone使用了域名，所以还需在本机的host文件中配置一下。

- 修改本地host文件

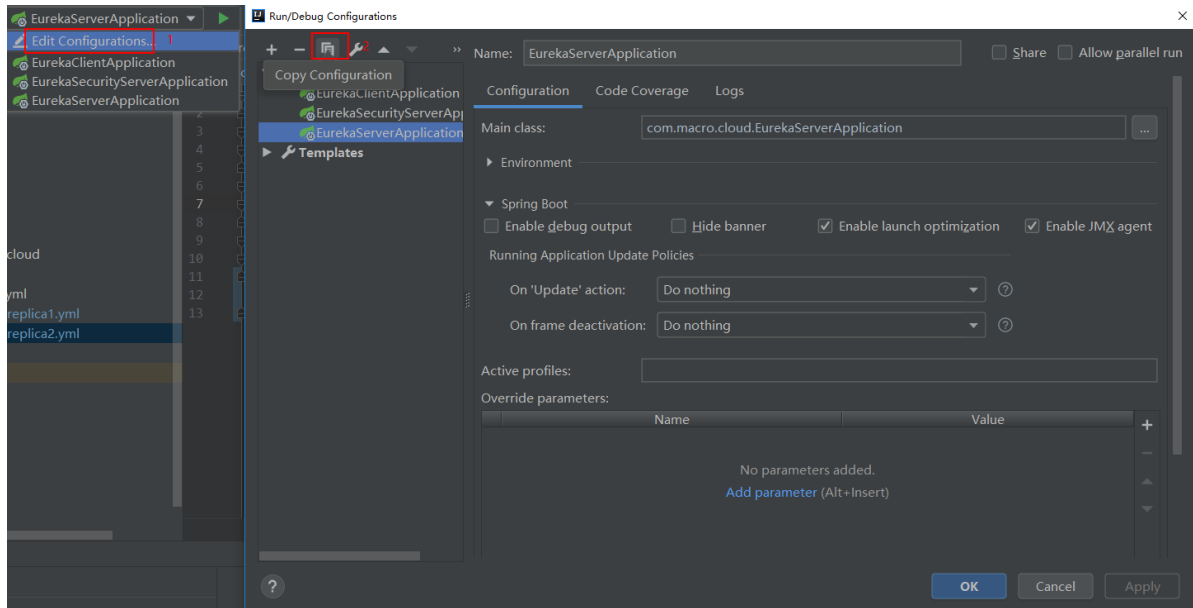
```
1  127.0.0.1 replica1
2  127.0.0.1 replica2Copy to clipboardErrorCopied
```

## 运行Eureka注册中心集群

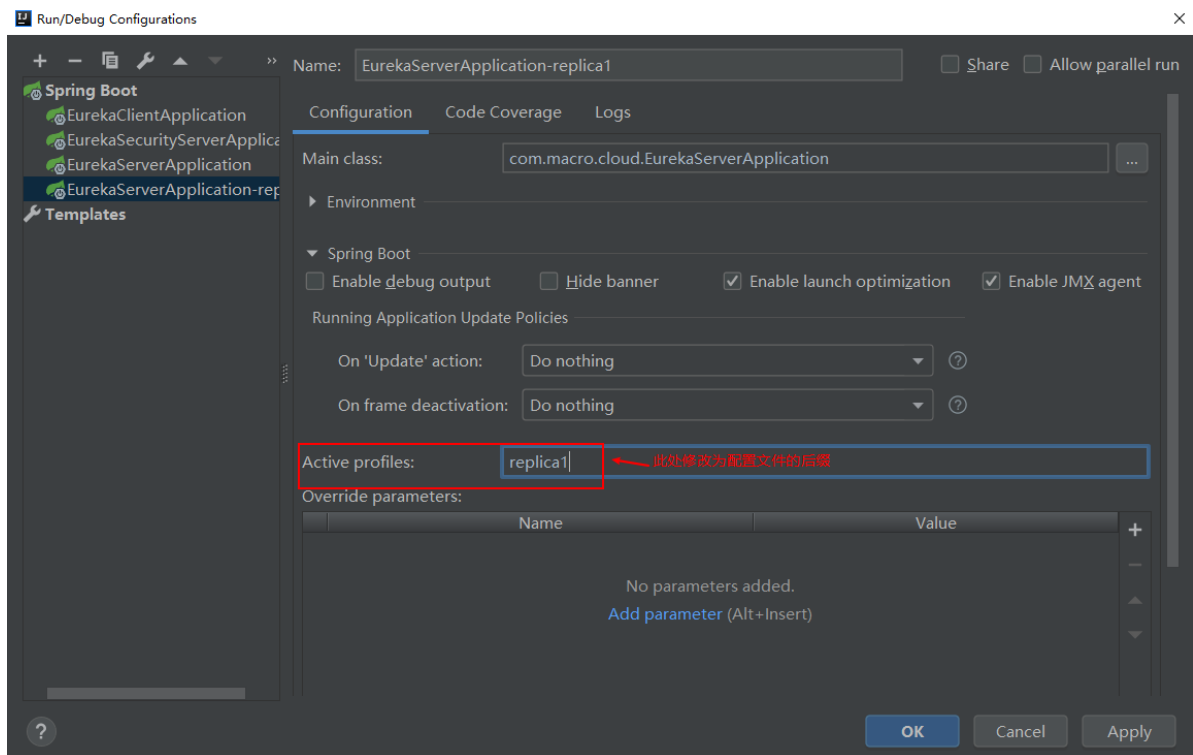
在IDEA中我们可以通过使用不同的配置文件来启动同一个SpringBoot应用。

- 添加两个配置，分别以application-replica1.yml和application-replica2.yml来启动eureka-server

## 从原启动配置中复制一个出来



## 配置启动的配置文件



- 启动两个eureka-server，访问其中一个注册中心 <http://replica1:8002/> 发现另一个已经成为其备份



## System Status

Environment	test	Current time	2019-09-07T20:52:14 +0800
Data center	default	Uptime	00:01
		Lease expiration enabled	true
		Renews threshold	3
		Renews (last min)	0

THE SELF PRESERVATION MODE IS TURNED OFF. THIS MAY NOT PROTECT INSTANCE EXPIRY IN CASE OF NETWORK/OTHER PROBLEMS.

## DS Replicas

replica2

## Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
EUREKA-SERVER	n/a (2)	(2)	UP (2) - 192.168.56.1:eureka-server:8003 , 192.168.56.1:eureka-server:8002

- 修改Eureka-client, 让其连接到集群

添加eureka-client的配置文件application-replica.yml, 让其同时注册到两个注册中心。

```
1  server:
2    port: 8102
3  spring:
4    application:
5      name: eureka-client
6  eureka:
7    client:
8      register-with-eureka: true
9      fetch-registry: true
10     service-url:
11       defaultZone: http://replica1:8002/eureka/,http://replica2:8003/eureka/ #同时
      注册到两个注册中心Copy to clipboardErrorCopied
```

以该配置文件启动后访问任意一个注册中心节点都可以看到eureka-client

## Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
EUREKA-CLIENT	n/a (1)	(1)	UP (1) - 192.168.56.1:eureka-client:8102
EUREKA-SERVER	n/a (2)	(2)	UP (2) - 192.168.56.1:eureka-server:8003 , 192.168.56.1:eureka-server:8002

## 给Eureka注册中心添加认证

## 创建一个eureka-security-server模块，在pom.xml中添加以下依赖

需要添加SpringSecurity模块。

```
1 <dependency>
2     <groupId>org.springframework.cloud</groupId>
3     <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
4 </dependency>
5
6 <dependency>
7     <groupId>org.springframework.boot</groupId>
8     <artifactId>spring-boot-starter-security</artifactId>
9 </dependency>Copy to clipboardErrorCopied
```

## 添加application.yml配置文件

主要是配置了登录注册中心的用户名和密码。

```
1 server:
2     port: 8004
3 spring:
4     application:
5         name: eureka-security-server
6     security: #配置SpringSecurity登录用户名和密码
7         user:
8             name: macro
9             password: 123456
10 eureka:
11     instance:
12         hostname: localhost
13     client:
14         fetch-registry: false
15         register-with-eureka: falseCopy to clipboardErrorCopied
```

## 添加Java配置WebSecurityConfig

默认情况下添加SpringSecurity依赖的应用每个请求都需要添加CSRF token才能访问，Eureka客户端注册时并不会添加，所以需要配置/eureka/\*\*路径不需要CSRF token。

```
1 @EnableWebSecurity
2 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
3
4     @Override
5     protected void configure(HttpSecurity http) throws Exception {
6         http.csrf().ignoringAntMatchers("/eureka/**");
7         super.configure(http);
8     }
9 }Copy to clipboardErrorCopied
```

运行eureka-security-server, 访问 <http://localhost:8004> 发现需要登录认证

## Please sign in

Sign in

## eureka-client注册到有登录认证的注册中心

- 配置文件中需要修改注册中心地址格式

```
1 http://${username}:${password}@${hostname}:${port}/eureka/ Copy to clipboardErrorCopied
```

- 添加application-security.yml配置文件, 按格式修改用户名和密码

```
1 server:
2   port: 8103
3 spring:
4   application:
5     name: eureka-client
6 eureka:
7   client:
8     register-with-eureka: true
9     fetch-registry: true
10    service-url:
11      defaultZone: http://macro:123456@localhost:8004/eureka/ Copy to clipboardErrorCopied
```

- 以application-security.yml配置运行eureka-client, 可以在注册中心界面看到eureka-client已经成功注册

### Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
EUREKA-CLIENT	n/a (1)	(1)	UP (1) - 192.168.56.1:eureka-client:8103

## Eureka的常用配置

```
1 eureka:
2   client: #eureka客户端配置
3     register-with-eureka: true #是否将自己注册到eureka服务端上去
4     fetch-registry: true #是否获取eureka服务端上注册的服务列表
5     service-url:
6       defaultZone: http://localhost:8001/eureka/ # 指定注册中心地址
```

```
7     enabled: true # 启用eureka客户端
8     registry-fetch-interval-seconds: 30 #定义去eureka服务端获取服务列表的时间间隔
9 instance: #eureka客户端实例配置
10    lease-renewal-interval-in-seconds: 30 #定义服务多久去注册中心续约
11    lease-expiration-duration-in-seconds: 90 #定义服务多久不去续约认为服务失效
12    metadata-map:
13      zone: jiangsu #所在区域
14    hostname: localhost #服务主机名称
15    prefer-ip-address: false #是否优先使用ip来作为主机名
16 server: #eureka服务端配置
17    enable-self-preservation: false #关闭eureka服务端的保护机制Copy to
clipboardErrorCopied
```

## 使用到的模块

---

```
1 springcloud-learning
2 |—— eureka-server -- eureka注册中心
3 |—— eureka-security-server -- 带登录认证的eureka注册中心
4 |—— eureka-client -- eureka客户端
```