**SpringCloud微服务介绍**

编写人员：修志明

编写日期：2017-10-27

**变更日志**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **变更内容** | **修订人** | **修订日期** |
| 1 | 文档编写 | 修志明 | 2017-10-27 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[一、 微服务简介](#_Toc14623)

[二、 SpringCloud解决方案](#_Toc1796)

[三、 SpringCloud服务介绍](#_Toc11956)

[3.1、服务的注册与发现（Eureka）](#_Toc32312)

[3.2、服务消费者[代理、集群]（ribbon、feign）](#_Toc3042)

[3.3、熔断、容错（Hystrix）](#_Toc15985)

[3.4、路由网关(zuul)](#_Toc25484)

[3.5、分布式配置中心(Spring Cloud Config+bus)](#_Toc3986)

[3.6、服务链路追踪(Spring Cloud Sleuth)](#_Toc1133)

[四、附录A](#_Toc6723)

[4.1、参考文档](#_Toc7681)

[4.2、示例工程清单](#_Toc7100)

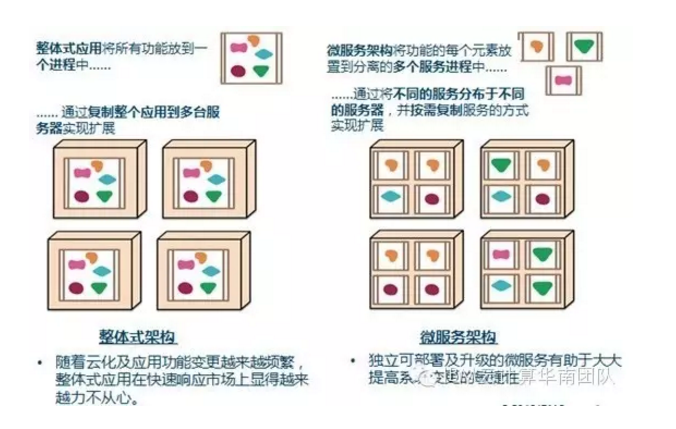
# 微服务简介

微服务是一种架构风格，一个大型复杂软件应用由一个或多个微服务组成。系统中的各个微服务可被独立部署，各个微服务之间是松耦合的。每个微服务仅关注于完成一件任务并很好地完成该任务。在所有情况下，每个任务代表着一个小的业务能力。

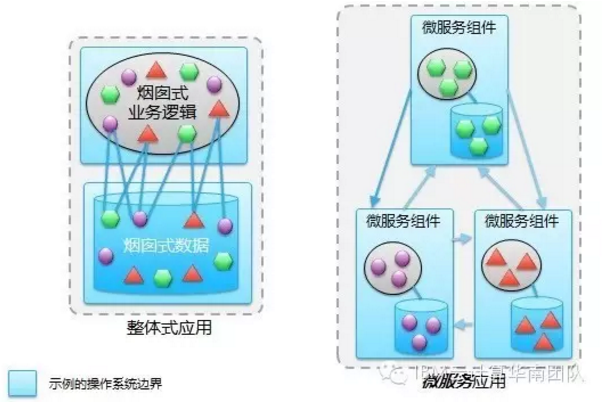
微服务的概念源于2014年3月Martin Fowler所写的一篇文章“Microservices”(<http://martinfowler.com/articles/microservices.html>)。

    尽管“微服务”这种架构风格没有精确的定义，但其具有一些共同的特性，如围绕业务能力组织服务、自动化部署、智能端点、对语言及数据的“去集中化”控制等等。

   微服务架构的思考是从与整体应用对比而产生的。



其中，对应用组件封装的方式是整体架构与微服务架构的主要差异，微服务架构将相关联的业务逻辑及数据放在一起形成独立的边界，其目的是能在不影响其他应用组件(微服务)的情况下更快地交付并推出市场。



优点：

* 每个服务都比较简单，只关注于一个业务功能。
* 微服务架构方式是松耦合的，可以提供更高的灵活性。
* 微服务可通过最佳及最合适的不同的编程语言与工具进行开发，能够做到有的放矢地解决针对性问题。
* 每个微服务可由不同团队独立开发，互不影响，加快推出市场的速度。
* 微服务架构是持续交付(CD)的巨大推动力，允许在频繁发布不同服务的同时保持系统其他部分的可用性和稳定性。

缺点：

* 运维开销及成本增加。
* 必须有坚实的开发运维一体化技能。
* 隐式接口及接口匹配问题。
* 代码重复，某些底层功能需要被多个服务所用。
* 分布式系统的复杂性。
* 可测性的挑战。

# SpringCloud解决方案

Spring Cloud为开发人员提供了快速构建分布式系统中一些常见模式的工具（例如配置管理，服务发现，断路器，智能路由，微代理，控制总线）。分布式系统的协调导致了样板模式, 使用Spring Cloud开发人员可以快速地支持实现这些模式的服务和应用程序。他们将在任何分布式环境中运行良好，包括开发人员自己的笔记本电脑，裸机数据中心，以及Cloud Foundry等托管平台。

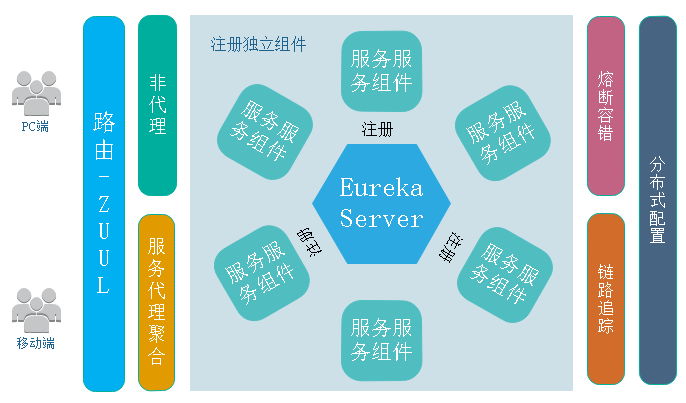


图2-1 Spring Cloud服务

**特性**

Spring Cloud专注于提供良好的开箱即用经验的典型用例和可扩展性机制覆盖。

* 分布式/版本化配置
* 服务注册和发现
* 路由
* service - to - service调用
* 负载均衡
* 断路器
* 分布式消息传递

# SpringCloud服务介绍

SpringCloud主要从“服务注册发现”、“熔断、容错处理”、“分布式配置”’、“链路追踪”、“代理、集群、扩展”、“路由”等方面提供微服务。

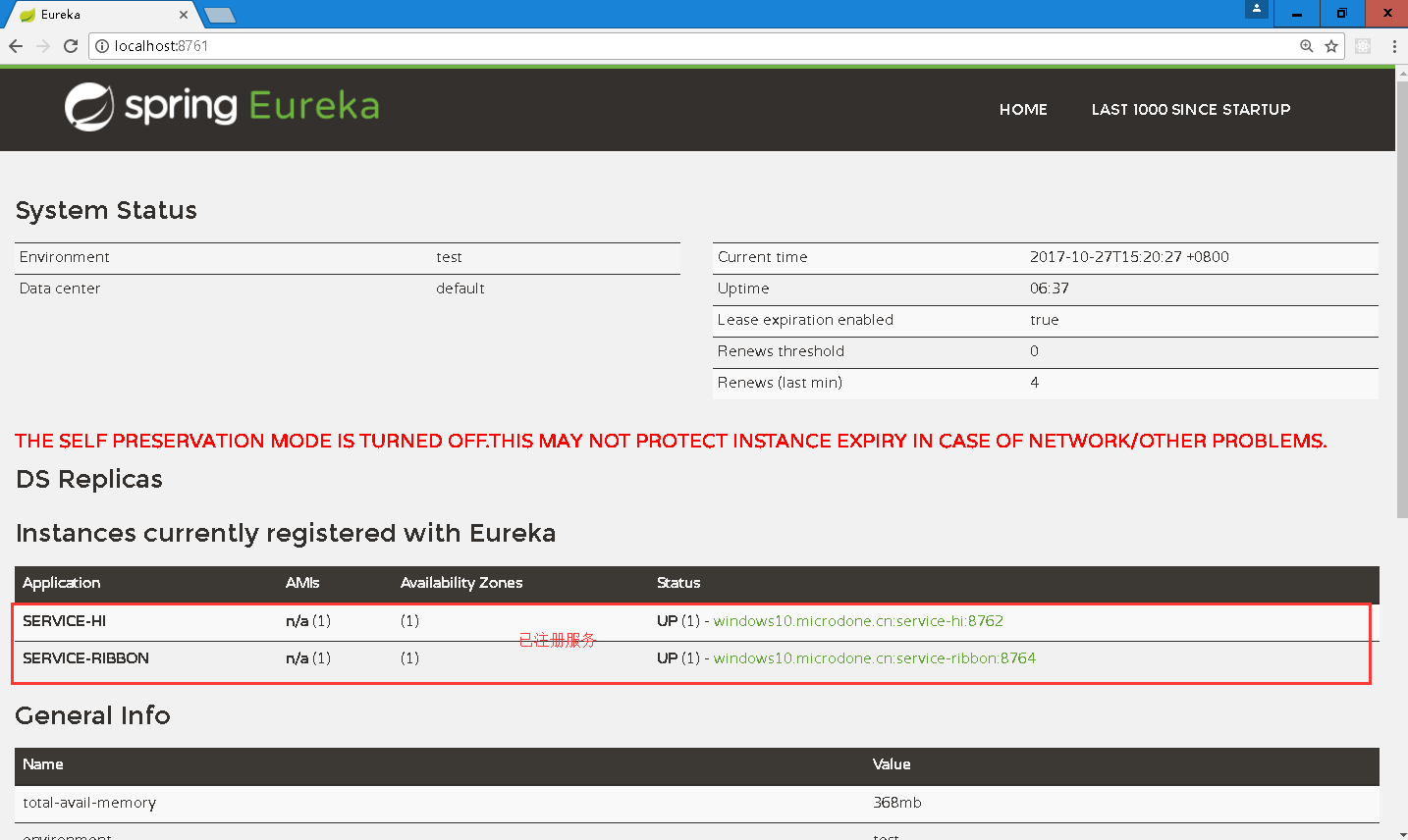


图3-1 Spring Cloud 基础服务

## 3.1、[服务的注册与发现（Eureka）](http://blog.csdn.net/forezp/article/details/69696915" \t "http://blog.csdn.net/forezp/article/details/_blank)

云端服务发现，一个基于 REST 的服务，用于定位服务，以实现云端中间层服务发现和故障转移；SpringCloud中文官网提供类Eureka、Consul、SpringCloud Zookeeper三种解决方案。

以下是Eureka管理界面：界面中红色部分为已经注册到该Server的服务



图：3-1-1 Eureka服务管理界面

## 3.2、[服务消费者[代理、集群]（ribbon、feign）](http://blog.csdn.net/forezp/article/details/69788938" \t "http://blog.csdn.net/forezp/article/details/_blank)

在微服务架构中，业务都会被拆分成一个独立的服务，服务与服务的通讯是基于http restful的；通过服务消费者可以代理调用微服务，也可以聚合微服务操作。Spring cloud有两种服务调用方式，一种是ribbon+restTemplate，另一种是feign；

**1、Ribbon**

提供云端负载均衡，有多种负载均衡策略可供选择，可配合服务发现和断路器使用。



图3-2-1使用restTemplate调用注册服务

**2、Feign**

Feign是一个声明式的伪Http客户端，它使得写Http客户端变得更简单。使用Feign，只需要创建一个接口并注解。它具有可插拔的注解特性，可使用Feign 注解和JAX-RS注解。Feign支持可插拔的编码器和解码器。Feign默认集成了Ribbon，并和Eureka结合，默认实现了负载均衡的效果

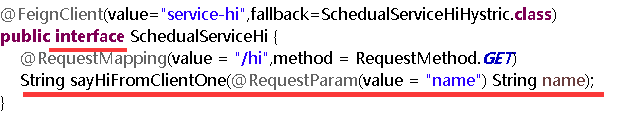


图3-2-2 以接口方式调用注册服务

## 3.3、[熔断、容错（Hystrix）](http://blog.csdn.net/forezp/article/details/69934399" \t "http://blog.csdn.net/forezp/article/details/_blank)

熔断器，容错管理工具，旨在通过熔断机制控制服务和第三方库的节点,从而对延迟和故障提供更强大的容错能力； 在微服务架构中，一个请求需要调用多个服务是非常常见的，较底层的服务如果出现故障，会导致连锁故障。当对特定的服务的调用的不可用达到一个阀值（Hystric 是5秒20次） 断路器将会被打开，断路打开后，可用避免连锁故障，通过fallback方法可以直接返回一个固定值或固定的处理逻辑。

## 3.4、[路由网关(zuul)](http://blog.csdn.net/forezp/article/details/69939114" \t "http://blog.csdn.net/forezp/article/details/_blank)

Zuul 是在云平台上提供动态路由,监控,弹性,安全等边缘服务的框架。Zuul 相当于是设备和 Web 网站后端所有请求的前门；Zuul的主要功能是路由转发和过滤器。

**主要功能：**

* 认证
* 压力测试
* 金丝雀测试
* 动态路由
* 负载削减
* 安全
* 静态响应处理
* 主动/主动交换管理

## 3.5、[分布式配置中心(Spring Cloud Config+bus)](http://blog.csdn.net/forezp/article/details/70037291" \t "http://blog.csdn.net/forezp/article/details/_blank)

让你可以把配置放到远程服务器，集中化管理集群配置，目前支持本地存储、Git以及Subversion。

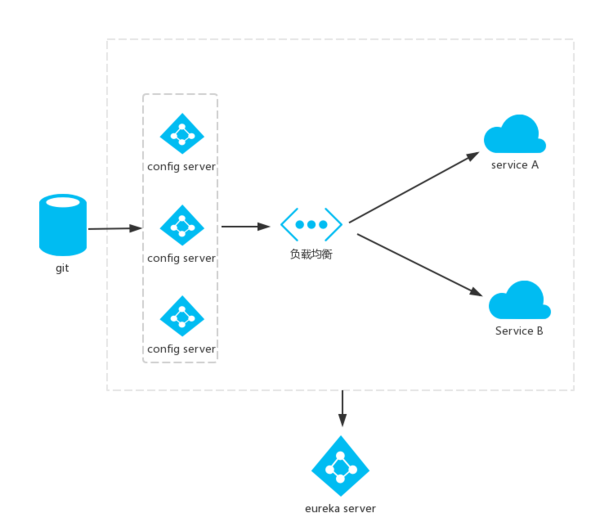


图3-5-1 Spring Cloud Config结构图

**方式1：**只使用SpringCloudConfig服务

使用该种方式，系统支持启动时从资源库（SVN,GIT）加载统一的配置文件，如果应用运行过程中需要同步配置文件，需要逐个应用手动调用/refresh刷新应用中的配置。

**方式2：**方式1+bus+rabbitMQ

使用Bus消息服务时需要依赖RabbitMQ等消息队列服务；使用该种方式只需要任意接入应用，调用/bus/refresh同步服务，既可以刷新所有应用的配置信息，也可以指定刷新指定服务的配置（Eureka Server中注册的服务）；结合GIT的Webhooks服务可以实现自动刷新。

## 3.6、[服务链路追踪(Spring Cloud Sleuth)](http://blog.csdn.net/forezp/article/details/70162074" \t "http://blog.csdn.net/forezp/article/details/_blank)

微服务架构上通过业务来划分服务的，通过REST调用，对外暴露的一个接口，可能需要很多个服务协同才能完成这个接口功能，如果链路上任何一个服务出现问题或者网络超时，都会形成导致接口调用失败。随着业务的不断扩张，服务之间互相调用会越来越复杂。

Spring Cloud Sleuth 封装了Dapper和log-based追踪以及Zipkin和HTrace操作，为SpringCloud应用实现了一种分布式追踪解决方案；通过该服务可以追踪微服务集群中各个服务间的依赖关系，跟踪集群中服务请求信息。

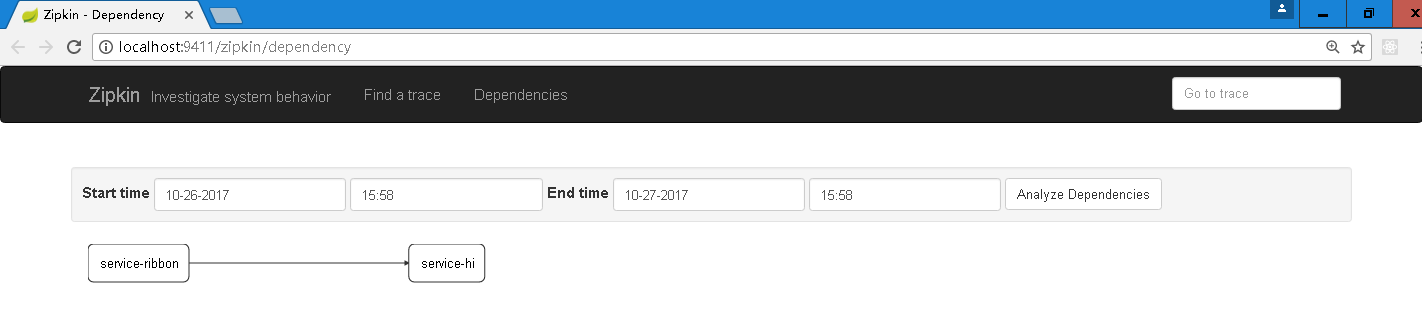




图3-6-1 链路追踪（service-ribbon 调用 service-hi服务）

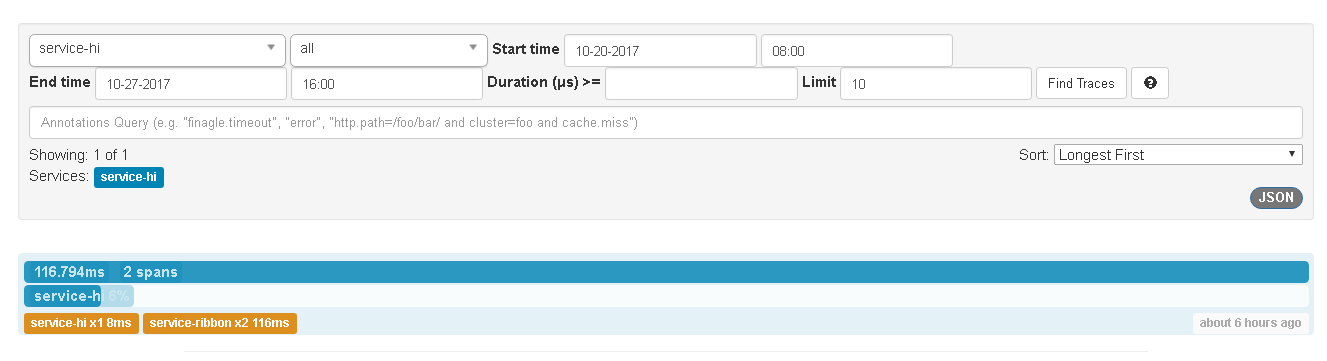


图3-6-2 请求跟踪页面

# 四、附录A

## 4.1、参考文档

示例搭建参考文档：<http://blog.csdn.net/forezp/article/details/70148833>

SpringCloud官网：<https://springcloud.cc/>

## 4.2、示例工程清单

以下是根据资料搭建对应DEMO示例清单；搭建服务时可以参考对应工程下的“说明.txt”文件运行示例。

|  |  |  |  |
| --- | --- | --- | --- |
| **示例代码清单** | | | |
| **序号** | **名称** | **SVN目录** | **端口** |
| SVN地址：svn://web.ccyonyou.com/ccyonyou/MicroService/spring-cloud | | | |
| 1 | 服务注册中心 | /demo/springcloud-eureka-server | 8761 |
| 2 | 服务提供者 | /demo/springcloud-eureka-client | 8762~8763 |
| 3 | 服务消费者(ribbon) | /demo/springcloud-eureka-ribbon | 8764 |
| 4 | 服务消费者(feign)-推荐 | /demo/springcloud-eureka-feign | 8765 |
| 5 | 路由网关 | /demo/springcloud-eureka-zuul | 8769 |
| 6 | 分布式配置中心 | /demo/springcloud-config-server | 8888 |
| 7 | 分布式配置文件 | /demo/springcloud-config-file | - |
| 8 | 分布式配置文件示例（可从配置中心同步） | /demo/springcloud-config-client | 8881~8882 |
| 9 | 链路追踪服务 | /demo/springcloud-zipkin-server | 9411 |
| **文档** | | | |
| **序号** | **名称** | | |
| 1 | SpringCloud微服务介绍文档V1.0.docx**（本文档）** | | |

# 五、YYRNDealer微服务环境搭建

## 5.1、工程清单

|  |  |  |  |
| --- | --- | --- | --- |
| **示例代码清单** | | | |
| **序号** | **名称** | **SVN目录** | **端口** |
| SVN地址：svn://web.ccyonyou.com/ccyonyou/YYRNDealer | | | |
| 1 | 服务注册中心 | /rndealer -eureka-server | 12050 |
| 2 | 路由网关 | /rndealer -eureka-zuul | 12051 |
| 3 | 服务消费者(ribbon) | /rndealer -eureka-ribbon | 12052 |
| 4 | 服务消费者(feign)-推荐 | /rndealer -eureka-feign | 12053 |
| 5 | 服务提供者(用户权限) | /rndealer-microservice-user | 12060 |
| 6 | 服务提供者(线索服务) | /rndealer-microservice-clue | 12061 |
| 7 | 服务提供者(客户服务) | /rndealer-microservice-customer | 12062 |
| 8 | 服务提供者(订单服务) | /rndealer-microservice-order | 12063 |
| 9 | 服务提供者(库存服务) | /rndealer-microservice-stock | 12064 |
| 10 | 服务提供者(车辆服务) | /rndealer-microservice-vehicle | 12065 |
| 11 | 服务提供者(其他服务) | /rndealer-microservice-other | 12066 |
| 12 | 服务提供者(组织机构) | /rndealer-microservice-org | 12067 |
| 13 | 服务提供者(日志服务) | /rndealer-microservice-log | 12068 |
| 14 | 分布式配置中心 | /rndealer -config-server | 12054 |
| 15 | 分布式配置文件 | /rndealer -config-file | - |
| 16 | 分布式配置文件示例（可从配置中心同步） | /rndealer -config-client | 13060-13069 |
| 17 | 链路追踪服务 | /rndealer-zipkin-server | 12055 |