Spring Cloud

(Finchley.RELEASE)

目录

[Spring Cloud是什么？ 3](#_Toc520908188)

[Spring Cloud有那些工具框架？ 3](#_Toc520908189)

# 1简介

## 1.1 Spring Cloud是什么？

1. Spring Cloud 是基于Spring Boot来实现的。
2. Spring Cloud 是一系列工具框架的集合体。
3. Spring Cloud 提供了一套完整的微服务解决方案。

## 1.2 Spring Cloud有那些工具框架？

1. Spring Cloud Config 配置中心，利用git集中管理程序的配置。
2. Spring Cloud Netflix 集成众多Netflix的开源软件。
3. Spring Cloud Bus 消息总线，利用分布式消息将服务和服务实例连接在一起，用于在一个集群中传播状态的变化。
4. Spring Cloud for Cloud Foundry 利用Pivotal Cloudfoundry集成你的应用程序。
5. Spring Cloud Cloud Foundry Service Broker 为建立管理云托管服务的服务代理提供一个起点。
6. Spring Cloud Cluster 基于Zookeeper，Redis，Hazelcast，Consul实现的领导选举和平明状态模式的抽象和实现。
7. Spring Cloud Consul 基于Hashicorp Consul实现的服务发现和配置管理。
8. Spring Cloud Security 在Zuul代理中为OAuth2 rest客户端和认证头转发提供负载均衡。
9. Spring Cloud Sleuth SpringCloud应用的分布式追踪系统，和Zipkin，HTrace，ELK兼容。
10. Spring Cloud Data Flow 一个云本地程序和操作模型，组成数据微服务在一个机构化的平台上。
11. Spring Cloud Stream 基于Redis，Rabbit，Kafka实现的消息微服务，简单声明模型用以在Spring Cloud应用中收发消息。
12. Spring Cloud Stream App Starters 基于Spring Boot为外部系统提供Spring的集成。
13. Spring Cloud Task短生命周期的微服务，为Spring Boot应用简单声明添加功能和非功能特性。
14. Spring Cloud Task App Starters
15. Spring Cloud Zookeeper 服务发现和配置管理基于Apache Zookeeper。
16. Spring Cloud for Amazon Web Service 快速和亚马逊网络服务集成。
17. Spring Cloud Connectors 便于Paas 应用在各种平台上连接到后端像数据库和消息经纪服务。
18. Spring Cloud Starters （项目已经终止在Angel.SR2后的版本和其他项目合并）
19. Spring Cloud CLI 插件用Groovy快速的创建Spring Cloud组件应用。

# 2 服务治理

## 2.1 简介

服务治理是微服务架构中最为核心和基础的模块，它主要用来实现各个微服务实例的自动化注册和发现。

## 2.2 Eureka服务治理

### 2.2.1 简介

Eureka是Spring Cloud Netflix 微服务套件中的一部分。Spring Cloud Eureka使用Netflix Eureka来实现服务注册与服务发现。Spring Cloud Eureka包含了服务端组件、客户端组件，并且服务端与客户端均采用java编写，所以Eureka主要适用于通过java实现分布式系统，或是与JVM兼容语言构建的系统。Eureka的服务端提供了比较完善的REST API，所以Eureka也支持将非java语言实现的服务纳入到Eureka服务治理体系中来，只需要其他语言平台实现自己的Eureka客户端。目前.Net平台的Steeltoe、Node.js的eureka-js-client等都已经实现了各自平台的Ereka客户端组件。

### 2.2.2 Eureka 服务端

Eureka服务端，即为注册中心，用于提供服务的注册与发现。

Eureka与其他服务注册中心一样，支持高可用配置，当集群中有分片出现故障时，Eureka自动进入保护模式。它允许在分片发生故障的时候继续提供服务的注册与发现，当故障恢复时，集群中的其他分片会把他们的状态再次同步回来。集群中的不同服务注册中心通过异步模式互相复制各自的状态，这也就意味着在给定的时间每个实例关于所有服务的状态可能存在不一致现象。

### 2.2.3 Eureka客户端

