

springcloud介绍

更多学习资料，一点课堂：www.yidiankt.com

一点课堂QQ群：984370849，QQ：2868289889

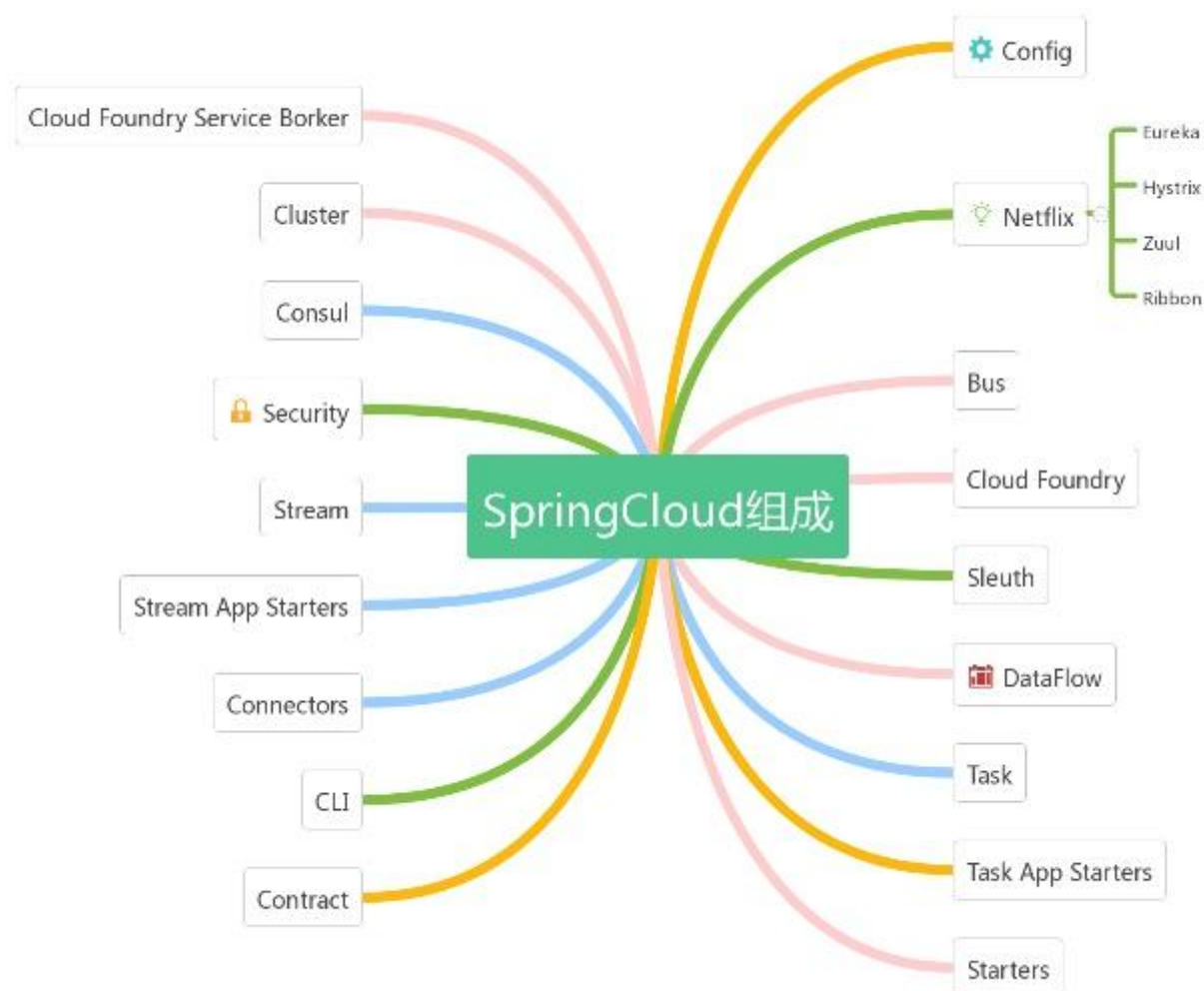
微信号：chengweixin9

免费公开课：<https://ke.qq.com/course/394307>

springcloud是微服务架构的集大成者，将一系列优秀的组件进行了整合。基于springboot构建，对我们熟悉spring的程序员来说，上手比较容易。

通过一些简单的注解，我们就可以快速的在应用中配置一下常用模块并构建庞大的分布式系统。

SpringCloud的组件相当繁杂，拥有诸多子项目。重点关注Netflix



<https://3dml.xiaohuoneng.com/V16732624778>

下面简单介绍下经常用的5个

服务发现——Netflix Eureka

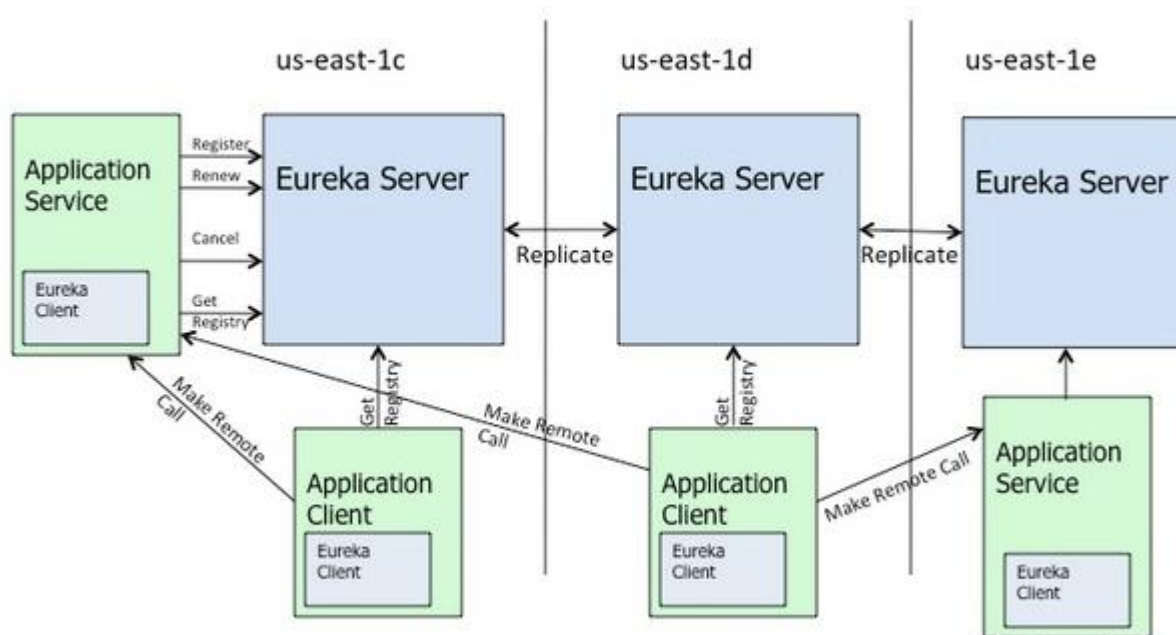
客户端负载均衡——Netflix Ribbon

断路器——Netflix Hystrix

服务网关——Netflix Zuul

分布式配置——Spring Cloud Config

Eureka



作用：实现服务治理（服务注册与发现）

简介：Spring Cloud Eureka是Spring Cloud Netflix项目下的服务治理模块。

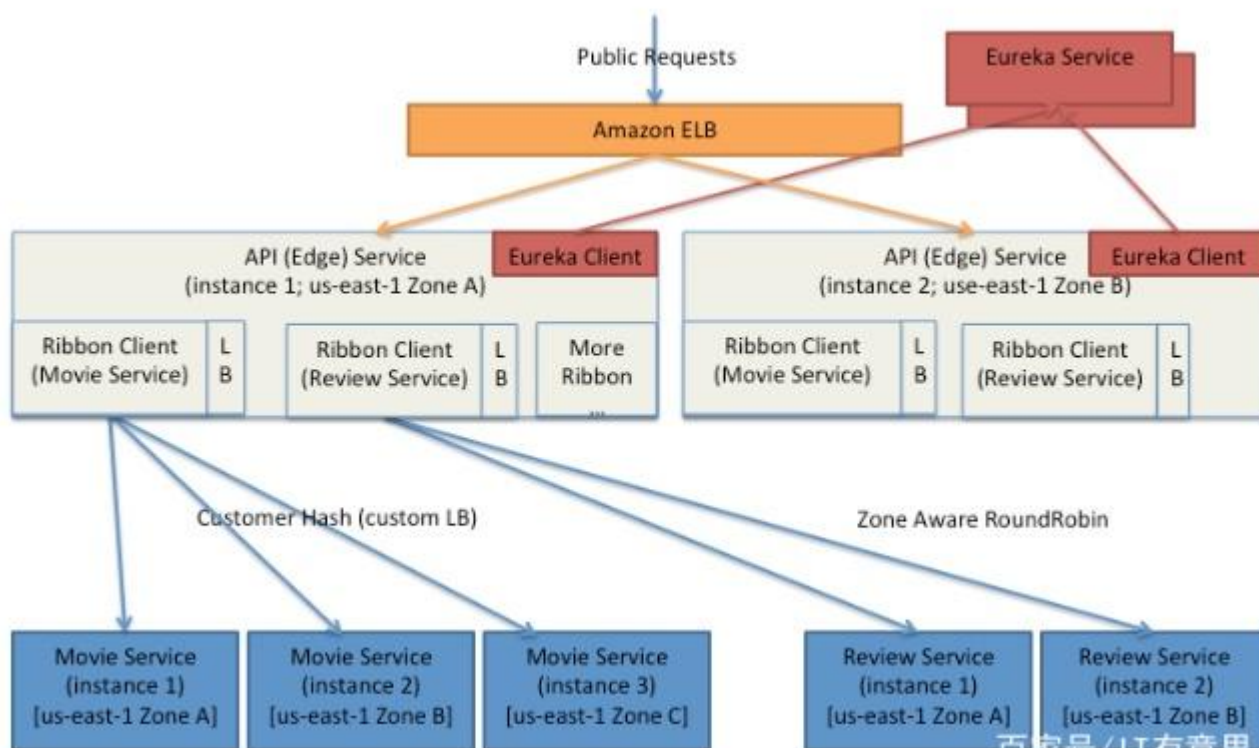
由两个组件组成：Eureka服务端和Eureka客户端。

Eureka服务端用作服务注册中心。支持集群部署。

Eureka客户端是一个java客户端，用来处理服务注册与发现。

在应用启动时，Eureka客户端向服务端注册自己的服务信息，同时将服务端的服务信息缓存到本地。客户端会和服务端周期性的进行心跳交互，以更新服务租约和服务信息。

Ribbon

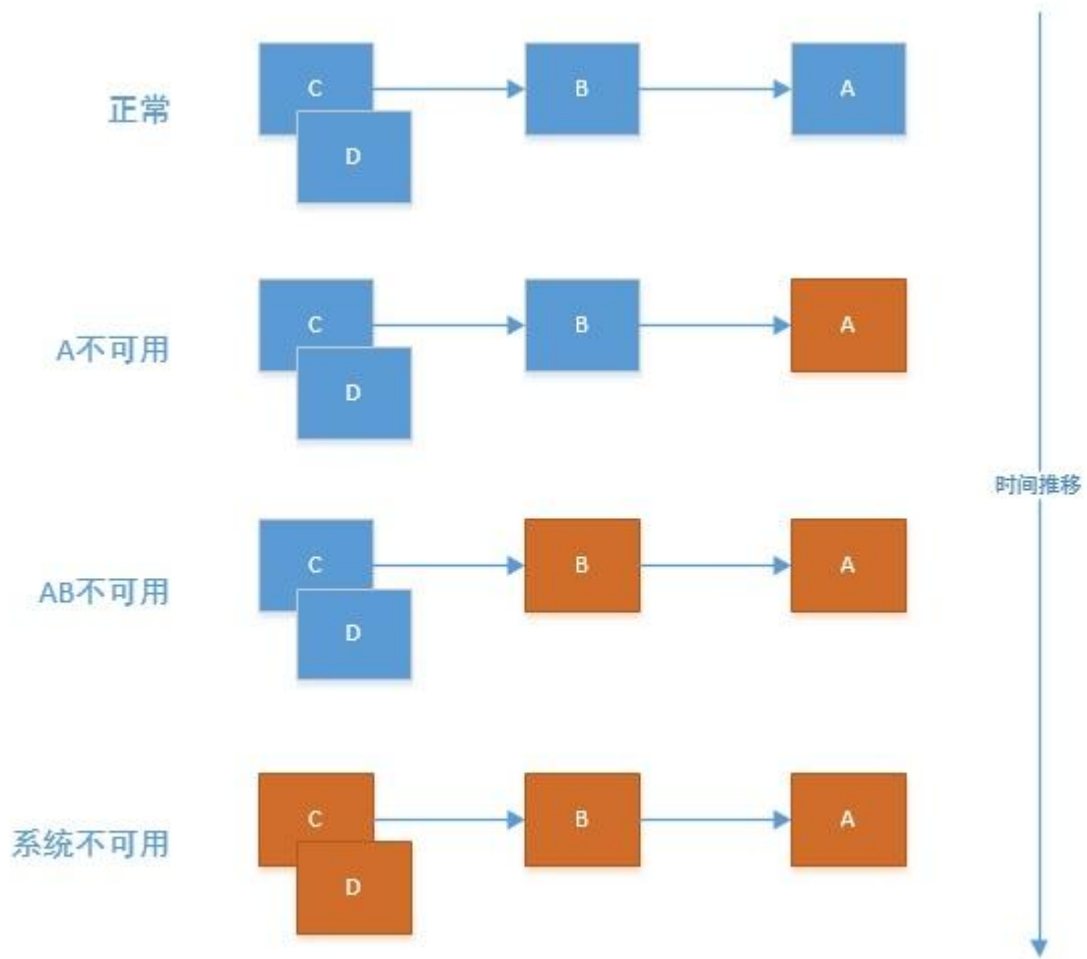


作用：Ribbon，主要提供客户侧的软件负载均衡算法。

简介：Spring Cloud Ribbon是一个基于HTTP和TCP的客户端负载均衡工具，它基于Netflix Ribbon实现。通过Spring Cloud的封装，可以让我们轻松地将面向服务的REST模版请求自动转换成客户端负载均衡的服务调用。

注意看上图，关键点就是将外界的rest调用，根据负载均衡策略转换为微服务调用。Ribbon有比较多的负载均衡策略，以后专门讲解。

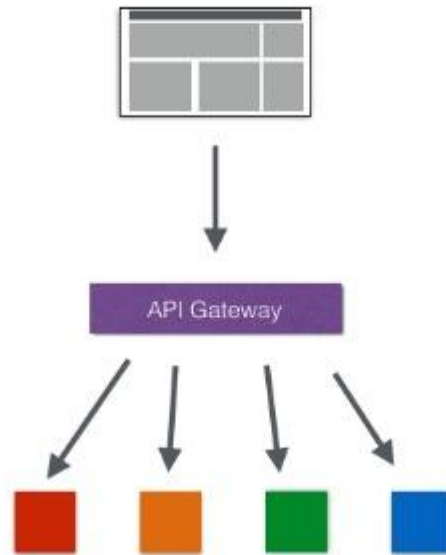
Hystrix



作用：断路器，保护系统，控制故障范围。

简介：为了保证其高可用，单个服务通常会集群部署。由于网络原因或者自身的原因，服务并不能保证100%可用，如果单个服务出现问题，调用这个服务就会出现线程阻塞，此时若有大量的请求涌入，Servlet容器的线程资源会被消耗完毕，导致服务瘫痪。服务与服务之间的依赖性，故障会传播，会对整个微服务系统造成灾难性的严重后果，这就是服务故障的“雪崩”效应。

Zuul

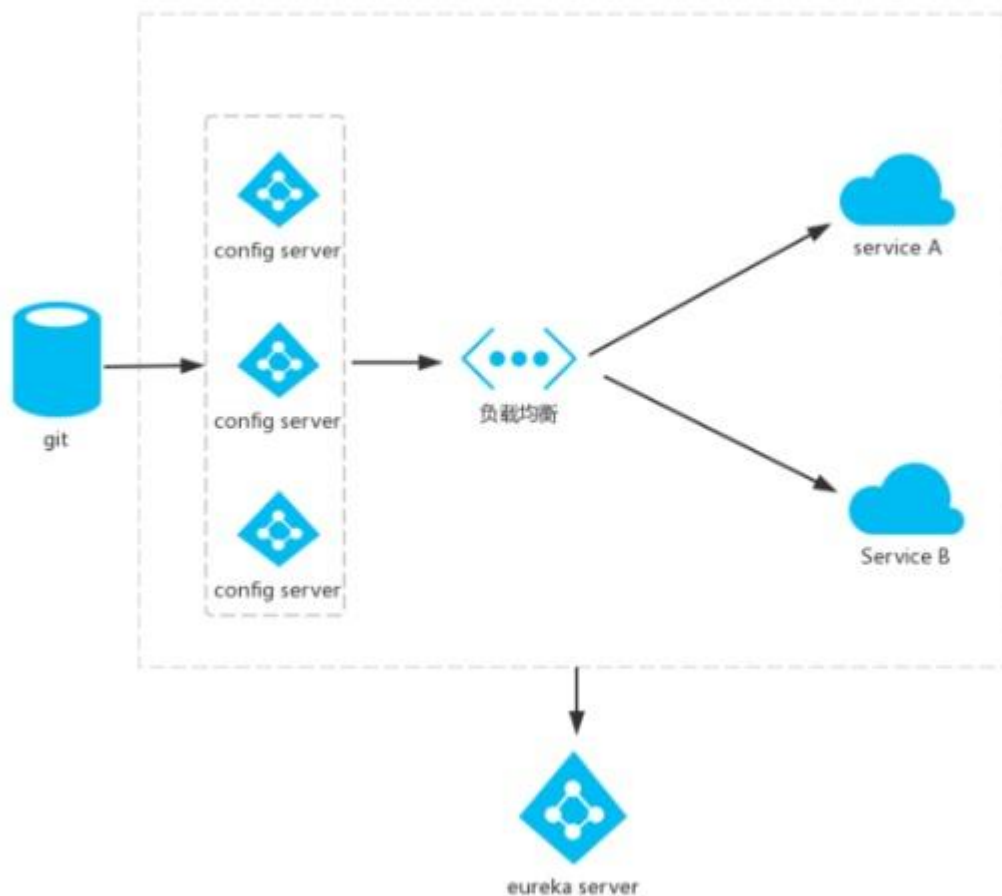


作用：api网关，路由，负载均衡等多种作用

简介：类似nginx，反向代理的功能，不过netflix自己增加了一些配合其他组件的特性。

在微服务架构中，后端服务往往不直接开放给调用端，而是通过一个API网关根据请求的url，路由到相应的服务。当添加API网关后，在第三方调用端和服务提供方之间就创建了一面墙，这面墙直接与调用方通信进行权限控制，后将请求均衡分发给后台服务端。

Config



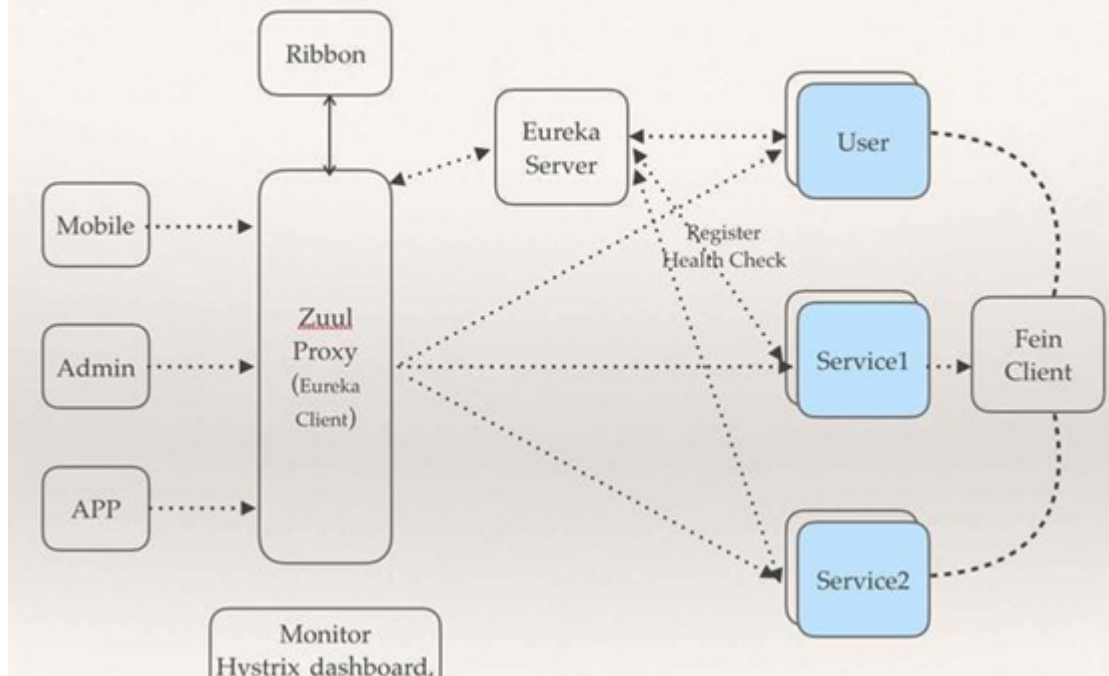
作用：配置管理

简介：SpringCloud Config提供服务器端和客户端。服务器存储后端的默认实现使用git，因此它轻松支持标签版本的配置环境，以及可以访问用于管理内容的各种工具。

这个还是静态的，得配合Spring Cloud Bus实现动态的配置更新。

相关组件架构图

Spring Cloud组件架构



从上图可以看出Spring Cloud各个组件相互配合，合作支持了一套完整的微服务架构。

- 其中Eureka负责服务的注册与发现，很好将各服务连接起来
- Hystrix 负责监控服务之间的调用情况，连续多次失败进行熔断保护。
- Hystrix dashboard,Turbine 负责监控 Hystrix的熔断情况，并给予图形化的展示
- Spring Cloud Config 提供了统一的配置中心服务
- 当配置文件发生变化的时候，Spring Cloud Bus 负责通知各服务去获取最新的配置信息
- 所有对外的请求和服务，我们都通过Zuul来进行转发，起到API网关的作用
- 最后我们使用Sleuth+Zipkin将所有的请求数据记录下来，方便我们进行后续分析

为什么使用SpringCloud?

Spring Cloud从设计之初就考虑了绝大多数互联网公司架构演化所需的功能，如服务发现注册、配置中心、消息总线、负载均衡、断路器、数据监控等。这些功能都是以插拔的形式提供出来，方便我们系统架构演进的过程中，可以合理的选择需要的组件进行集成，从而在架构演进的过程中会更加平滑、顺利。

微服务架构是一种趋势，Spring Cloud提供了标准化的、全站式的技术方案，意义可能会堪比当前Servlet规范的诞生，有效推进服务端软件系统技术水平的进步。

springcloud官网

Spring Cloud中文文档: <https://springcloud.cc/>

官网: <https://projects.spring.io/spring-cloud/>