

Spring Cloud 是一个基于 Spring Boot 实现的微服务架构开发工具。它为微服务架构中设计的配置管理、服务治理、断路器、智能路由、微代理、控制总线、全局锁、决策竞选、分布式会话和集群状态管理等操作提供了一种简单的开发方式。

Spring Cloud 包含了多个子项目：

- Spring Cloud Config：配置管理工具，支持使用 Git 存储配置内容，可以使用它实现应用配置的外部化存储，并支持客户端配置信息刷新、加密/解密配置内容等。
- Spring Cloud Netflix：核心组件，对多个 Netflix OSS 开源套件进行整合。
 - Eureka：服务治理组件，包含服务注册中心、服务注册与发现机制的实现。
 - Hystrix：容错管理组件，实现断路器模式，帮助服务依赖中出现延迟和为故障提供强大的容错能力。
 - Ribbon：客户端负载均衡的服务调用组件。
 - Feign：基于 Ribbon 和 Hystrix 的声明式服务调用组件。
 - Zuul：网关组件，提供智能路由、访问过滤等功能。
 - Archaius：外部化组件配置
- Spring Cloud Cluster：针对 ZooKeeper、Redis、Hazelcast、Consul 的选举算法和通用状态模式的实现。
- Spring Cloud Cloudfoundry：与 Pivotal Cloudfoundry 的整合支持。
- Spring Cloud Consul：服务发现与配置管理工具。
- Spring Cloud Stream：通过 Redis、Rabbit 或者 Kafka 实现的消费微服务，可以通过简单的声明式模型来发送和接收消息。
- Spring Cloud AWS：用于简化整合 Amazon Web Service 的组件。
- Spring Cloud Security：安全工具包，提供在 Zuul 代理中对 OAuth2 客户端请求的中继器。

- Spring Cloud Sleuth: Spring Cloud 应用的分布式跟踪实现, 可以完美整合 Zipkin。
- Spring Cloud Zookeeper: 基于 ZooKeeper 的服务发现与配置管理组件。
- Spring Cloud Starters: Spring Cloud 的基础组件, 它是基于 Spring boot 风格项目的基础依赖模块。
- Spring Cloud CLI: 用于在 Groovy 中快速创建 Spring Cloud 应用的 Spring Boot CLI 插件