# SpringCloud概述

## 为什么使用SpringCloud

Spring Cloud提供了一套完整的微服务整体解决方案。提供了分布式配置中心、分布式锁、分布式跟踪、分布式服务治理、分布式任务调度平台等模块解决对应的问题。

## 什么是SpringCloud

Spring Cloud是一个分布式的整体解决方案。Spring Cloud 为开发者提供了在分布式系统（配置管理，服务发现，熔断，路由，微代理，控制总线，一次性token，全局琐，leader选举，布式session，集群状态） 中快速构建的工具，使用Spring Cloud的开发者可以快速的启动服务或构建应用、同时能够快速和云平台资源进行对接。

**SpringCloud分布式常用组件**

服务发现一 Netflix Eureka

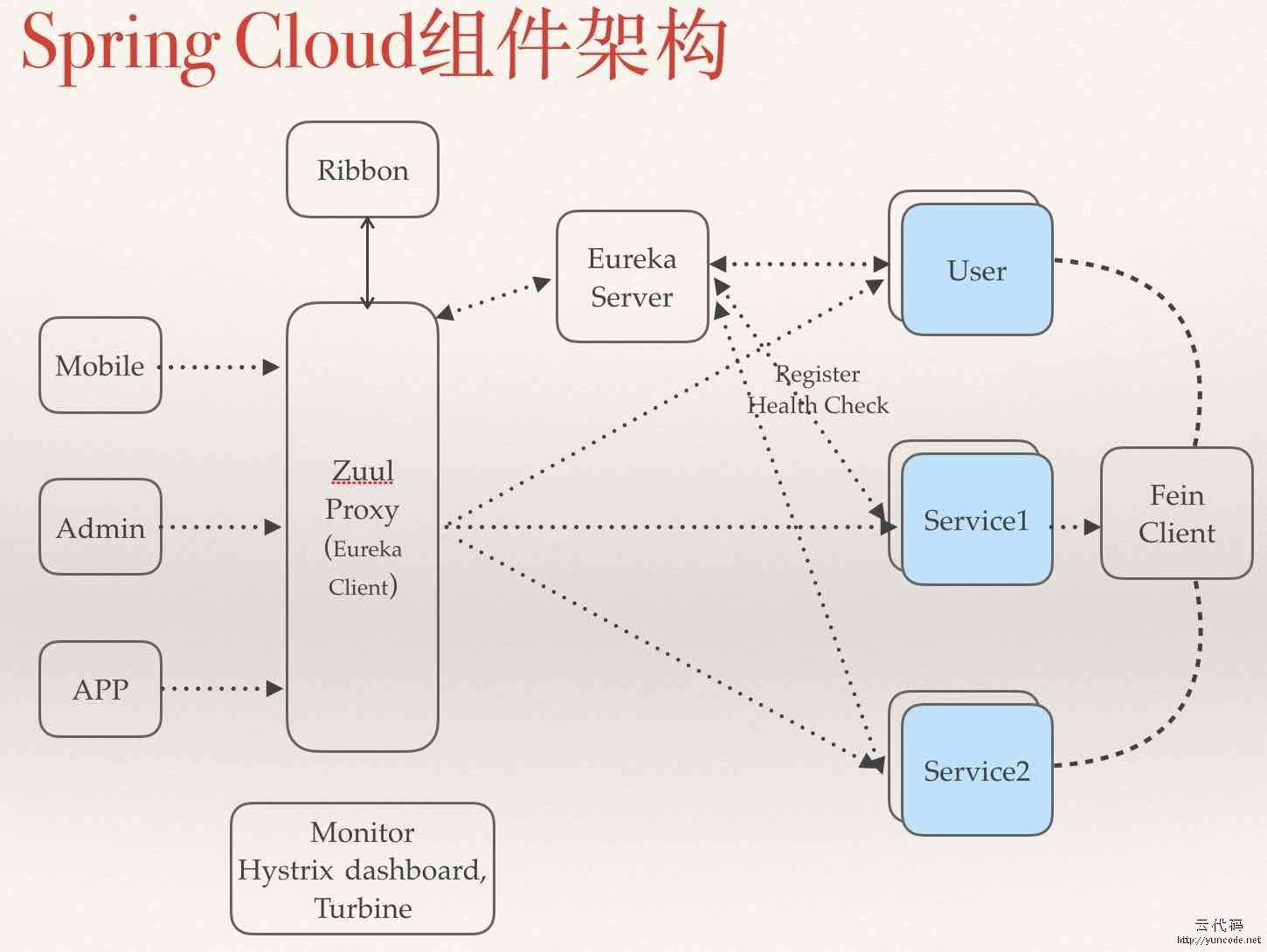
客服端负载均衡一Netflix Ribbon

断路器-Netflix Hystrix

服务网关-Netflix Zuul

分布式配置中心一Spring Cloud Config

服务调用组件—Feign



# 服务发现与注册

一个应用拆分为多个独立微服务。多个微服务可以使用Docker容器化进行多实例部署。

架构演化到这里遇到了一个难题，如果要查询用户所有的订单，用户服务可能会依赖订单服务，用户服务如何与订单服务交互呢？订单服务有多个实例该访问哪一个？服务注册和发现成为了一种很好的解决方案。

[微服务](https://so.csdn.net/so/search?q=%E5%BE%AE%E6%9C%8D%E5%8A%A1&spm=1001.2101.3001.7020)架构中最核心的部分是服务治理，服务治理最基础的组件是注册中心。随着微服务架构的发展，出现了很多微服务架构的解决方案，其中包括我们熟知的 Dubbo 和 Spring Cloud。

在SpringCloud 中提供了多种服务注册与发现组件：Eureka,Consul,Zookeeper。官方推荐使用Eureka。

在dubbo 支持了 Zookeeper、Redis、Multicast 和 Simple，官方推荐 Zookeeper。

## Springboot整合Eureka

## 单体应用

**1.在父项目中添加springCloud依赖**

<dependencyManagement>

<dependencies>

<!-- springCloud核心包 -->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>2021.0.4</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

**2.创建explore-eureka-server模块项目并添加对应依赖**

<!-- eureka server-->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>

**3.配置eureka注册中心**

server:

port: 9000 #服务注册中心端口号

spring:

application:

name: explore-eureka-server

eureka:

instance:

hostname: 127.0.0.1 #服务注册中心IP

client:

service-url:

defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka #注册地址

register-with-eureka: false #是否往服务注册中心注册自己，默认为true。单机版的可设置成false。集群搭建中，设置true。

fetch-registry: false #单机版的可设置成false，集群版的由于需要同步其他节点的服务注册数据，故设成true。

server:

enable-self-preservation: false #是否开启自我保护模式，默认值true

eviction-interval-timer-in-ms: 60000 #扫描失效服务的间隔时间，单位毫秒，默认值 60 \* 1000

**4.配置启动类**

@SpringBootApplication

@EnableEurekaServer

**public** **class** EurekaApp {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(EurekaApp.**class**, args);

}

}

**5.创建服务模块(explore-member-service)并父项目中添加依赖**

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

**6.配置eurek将服务注册到eureka注册中心**

server:

port: 9010

spring:

application:

name: explore-member-service

profiles:

active: dev

eureka:

client:

service-url:

defaultZone: http://@eureka.instance.hostname@:@server.eureka.port@/eureka

instance:

prefer-ip-address: true

**7.配置启动类**

@EnableEurekaClient

@SpringBootApplication

**public** **class** MemberApp {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(MemberApp.**class**, args);

}

}

## 集群

**1.在父项目中添加springCloud的依赖**

<dependencyManagement>

<dependencies>

<!-- springCloud核心包 -->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>2021.0.4</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

**2.搭建Eureka模块项目(explore-eureka-server)并在模块Eureka模块中添加依赖**

<!-- eureka server-->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>

**3.配置eureka注册中心**

server:

port: 9000 #服务注册中心端口号

spring:

application:

name: explore-eureka-server

eureka:

instance:

hostname: 127.0.0.1 #服务注册中心IP

client:

service-url:

defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka #注册地址

register-with-eureka: true #是否往服务注册中心注册自己，默认为true。单机版的可设置成false。集群搭建中，设置true。

fetch-registry: true #单机版的可设置成false，集群版的由于需要同步其他节点的服务注册数据，故设成true。

server:

enable-self-preservation: false #是否开启自我保护模式，默认值true

eviction-interval-timer-in-ms: 60000 #扫描失效服务的间隔时间，单位毫秒，默认值 60 \* 1000

**4.编写启动类**

@SpringBootApplication

@EnableEurekaServer

**public** **class** EurekaApp {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(EurekaApp.**class**, args);

}

}

**5.创建服务模块(explore-member-service)并在父项目中添加对应依赖**

<!-- eureka server-->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>

**6.将服务注册到Eureka**

server:

port: 9010

spring:

application:

name: explore-member-service

profiles:

active: dev

eureka:

client:

service-url:

defaultZone: http://127.0.0.1:9000/eureka,http://127.0.0.1:9001/eureka

instance:

prefer-ip-address: true