# 什么时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，- 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分布式开友五大常用组件**

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

分布式配置一Spring Cloud Contia

# 服务注册和发现

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

### 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