

# 1. 课程介绍

---

## 1.1 课程内容

1. 介绍微服务的由来，以及微服务和 Spring Cloud 之间的关系
2. 介绍 Spring Cloud 核心组件的使用，使小伙伴们通过核心组件可以快速搭建一个微服务架构
3. 介绍 Spring Cloud 中的辅助类组件，例如微服务监控、链路追踪等等
4. 介绍 Spring Cloud Alibaba，以及相关核心组件的具体用法

## 1.2 课程收获

1. 了解微服务的由来以及基本原理
2. 学会 Spring Cloud 中各个组件的使用
3. 了解 Spring Cloud 中核心组件的运行原理
4. 掌握通过 Spring Cloud 搭建微服务架构
5. 掌握辅助组件的用法

# 2. 微服务介绍

---

微服务架构越来越流行，这个没有异议。

2009 年，Netflix 重新定义了它的应用程序员的开发模型，这个算是微服务的首次探索。

20014 年，[《Microservices》](#)，这篇文章以一个更加通俗易懂的方式，为大家定义了微服务。

为什么要用微服务？

互联网应用产品的两大特点：

1. 需求变化快
2. 用户群体庞大

在这样的情况下，我们需要构建一个能够灵活扩展，同时能够快速应对外部环境变化的一个应用，使用传统的开发方式，显然无法满足需求。这个时候，微服务就登场了。

## 2.1 什么是微服务

简单来说，微服务就是一种将一个单一应用程序拆分为一组小型服务的方法，拆分完成后，每一个服务都运行在独立的进程中，服务于服务之间采用轻量级的通信机制来进行沟通（Spring Cloud 中采用基于 HTTP 的 RESTful API）。

每一个服务，都是围绕具体的业务进行构建，例如一个电商系统，订单服务、支付服务、物流服务、会员服务等等，这些拆分后的应用都是独立的应用，都可以独立的部署到生产环境中。就是在采用微服务之后，我们的项目不再拘泥于一种语言，可以 Java、Go、Python、PHP 等等，混合使用，这在传统的应用开发中，是无法想象的。而使用了微服务之后，我们可以根据业务上下文来选择合适的语言和构建工具进行构建。

微服务可以理解为是 SOA 的一个传承，一个本质的区别是微服务是一个真正分布式、去中心化的，微服务的拆分比 SOA 更加彻底。

## 2.2 微服务优势

1. 复杂度可控
2. 独立部署

3. 技术选型灵活
4. 较好的容错性
5. 较强的可扩展性

## 2.3 使用 Spring Cloud 的优势

Spring Cloud 可以理解为微服务这种思想在 Java 领域的一个具体落地。Spring Cloud 在发展之初，就借鉴了微服务的思想，同时结合 Spring Boot，Spring Cloud 提供了组件的一键式启动和部署的能力，极大的简化了微服务架构的落地。

Spring Cloud 这种框架，从设计之初，就充分考虑了分布式架构演化所需要的功能，例如服务注册、配置中心、消息总线以及负载均衡等。这些功能都是以可插拔的形式提供出来的，这样，在分布式系统不断演化的过程中，我们的 Spring Cloud 也可以非常方便的进化。

## 3. Spring Cloud 介绍

---

### 3.1 什么是 Spring Cloud

Spring Cloud 是一系列框架的集合，Spring Cloud 内部包含了许多框架，这些框架互相协作，共同来构建分布式系统。利用这些组件，可以非常方便的构建一个分布式系统。

### 3.2 核心特性

1. 服务注册与发现
2. 负载均衡
3. 服务之间调用
4. 容错、服务降级、断路器
5. 消息总线
6. 分布式配置中心
7. 链路器

### 3.3 版本名称

不同于其他的框架，Spring Cloud 版本名称是通过 A (Angel) 、 B (Brixton) 、 C (Camden) 、 D (Dalston) 、 E (Edgware) 、 F (Finchley) 。。这样来命名的，这些名字使用了伦敦地铁站的名字，目前最新版是 H (Hoxton) 版。

Spring Cloud 中，除了大的版本之外，还有一些小版本，小版本命名方式如下：

- M，M 版是 milestone 的缩写，所以我们会看到一些版本叫 M1、M2
- RC，RC 是 Release Candidate，表示该项目处于候选状态，这是正式发版之前的一个状态，所以我们会看到 RC1、RC2
- SR，SR 是 Service Release，表示项目正式发布的稳定版，其实相当于 GA (Generally Available) 版。所以，我们会看到 SR1、SR2
- SNAPSHOT，这个表示快照版

## 4. Spring Cloud 体系

---

### 4.1 Spring Cloud 包含的组件

- Spring Cloud Netflix，这个组件，在 Spring Cloud 成立之初，立下了汗马功劳。但是，2018 年的断更，也是 Netflix 掉链子了。
- Spring Cloud Config，分布式配置中心，利用 **Git**/Svn 来集中管理项目的配置文件
- Spring Cloud Bus，消息总线，可以构建消息驱动的微服务，也可以用来做一些状态管理等

- Spring Cloud Consul, 服务注册发现
- Spring Cloud Stream, 基于 Redis、RabbitMQ、Kafka 实现的消息微服务
- Spring Cloud OpenFeign, 提供 OpenFeign 集成到 Spring Boot 应用中的方式, 主要解决微服务之间的调用问题
- Spring Cloud Gateway, Spring Cloud 官方推出的网关服务
- Spring Cloud Cloudfoundry, 利用 Cloudfoundry 集成我们的应用程序
- Spring Cloud Security, 在 Zuul 代理中, 为 OAuth2 客户端认证提供支持
- Spring Cloud AWS, 快速集成亚马逊云服务
- Spring Cloud Contract, 一个消费者驱动的、面向 Java 的契约框架
- Spring Cloud Zookeeper, 基于 Apache Zookeeper 的服务注册和发现
- Spring Cloud Data Flow, 在一个结构化的平台上, 组成数据微服务
- Spring Cloud Kubernetes, Spring Cloud 提供的针对 Kubernetes 的支持
- Spring Cloud Function
- Spring Cloud Task, 短生命周期的微服务

## 4.2 Spring Cloud 和 Spring Boot 版本关系

Spring Cloud	Spring Boot
Hoxton	2.2.x
Greenwich	2.1.x
Finchley	2.0.x
Edgware	1.5.x
Dalston	1.5.x