# 微服务开发

## 微服务概念

### 概念

核心将传统一站式应用，根据业务拆成一个个服务，彻底的去偶合，每一个微服务提供单个业务功能的服务（一个服务只做一件事情，把事情交给专业的的机制去做）；

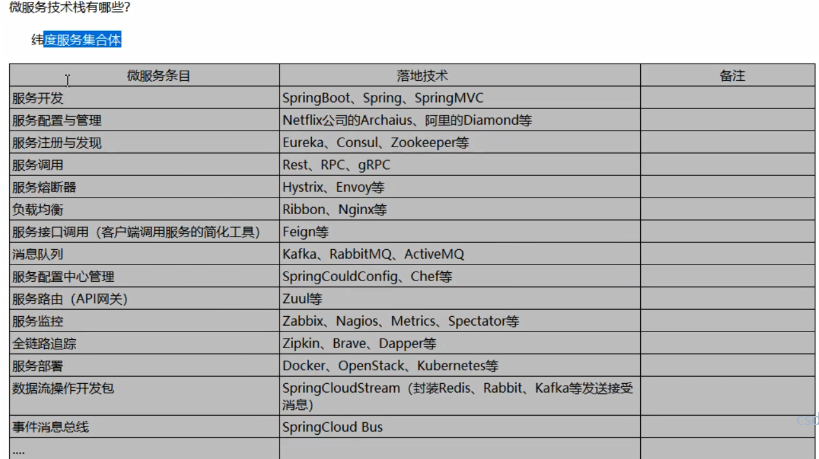
从技术角度看其实就是一种小而独立的处理过程，类似于进程的概念，每个服务运行在自己的进程中，能够自行单独启动或者销毁，拥有自己独立的数据库；

## 微服务框架

### 介绍

SpringBoot、SpringCloud、Dubbo、Springblade

### 技术栈



### 优缺点

优点：

1.每个服务足够内聚，足够小，代码容易理解，模块足够松耦合，部署阶段独立；

2.开发简单，开发效率高，一个服务专做一件事情。是通过轻量级机制HttpApi通信；

3.微服务能被小团队单独开发，（2~5个人可开发）

4.可以使用不同语言开发；

5.易于和第三方集成；

6.运行使用融合最新技术；

7.微服务只能是业务逻辑代码，不会和前端页面组件混合；

8.每个模块可以有单独的数据库（有独立的存储功能）或者同一个数据库；

缺点：

1.开发人员要处理分布式系统的的复杂性；

2.多服务运维难度，能随着服务的增多，运维压力增大；

3.服务之间的通信成本增加；

4.数据的一致性；

5.系统的继承测试；

6.性能监控；

## 三、微服务框架技术

### 1.SpringBoot

### 2.SpringCloud

#### 2.1介绍

SpringCloud（属于NetFlix）微服务开发涉及到的：

配置管理，服务治理，熔断机制智能路由，微代理，控制总线，一次性Token，全局一致性锁，Leader选举，分布式Session，

集群状态管理等操作提供了一种简单开发方式

SpringCloud对微服务基础框架NetFlix的多个开元组件进行了封装，同时有实现了和云端平台以及 Springboot框架的集成；通过HttpApi进行通信;

#### 2.2技术组件

### 3.Dubbo

### SpringBlade

#### 4.1介绍

SpringBlade是一个基于Spring+SpringMVC+Beetl+Beetlsql+Shiro的开发框架。具有权限管理，多角色，父子角色，权限代理的功能，权限控制精确到按钮

#### 4.2技术组件