目录

[**第一章** **微服务简介** 1](#_Toc42005276)

[**1.1** **单体架构** 1](#_Toc42005277)

[**1.1.1** **单体架构得优缺点** 1](#_Toc42005278)

[**1.2** **微服务是什么** 1](#_Toc42005279)

[**1.3** **微服务具备得特性** 1](#_Toc42005280)

[**1.4** **微服务优缺点** 1](#_Toc42005281)

[**1.5** **微服务设计原则** 1](#_Toc42005282)

[**第二章** **微服务技术知识** 2](#_Toc42005283)

[**2.1 Spring Cloud五大组件** 2](#_Toc42005284)

1. **微服务简介**
   1. **单体架构**
      1. **单体架构得优缺点**

优点：1.便于开发2.易于测试3.易于部署

缺点：1.复杂性高2.开发速度慢3.扩展性差4.代码部署周期长5.如果系统出现内部错误，有可能直接可能造成系统宕机。6.阻碍技术创新

* 1. **微服务是什么**

把一个庞大的系统拆分为若干个小模块，每个模块都可以独立应用服务，各个服务之间采用HTTP等轻量级的机制进行通讯。这些服务围绕业务功进行构建，通过全自动部署进行独立部署。各个服务可以采用不同语言进行编写，使用不同的数据存储技术。

* 1. **微服务具备得特性**

1. 每个服务可独立运行在自己的进程里；
2. 一系列独立运行的微服务共同构建起了整个系统；
3. 每个服务为独立的业务开发，一个微服务一般完成某个特定的功能，比如：订单管理、用户管理等；
4. 微服务之间通过一些轻量的通信机制进行通信，例如通过REST API或者RPC的方式进行调用。
   1. **微服务优缺点**

优点：

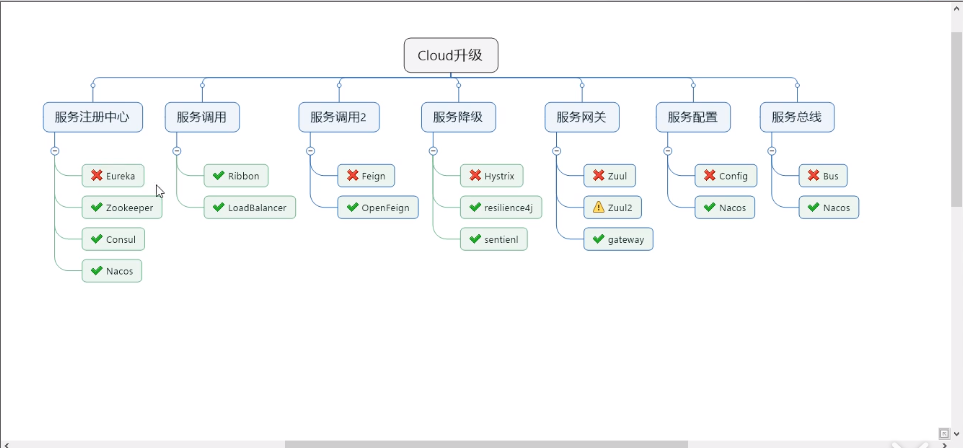
1. 易于开发和维护
2. 启动较快
3. 局部修改容易部署
4. 技术栈不受限
5. 按需伸缩

缺点：

1. 微服务过多，服务治理成本高，不利于系统维护
2. 分布式系统开发的技术成本高（容错、分布式事务等），对团队挑战大。
   1. **微服务设计原则**
3. 单一职责原则
4. 服务自治原则
5. 轻量级通信原则
6. 接口明确原则
7. **微服务技术知识**

**2.1 Spring Cloud五大组件**

1. 服务注册中心——Netflix Eureka——Zookeeper——Consul——AlibabaNacos
2. 客户端负载均衡——Netflix Ribbon——LoadBalance
3. 断路器——Netflix Hystrix——resilience4j——Sentinel
4. 服务网关——Netflix Zuul——Gateway
5. 分布式配置——Spring Cloud Config——AlibabaNacos



* 1. **Netflix Eureka**
     1. **Eureka基本知识**

什么是服务治理？

Spring Cloud封装了Netfix公司开发的Eureka模块来实现服务治理

在传统得rpc远程调用框架中，管理每个服务之间依赖关系比较复杂，管理比较复杂，所以需要使用服务治理，服务之间以来关系，可以实现服务调用、负载均衡、容错等，实现服务发现与注册。

