# Spring Cloud答案

# 简述题：简述互联网应用架构演进

互联网应用架构经历了从单体应用架构、垂直应用架构、SOA应用架构到微服务应用架构的演进。

（1）单体应用架构：

在诞⽣之初，系统的⽤户量、数据量规模都⽐较⼩，项目所有的功能模块都放在一个工程中编码、编译、打包并且部署在一个Tomcat容器中的架构模式就是单体应用架构，这样的架构既简单实 ⽤、便于维护，成本⼜低，成为了那个时代的主流架构⽅式。

（2）垂直应用架构：

为了单体应用的那些问题，开始做模块的垂直划分，做垂直划分的原则是基于系统现有的业务特性来做，核心目标标第⼀个是为了业务之间互不影响，第⼆个是在研发团队的壮⼤后为了提⾼效率，减少组件之间的依赖。

（3）SOA架构

​ 在做了垂直划分以后，模块随之增多，维护的成本在也变⾼，⼀些通⽤的业务和模块重复的越来越多，为了解决上⾯提到的接⼝协议不统⼀、服务⽆法监控、服务的负载均衡，引⼊了阿⾥巴巴开源的 Dubbo ，⼀款⾼性能、轻量级的开源Java RPC框架，可以和Spring框架无缝集成。它提供了三⼤核⼼能⼒：⾯向接⼝的远程⽅法调⽤，智能容错和负载均衡，以及服务⾃动注册和发现。

SOA (Service-Oriented Architecture)，即面向服务的架构。根据实际业务，把系统拆分成合适的、独立部署的模块，模块之间相互独立（通过Webservice/Dubbo等技术进行通信）。

（4）微服务架构

微服务架构可以说是SOA架构的一种拓展，这种架构模式下它\*\*拆分粒度更小\*\*、服务更独立。把应用拆分成为一个个微小的服务，不同的服务可以使用不同的开发语言和存储，服务之间往往通过Restful等轻量级通信。微服务架构关键在于微小、独立、轻量级通信。

微服务是在 SOA 上做的升华粒度更加细致，微服务架构强调的⼀个重点是业务需要彻底的组件化和服务化

# 面试题：Spring Cloud对比Dubbo

Dubbo是阿里巴巴公司开源的一个高性能优秀的服务框架，基于RPC调用，对于目前使用率较高的Spring Cloud Netflix来说，它是基于HTTP的，所以效率上没有Dubbo高，但问题在于Dubbo体系的组件不全，不能够提供一站式解决方案，比如服务注册与发现需要借助于Zookeeper等实现，而Spring Cloud Netflix则是真正的提供了一站式服务化解决方案，且有Spring大家族背景。

# 面试题：Spring Cloud对比Spring Boot

Spring Cloud 只是利用了Spring Boot 的特点，让我们能够快速的实现微服务组件开发，否则不使用Spring Boot的话，我们在使用Spring Cloud时，每一个组件的相关Jar包都需要我们自己导入配置以及需要开发人员考虑兼容性等各种情况。所以Spring Boot是我们快速把Spring Cloud微服务技术应用起来的一种方式。

# 简述题：简述常用的服务注册中心有哪些

常用的服务注册中心：Eureka、Nacos、Zookeeper、Consul

# 面试题：Eureka工作原理

Eureka : 就是服务注册中心(可以是一个集群),对外暴露自己地址;

提供者 : 启动后向Eureka注册自己信息(地址,提供什么服务)

消费者 : 向Eureka 订阅服务,Eureka会将对应服务的服务列表发送给消费者,并且定期更新

心跳(续约): 提供者定期通过http方式向Eureka刷新自己的状态

# 面试题：什么是服务注册

服务提供者在启动时,会向Eureka Server发起一次请求,将自己注册到Eureka注册中心中去

# 面试题：什么是服务续约

在注册服务完成以后,服务提供者会维持一个心跳(每30s定时向Eureka Server 分发起请求)告诉Eureka Server "我还活着"

# 面试题：什么是失效剔除

有时候,我们的服务提供方并不一定是正常下线,可能是内存溢出,网络故障等原因导致服务无法正常工作.Eureka Server会将这些失效的服务剔除服务列表.因此它会开启一个定时任务.每隔60秒会对失效的服务进行一次剔除

# 面试题：什么是自我保护

当服务未按时进行心跳续约时,在生产环境下,因为网络原因,此时就把服务从服务列表中剔除并不妥当发,因为服务也有可能未宕机.Eureka就会把当前实例的注册信息保护起来,不允剔除.这种方式在生产环境下很有效,保证了大多数服务依然可用