目录

产品常见问题

微服务和普通应用有什么不同？

微服务:是一种架构模式，其核心是将一个单体应用分成多个服务进行开发，其本质上是一个分布式应用。基于微服务架构，构建的应用程序，可以让业务更加灵活，整体系统可靠性更高。
普通应用:逻辑复杂、模块耦合、代码臃肿、修改难度大、版本迭代效率低下。线上问题修复周期长，一个线上问题修复都可能需要对整个应用系统进行全面升级。

微服务有什么好处？

第一，从领域模型纬度上，把原来很复杂的庞大系统拆分成了各个系统，各个系统之间通过接口来进行交互，这样各个业务系统变得很清晰。
第二，从软件开发角度，通过微服务的拆分，形成了统一的开发标准，开发人员只需关注自己模块实现方式，有利于项目快速上线，提升跨部门协作，提高了研发效率。

开发常见问题

SpringCloud和Spring boot区别？

Spring Cloud：微服务工具包，为开发者提供了在分布式系统的配置管理、服务发现、断路器、智能路由、微代理、控制总线等开发工具包。
Spring Boot:旨在简化创建产品级的Spring应用和服务，简化了配置文件，使用嵌入式web服务器，含有诸多开箱即用微服务功能，可以和spring cloud联合部署。

什么是健康检查？

每个微服务必须为它自身的状态负责，所以每个微服务都应提供一个健康检查的接口。通过调用这个健康检查接口，外界可以判断这个服务当前的状态。
一般情况下，并不是容器启动后容器中的应用就马上就绪了，应用一般还有一个启动或初始化的过程。因此，必须有一种手段让平台检查微服务应用的就绪状态。
平台通过检查Readiness Probe接口，只有在确认服务就绪后，才会将外界的流量转发至服务。如果一个服务的Liveness Probe探测结果返回失败，平台就会判定这个容器实例出现了问题，相应的容器会被停止。

服务要怎么拆分？

垂直拆分：按照业务功能去拆分。如将会员系统和交易系统拆分为两个不同的服务，因为它们在业务上有很强的隔离性。会员系统，主要实现会员注册、登陆、积分、个人信息等功能；交易系统，主要实现下单、付费等要功能。
水平拆分：按照代码层面去拆分。按照开发人员纬度，如退款有专门的团队来管理，这时候就需要拆分；如果交易和退款是同一个团队开发的，拆不拆无所谓。服务拆分的核心原则是高内聚低耦合。