

一、微服务介绍

更多学习资料，一点课堂：www.yidiankt.com

一点课堂QQ群：984370849，QQ：2868289889

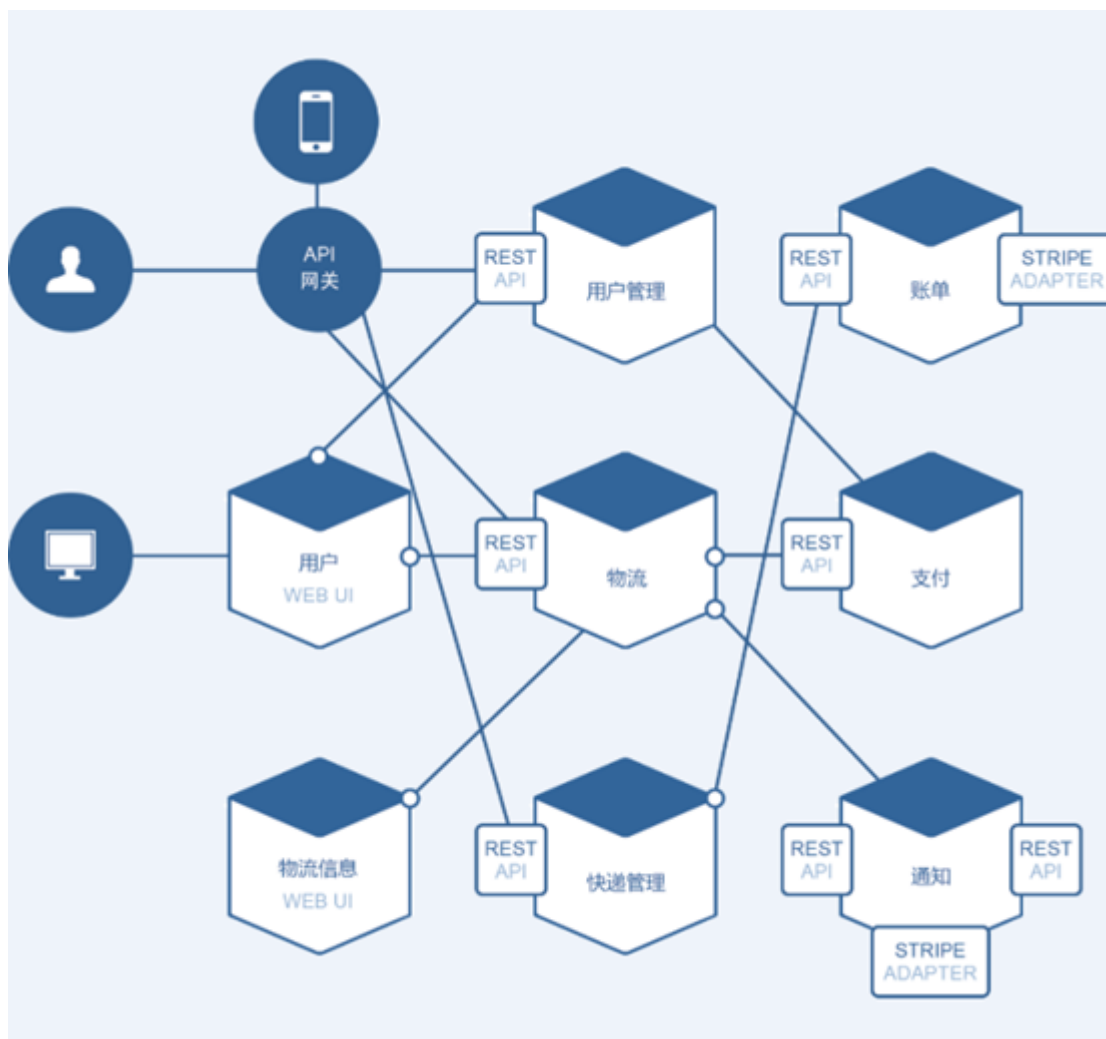
微信号：chengweixin9

免费公开课：<https://ke.qq.com/course/394307>

1. 什么是微服务？

- 微服务是一种架构风格，也是一种服务；
- 微服务的颗粒比较小，一个大型复杂软件应用由多个微服务组成，比如Netflix目前由500多个的微服务组成；
- 它采用UNIX设计的哲学，每种服务只做一件事，是一种松耦合的能够被独立开发和部署的无状态化服务（独立扩展、升级和可替换）。

2. 微服务架构图



3. 微服务的好处

- 技术异构性：在一个由多个服务相互协作的系统中，可以在不同的服务中使用最适合该服务的技术。
- 弹性：如果系统中的一个组件不可用了，但并没有导致级联故障，那么系统的其他部分还可以正常运行。

- 扩展：可以只对那些需要扩展的服务进行扩展。
- 简化部署：各个服务的部署是独立的，这样就可以更快地对特定部分的代码进行部署。
- 与组织结构相匹配：可以很好地将架构与组织结构相匹配，避免出现过大的代码库，从而获得理想团队大小及生产力。
- 可组合性：不同服务模块的接口可以再进行重用，成为其他产品中的一个组件；
- 对可替代性的优化：可以在需要时轻易地重写服务，或者删除不再使用的服务

4. 微服务的缺点

1. 运维开销 更多的服务也就意味着更多的运维，产品团队需要保证所有的相关服务都有完善的监控等基础设施，传统的架构开发者只需要保证一个应用正常运行，而现在却需要保证几十甚至上百道工序高效运转，这是一个艰巨的任务。
2. DevOps要求 使用微服务架构后，开发团队需要保证一个Tomcat集群可用，保证一个数据库可用，这就意味着团队需要高品质的DevOps和自动化技术。而现在，这样的全栈式人才很少。
3. 隐式接口 服务和接口之间通过接口来“联系”，当某一个服务更改接口格式时，可能涉及到此接口的所有服务都需要做调整。
4. 重复劳动 在很多服务中可能都会使用到同一个功能，而这一功能点没有足够大到提供一个服务的程度，这个时候可能不同的服务团队都会单独开发这一功能，重复的业务逻辑，这违背了良好的软件工程中的很多原则。
5. 分布式系统的复杂性 微服务通过REST API或消息来将不同的服务联系起来，这在之前可能只是一个简单的远程过程调用。分布式系统也就意味着开发者需要考虑网络延迟、容错、消息序列化、不可靠的网络、异步、版本控制、负载等，而面对如此多的微服务都需要分布式时，整个产品需要有一整套完整的机制来保证各个服务可以正常运转。
6. 事务、异步、测试面临挑战 跨进程之间的事务、大量的异步处理、多个微服务之间的整体测试都需要有一整套的解决方案，而现在看起来，这些技术并没有特别成熟。

注意：

需要根据业务的不同的阶段进行合理的引入，不能完全为了微服务而“微服务”。