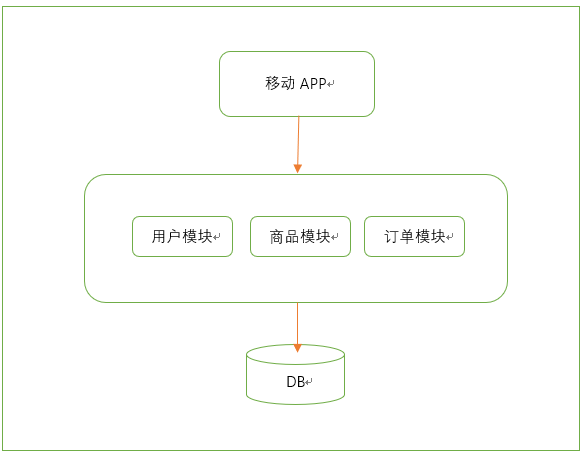
1、微服务架构简介

随着互联网的快速发展，业务场景越来越复杂。传统的单体应用架构已经很难满足业务的发展要求。所以微服务架构闯入了各行各业的开发者视野 成为突破现有业务发展瓶颈的热门技术。最近两年微服务在各大技术会议、文章、书刊、博客等出现的频率已是非常高了，同时它也给软件开发领域带来了巨大的影响。关于更多的微服务理论知识，本书就不再过多提及了，网上已有很多了。

1.1.1、单体应用架构

传统的单体应用架构将所有的业务需求按模块的形式进行划分，例如购物类APP应用，所有的功能模块都在一个war或jar中部署，某一个模块的需求发生变化时，需要需要将所有的模块编译、打包、上线，各模块间相互耦合相互依赖。久而久之应用变得越来越臃肿、灵活性下降。



1.1.2、单体应用的不足

1. 代码复杂性增加了，随着需求的快速迭代，代码的体量也在快速增加，这样造成了在原有的模块上的代码越来越臃肿
2. 灵活性降低了，当需求发生变更是，需要自上而下的去修改，测试时必须等到整个程序部署完后才能看出效果。在开发过程可能需要等待其他的开发人员开发完成才能完成部署，降低团队的灵活。
3. 系统的扩展性差，单体应用只能对整体进行扩展，无法根据业务和用户使用情况进行对具体的业务模块进行不同程度的伸缩扩展。
4. 系统启动慢，一个进程包含了所有的业务逻辑，涉及到的启动模块过多，导致系统的启动、重启时间周期过长。

1.2 微服务

1.2.1 、微服务简介



Martin Fowler 提出 “微服务架构”，它用于描述一种独立部署的软件应用设计方式。这种架构方式并没有非常明确的定义，但有一些共同的特点就是围绕在业务能力、自动化布署、端到端的整合以及语言和数据的分散控制上面。总之，微服务这种架构风格就是把一组小服务演化成为一个单一的应用的一种方法。每个应用都运行在自己的进程中，并通过轻量级的机制保持通信，就像HTTP这样的API。这些服务要基于业务场景，并使用自动化布署工具进行独立的发布。可以有一个非常轻量级的集中式管理来协调这些服务，可以使用不同的语言来编写服务，也可以使用不同的数据存储。

1.2.2、微服务的优势

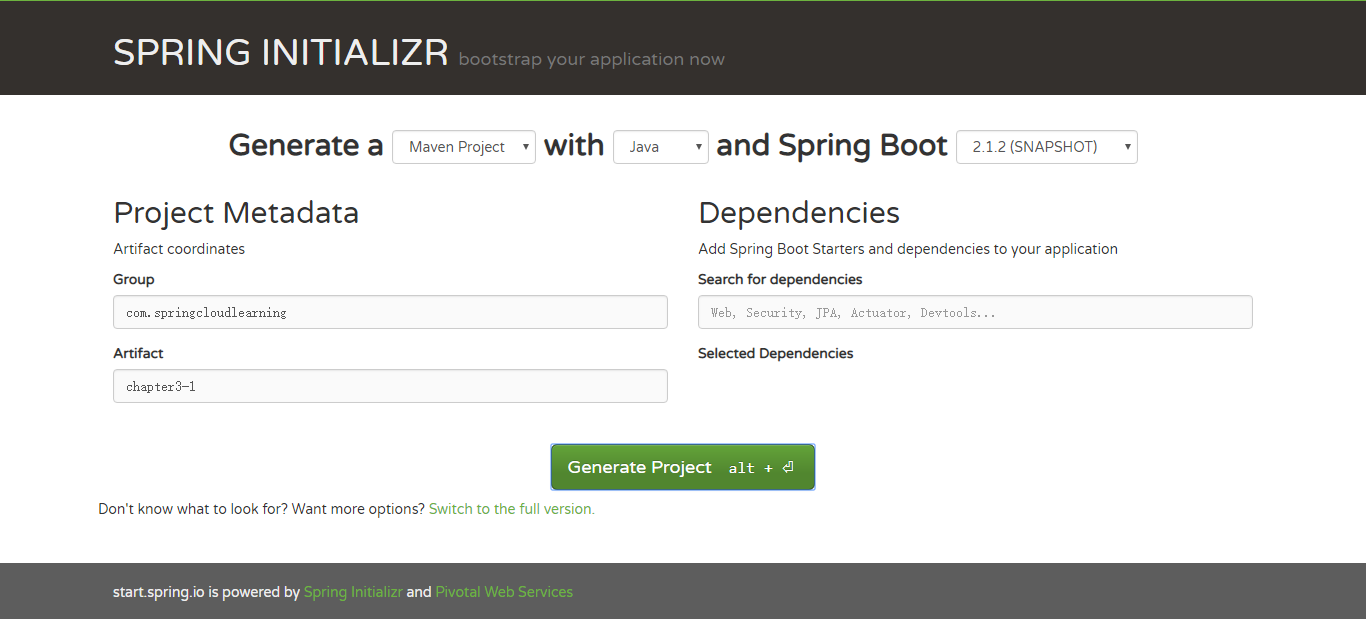
1. 它解决了复杂问题。它把可能会变得庞大的单体应用程序分解成一套服务。虽然功能数量不变，但是应用程序已经被分解成可管理的块或者服务。每个服务都有一个以远程过程调用（RPC）驱动或者消息驱动的 API 明确定义的边界。微服务架构模式强制一定程度的模块化，实际上，使用单体基础代码来实现是极其困难的。因此，个体服务能被更快地开发，并更容易理解与维护。
2. 这种架构使得每个服务都可以由一个团队独立专注开发。开发者可以自由选择任何符合服务 API 契约的技术。当然，更多的组织是希望通过技术选型限制来避免完全混乱的状态。然而，这种自由意味着开发人员不再有可能在这种自由的新项目开始时使用过时的技术。当编写一个新服务时，他们可以选择当前的技术。此外，由于服务比较小，使用当前技术重写旧服务将变得更加可行。
3. 微服务架构模式可以实现每一个微服务独立部署。开发人员根本不需要去协调部署本地变更到它们的服务。这些变更一经测试即可立即部署。比如，UI 团队可以执行 A|B 测试，并快速迭代 UI 变更。微服务架构模式使得持续部署成为可能。
4. 微服务架构模式使得每个服务能够独立扩展。您可以仅部署满足每个服务的容量和可用性约束的实例数目。此外，您可以使用与服务资源要求最匹配的硬件。例如，您可以在 EC2 Compute Optimized 实例上部署一个 CPU 密集型图像处理服务，并且在 EC2 Memory-optimized 实例上部署一个内存数据库服务。

1.3、Spring Boot 简介

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建及开发的过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Boot致力于在蓬勃发展的快速应用开发领域（rapid application development）成为领导者。

3.1 Spring Boot 之 HelloWorld实战

快速搭建Spring Boot项目工程可以在<https://start.spring.io/>网址选择Spring Boot版本下载，再导入工程



导入成功后添加一个Hello World的HelloWorldControlle类

@RestController

public class HelloWorldController {

@RequestMapping("/hello")

public String index() {

return "Hello World";

}

}

运行成功结果如下：

