

Spring Cloud在国内中小型公司能用起来吗?

2017-09-13 张强 Docker



今天吃完饭休息的时候瞎逛知乎，突然看到这个一个问题『Spring Cloud在国内中小型公司能用起来吗?』，吸引了我的注意。仔细的看了题主的问题，发现这是一个好问题，题主经过了一番思考，并且用图形全面的将自己的疑问表达了出来，作为一个研究并使用Spring Boot和Spring Cloud近两年的程序员，看的我手痒痒不答不快呀。

好问题

好问题必须配认真的回答，仔细的看了题主的问题，发现这个问题非常具有代表性，可能是广大网友想使用Spring Cloud却又对Spring Cloud不太了解的想法，题主对Spring Cloud使用的方方面面都进行过了思考，包括市场，学习、前后端、测试、配置、部署、开发以及运维，下面就是题主原本的问题：

想在公司推广Spring Cloud，但我对这项技术还缺乏了解,画了一张脑图，总结了种种问题。



微服务是这样一个结构吗：

前端或二方 -> ng集群 -> zuul集群 -> eureka-server集群 -> service provider集群（二方指其他业务部门）

想要明白这个问题，首先需要知道什么是Spring Boot，什么是Spring Cloud，以及两者之间有什么关系？

什么是Spring Boot？

Spring Boot简化了基于Spring的应用开发，通过少量的代码就能创建一个独立的、产品级别的Spring应用。Spring Boot为Spring平台及第三方库提供开箱即用的设置，这样你就可以有条不紊地开始。多数Spring Boot应用只需要很少的Spring配置。

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应用的初始搭建以及开发过程。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。用我的话来理解，就是Spring Boot其实不是什么新的框架，它默认配置了很多框架的使用方式，就像maven整合了所有的jar包，Spring Boot整合了所有的框架（不知道这样比喻是否合适）。

Spring Boot的核心思想就是约定大于配置，一切自动完成。采用Spring Boot可以大大的简化你的开发模式，所有你想集成的常用框架，它都有对应的组件支持。如果你对Spring Boot完全不了解，可以参考我的这篇文章：《SpringBoot（一）：入门篇》（<http://t.cn/R9NHvPJ>）。

什么是Spring Cloud？

Spring Cloud是一系列框架的有序集合。它利用Spring Boot的开发便利性巧妙地简化了分布式系统基础设施的开发，如服务发现注册、配置中心、消息总线、负载均衡、断路器、数据监控等，都可以用Spring Boot的开发风格做到一键启动和部署。Spring并没有重复制造轮子，它只是将目前各家公司开发的比较成熟、经得起实际考验的服务框架组合起来，通过Spring Boot风格进行再封装屏蔽掉了复杂的配置和实现原理，最终给开发者留出了一套简单易懂、易部署和易维护的分布式系统开发工具包。

微服务是可以独立部署、水平扩展、独立访问（或者有独立的数据库）的服务单元，Spring Cloud就是这些微服务的大管家，采用了微服务这种架构之后，项目的数量会非常多，Spring Cloud作为大管家就需要提供各种方案来维护整个生态。

Spring Cloud就是一套分布式服务治理的框架，既然它是一套服务治理的框架，那么它本身不会提供具体功能性的操作，更专注于服务之间的通讯、熔断、监控等。因此就需要很多的组件来支持一套功能，如果你对Spring Cloud组件不是特别了解的话，可以参考我的这篇文章：《Spring Cloud（一）：大话Spring Cloud》（<http://t.cn/RX3bXL1>）。

Spring Boot和Spring Cloud的关系

Spring Boot是Spring的一套快速配置脚手架，可以基于Spring Boot快速开发单个微服务，Spring Cloud是一个基于Spring Boot实现的云应用开发工具；Spring Boot专注于快速、方便集成的单个微服务个体，Spring Cloud关注全局的服务治理框架；Spring Boot使用了默认大于配置的理念，很多集成方案已经帮你选择好了，能不配置就不配置，Spring Cloud很大一部分是基于Spring Boot来实现，可以不基于Spring Boot吗？不可以。

Spring Boot可以离开Spring Cloud独立使用开发项目，但是Spring Cloud离不开Spring Boot，属于依赖的关系。

Spring -> Spring Boot > Spring Cloud 这样的关系。

回答

以下为我在知乎的回答。

首先楼主问的这些问题都挺好的，算是经过了自己的一番思考，我恰好经历了你所说的中小公司，且都使用Spring Cloud并且已经投产上线。第一家公司技术开发人员15人左右，项目实例

30多，第二家公司开发人员100人左右，项目实例达160多。



实话说Spring Boot、Spring Cloud仍在高速发展，技术生态不断的完善和扩张，不免也会有一些小的bug，但对于中小公司的使用来将，完全可以忽略，基本都可以找到解决方案，接下来回到你的问题。

1、市场

据我所知有很多知名互联网公司都已经使用了Spring Cloud，比如阿里、美团但都是小规模，没有像我经历的这两家公司，业务线全部拥抱Spring Cloud；另外Spring Cloud并不是一套高深的技术，普通的Java程序员经过一到俩个月完全就可以上手，但前期需要一个比较精通人的来带队。

2、学习

有很多方式，现在Spring Cloud越来越火的情况下，各种资源也越来越丰富，查看官方文档和示例，现在很多优秀的博客在写spring cloud的相关教程，我这里收集了一些Spring Boot和Spring Cloud的相关资源可以参考，找到博客也就找到人和组织了。

3、前后职责划分

其实这个问题是每个系统架构都应该考虑的问题，Spring Cloud只是后端服务治理的一套框架，唯一和前端有关系的是thymeleaf，Spring推荐使用它做模板引擎。一般情况下，前端App或者网页通过Zuul来调用后端的服务，如果包含静态资源也可以使用Nginx做一下代理转发。

4、测试

Spring-boot-starter-test支持项目中各层方法的测试，也支持controller层的各种属性。所以一般测试的步奏是这样，首先开发人员覆盖自己的所有方法，然后测试微服务内所有对外接口保证微服务内的正确性，再进行微服务之间集成测试，最后交付测试。

5、配置

session共享有很多方式，比如使用tomcat session共享机制，但我比较推荐使用Redis缓存来做session共享。完全可以分批引入，我在上一家公司就是分批过渡上线，新旧项目通过Zuul进行交互，分批引入的时候，最好是新业务线先使用Spring Cloud，老业务做过渡，当完全掌握之后在全部替换。如果只是请求转发，Zuul的性能不一定比Nginx低，但是如果涉及到静态资

源，还是建议在前端使用Nginx做一下代理。另外Spring Cloud有配置中心，可以非常灵活的做所有配置的事情。



6、部署

多环境不同配置，Spring Boot最擅长做这个事情了，使用不同的配置文件来配置不同环境的参数，在服务启动的时候指明某个配置文件即可，例如：`java -jar app.jar --spring.profiles.active=dev`就是启动测试环境的配置文件；Spring Cloud 没有提供发布平台，因为Jenkins已经足够完善了，推荐使用Jenkins来部署Spring Boot项目，会省非常多的事情；灰度暂时不支持，可能需要自己来做，如果有多个实例，可以一个一个来更新；支持混合部署，一台机器部署多个是常见的事情。

7、开发

你说的包含html接口就是前端页面吧，Spring Boot可以支持，但其实也是Spring Mvc在做这个事情，Spring Cloud只做服务治理，其它具体的功能都是集成了各种框架来解决而已；excel报表可以，其实除过swing项目外，其它Java项目都可以想象；Spring Cloud和老项目可以混合使用，通过Zuul来支持。是否支持callback，可以通过MQ来实现，还是强调Spring Cloud只是服务治理。

8、运维

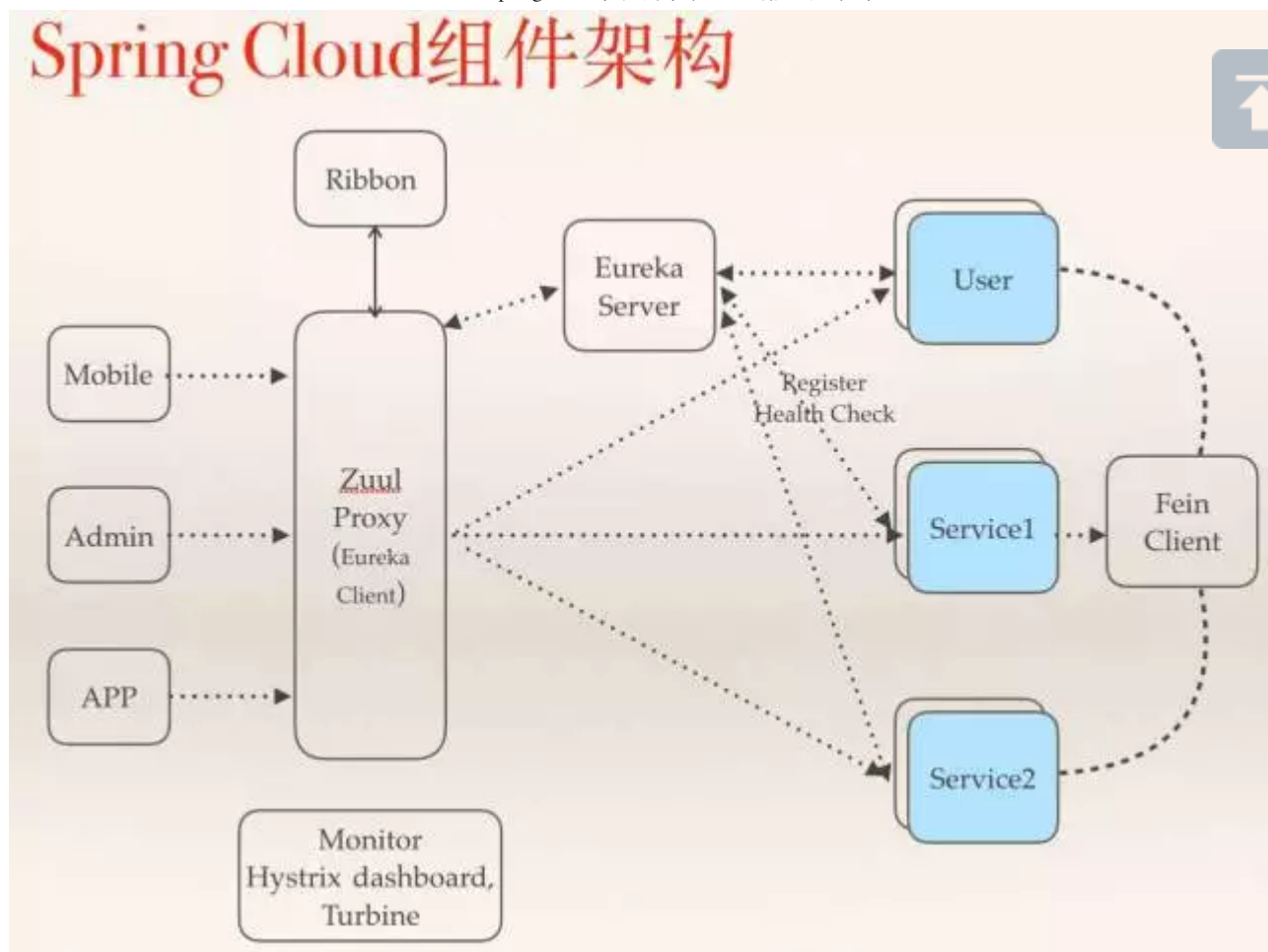
Turbine、Zipkin可以用来做熔断和性能监控；动态上下线某个节点可以通过Jenkins来实现；Provider下线后，会有其它相同的实例来提供服务，Eureka会间隔一段时间来检测服务的可用性；不同节点配置不同的流量权重目前还不支持。注册中心必须做高可用集群，注册中心挂掉之后，服务实例会全部停止。

总结，中小企业是否能用的起来Spring Cloud，完全取决于自己公司的环境，如果是一个技术活跃型的团队就大胆的去尝试吧，目前Spring Cloud是所有微服务治理中最优秀的方案，也是一个趋势，未来一两年可能就会像Spring一样流行，早接触早学习岂不更好。

希望能解答了你的疑问。

Spring Cloud 架构

我们从整体来看一下Spring Cloud主要的组件，以及它的访问流程：



- 外部或者内部的非Spring Cloud项目都统一通过API网关（Zuul）来访问内部服务
- 网关接收到请求后，从注册中心（Eureka）获取可用服务
- 由Ribbon进行均衡负载后，分发到后端的具体实例
- 微服务之间通过Feign进行通信处理业务
- Hystrix负责处理服务超时熔断
- Turbine监控服务间的调用和熔断相关指标

图中没有画出配置中心，配置中心管理各微服务不同环境下的配置文件。

以上就是一个完整的Spring Cloud生态图。

最后送一个完整示例的Spring Cloud开源项目等你去拿：

<https://github.com/ityouknow/spring-cloud-examples>

本文转载自公众号：纯洁的微笑，ID：keepuresmile。

烧脑式Kubernetes实战训练营

本次培训理论结合实践，主要包括：Kubernetes架构和资源调度原理、Kubernetes DNS与服务发现、基于Kubernetes和Jenkins的持续部署方案、Kubernetes网络部署实践、监控、日志、Kubernetes与云原生应用、在CentOS中部署Kubernetes集群、Kubernetes中的容器设计模式、开发Kubernetes原生应用步骤介绍等，点击识别下方二维码加微信好友了解**具体培训内容**。

9月21-23日

Kubernetes 实战训练营

识别
二
维
码
咨
询



- 每晚现场答疑 • 一线资深讲师 • 国家认证证书 •

点击[阅读原文](#)链接即可报名。

[阅读原文](#)