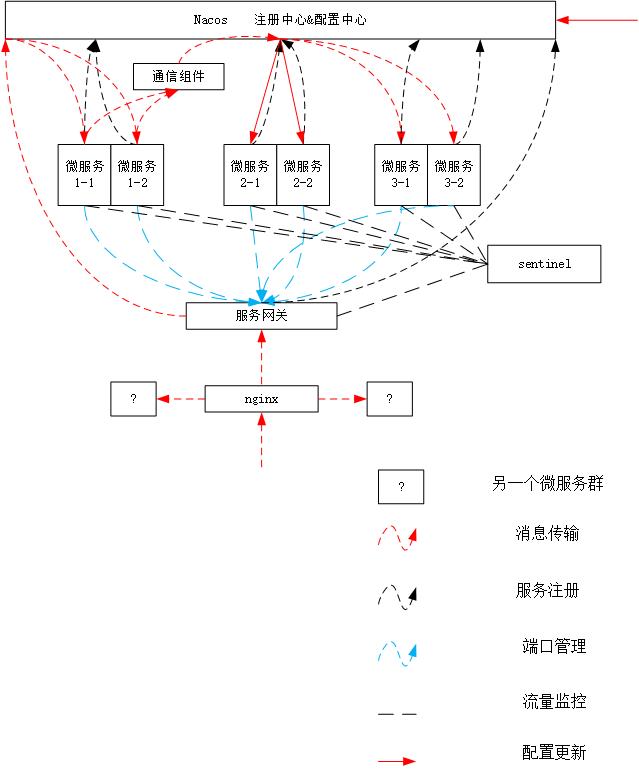
springCloud-Alibaba组件

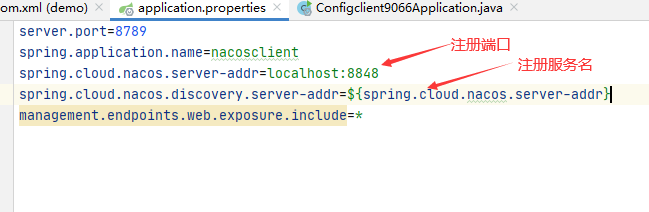
## SpringCloud-Alibaba概述

1. Springcloud-Alibaba在传统的SpringCloud的基础上，推出了更加便捷高效的组件来替换原先的组件。
2. 主要推出两个组件：用Nacos来代替原先的注册中心并整合bus和配置中心，用sentinel来取代原先的熔断器



## 组件一，Nacos

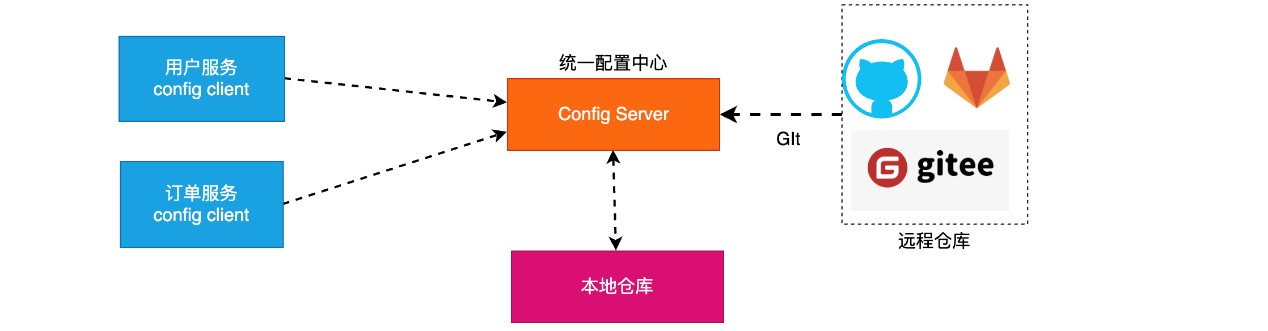
1. Nacos兼具着配置中心和注册中心功能，使用方法也和consul一样，先从网上下载一个可启动的脚本，在微服务配置文件中加相应的配置，即可完成注册



1. 也可以通过<http://localhost:8848/nacos/> 来访问nacos，得到注册中心和配置中心的相关信息



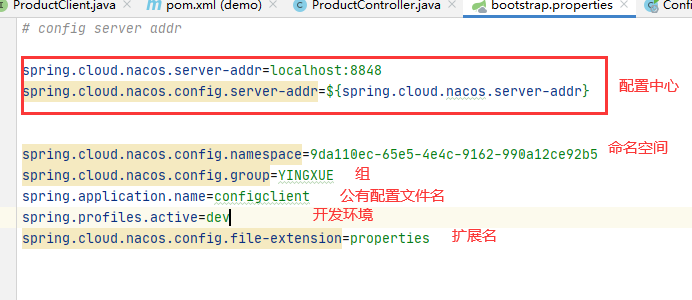
1. Nacos替代配置中心：
2. 之前我们在完成配置中心时，会构建一个配置中心微服务，然后连接git仓库，将配置文件交给git管理，git上的配置文件改变时，会通过webhook或主动的向配置中心发送post请求，然后配置中心依靠MQ消息组件向各个微服务发送消息来更新配置，这个通过很多功能单一的组件合作完成了一个消息配置中心



1. 由于配置中心构建是单一组件的拼装，很不便捷，nacos在替换注册中心之外，也替换了配置中心，nacos和git一样，可以对配置文件进行版本管理，通过感知配置文件的MD5码，来通知对应的服务进行配置文件的更新，且不需要构建配置中心微服务，只需要在配置文件中加入相应配置即可完成配置的管理



1. 配置文件的调用一般需要指定命名空间、组、配置文件（服务名-开发环境.扩展名），不可以对公有配制和开发、测试、生产等环境私有配置进行堆叠，他们是独立的

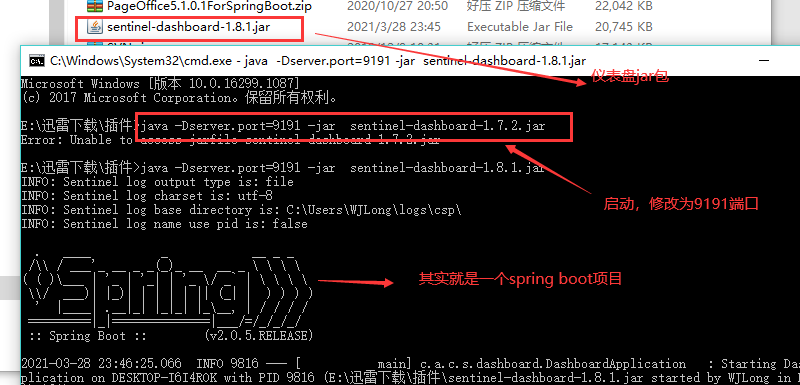


1. Nacos配置中心的配置文件在更新发布以后，会自动的进行MD5码的比对，并自动对微服务进行配置文件的更新，但是如果代码中用了配置文件中的变量，必须用@RefreshScope，才能及时的加载最新的变量



## 组件二sentinel

1. Hystrix组件在做熔断时，需要给每个类甚至每个方法写熔断后返回值fallback的方法，当业务量特别大的时候，这样做是特别麻烦甚至是冗余的，

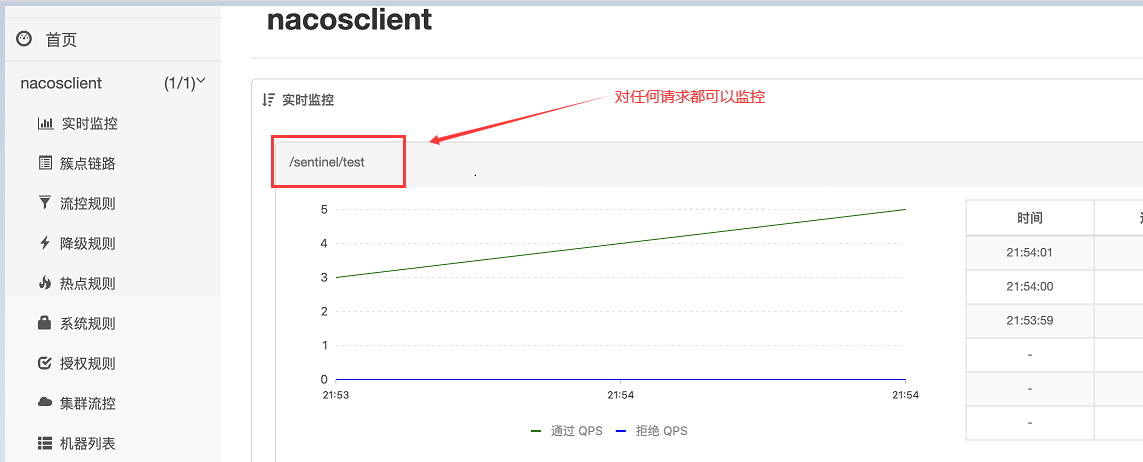
sentinel就很好的解决了这个问题，sentinel和Hystrix一样，都提供一个端口来做web仪表盘，但是sentinel仪表盘除了监控展示外，还会提供另一个端口对请求进行监控手动对服务进行熔断、降级等操作，sentinel服务是一个打成jar包的springboot项目，我们需要从github上下载下来，java -jar启动，可以通过端口进行仪表盘的访问等操作。

然后用端口进行访问：<http://localhost:9191/>



1. 在配置文件中加入仪表盘的地址，以及监控的端口，不需要写任何相关的代码，当我们对微服务进行某个请求时，该请求的相关数据就会在仪表盘中进行展示，我们可以根据每秒的访问数（QPS）或者并发量来对服务进行流量控制，根据响应时间、异常率、异常数量来熔断降级等操作，sentinel会提供默认的fallback信息、冷启动或者等待排队。







1. Sentinel提供了在仪表盘手动进行针对性的流量控制、熔断降级的功能，还提供了注解@SentinelResource 来自动定义出现流量控制、熔断降级的页面以及出现其他异常的返回页面

