Table of Contents

写作背景

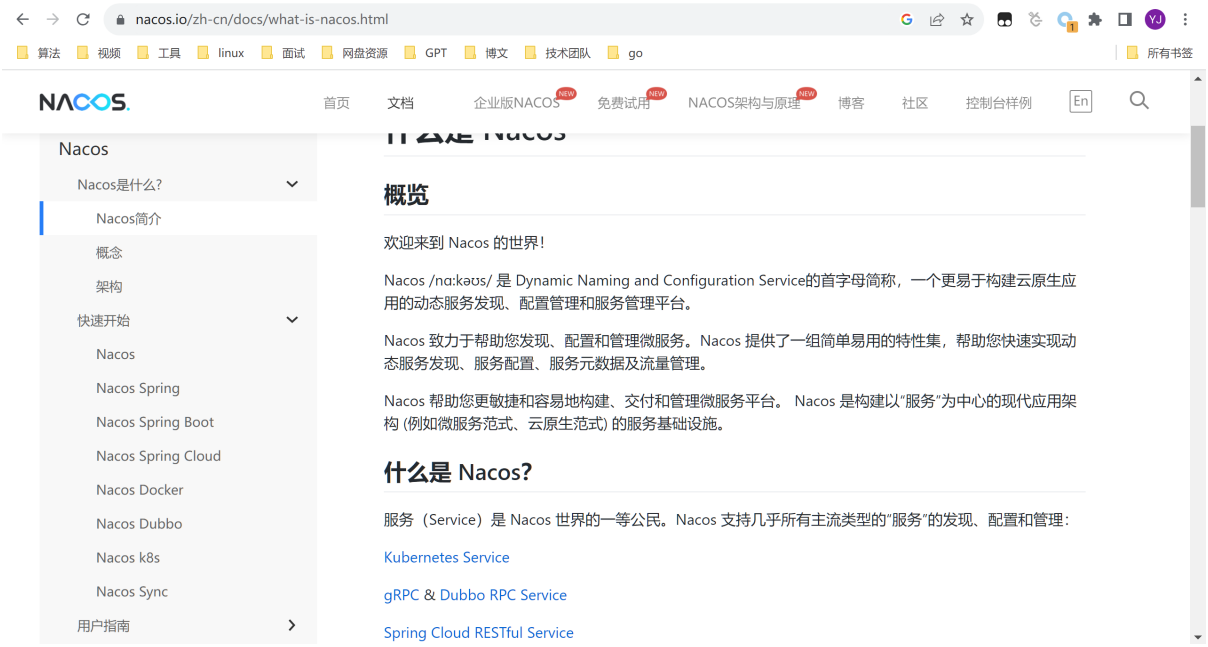
这本书是本人写的第一本gitbook书，大概耗费一个下午 将gitbook给安装。因为最近在研究nacos这个配置分发和注册中心，所以就想以此为题写一本书。希望之后的自己看到后，能够专心的把这本书给写出。

希望本书的写作内容，能给后来者一些启发，本书不仅仅是包含内容干活，更是面向个人的学习习惯触发。因为观念不同，可能我们看框架的视角不同，可做参考，但不可作为依赖。

本书基于Nacos2.2.3进行编写，如有错误的地方欢迎指正。

本书简介

首先，我们学习一门技术最好的方案就是从官方网站开始，下面是nacos的官方网站。



其中包含了其基本的介绍和与其它项目的集成,接下来我们将会从基本的概念和架构开始，先进行了解其基本的作用和实现，再与其他项目集成，最后我们再深挖源码，下面就开始我们的学习之旅把！

1.1什么是nacos?

一门技术的兴起一定会有它自己独特的功能，正如缓存、数据库、消息队列等一系列的中间件一样，nacos也有它自己独特的起源。

nacos主要是阿里的开源产品，伴随的是阿里的生产实践以及借鉴其他的注册中心而有的孵化品，在官网上我们可以看到有篇关于阿里巴巴服务注册中心产品的发展回顾，《阿里巴巴服务注册中心产品ConfigServer 10年技术发展回顾》，这篇文档的大概意思是在阿里的业务拓展下，最初的服务注册发现产品Eureka不再符合阿里的业务，于是在2018年左右，阿里开始了自研服务注册中心的道路，最开始也是借鉴Eureka的设计理念，往后推进时也添加上了一些自己的思考和阿里线上的具体实践，一步步的迭代，从最初的SDK，到单机版，再到集群一步步的解决了服务注册发现方面的一些问题，然后就形成了我们今天所看到的从ConfigServer进化而来的nacos。

	ConfigServer	Eureka
2008年	V1.0：单机版，定义了服务发现的领域模型	
2009年初	V1.5：应用和ConfigServer集群发布解耦	
2009年7月	V2.0：基于客户端模式同步数据，支持集群部署	
2010年底	V2.5：优化集群间数据同步模式，申请国家专利。	
2012年9月1号		Eureka1.0正式开源
2012年底	V3.0：支持session和data分层部署	
2014年	V3.5：支持异地多活等细分场景	
2015年		Eureka2.0架构升级方案公布
2017年	V4.0：支持data分片能力	
2018年7月		Eureka2.0架构升级宣布停止

在分布式系统中，有三个特性一致性、可用性、分区容错性，而注册中心必然处于分布式系统中，那么它必然要满足CAP原则。而Eureka和ConfigServer则是同属于AP类型的注册中心，他们两个在之后的业务拓展中拥有着相似的阻碍，而阿里巴巴则将ConfigServer的技术架构和生产环境的发现融合到了开源产品nacos中，继往开来在云原生、微服务的时代继续着发光发热。

上一节我们讨论了nacos的发展及一些基本的注册中心知识，这一节我们将对于它的功能和生态进行进一步的探讨。

1.2nacos功能和生态

对于我们学习技术来说，最重要的就是这门新的技术有什么样的功能，和他的生态是否强大。功能决定了它的业务适用性，任何的技术都是为业务而生。而生态则是技术的后备支持，比如说漏洞维护，功能新增等一系列的技术支持。很显然Nacos是阿里巴巴的产品，而且可以与多种技术进行集成，由此决定了它在微服务中的适用性。