**配置管理**

1. **介绍**

在分布式系统中，由于服务数量巨多，为了方便服务配置文件统一管理，实时更新，所以需要分布式配置中心组件。

1. **解决问题**

配置文件分散在各个项目里，不方便维护。

配置内容安全与权限，实际开发中，开发人员是不知道线上环境的配置的。

更新配置后，项目需要重启。

配置管理流程：

配置的权限管控、灰度发布、版本管理、格式检验和安全配置等一系列的配置管理相关的特性也是配置中心不可获取的一部分。

1. **对比**

常见的配置中心：Disconf，Spring Cloud Config，Apollo，Nacos。

Disconf

2014年7月百度开源的配置管理中心，同样具备配置的管理能力，不过目前已经不维护了，最近的一次提交是两年前了。

Spring Cloud Config

2014年9月开源，Spring Cloud 生态组件，可以和Spring Cloud体系无缝整合。

Apollo

2016年5月，携程开源的配置管理中心，具备规范的权限、流程治理等特性。

Nacos

2018年6月，阿里开源的配置中心，也可以做DNS和RPC的服务发现。



|  |  |  |
| --- | --- | --- |
| 对比项目/配置中心 | **apollo** | **nacos** |
| 开源时间 | 2016.5 | 2018.6 |
| 配置实时推送 | 支持（HTTP长轮询1s内） | 支持（HTTP长轮询1s内） |
| 版本管理 | 自动管理 | 自动管理 |
| 配置回滚 | 支持 | 支持 |
| 权限管理 | 支持 | 待支持 |
| 多集群多环境 | 支持 | 支持 |
| 监听查询 | 支持 | 支持 |
| 多语言 | Go,C++,Python,Java,.net,OpenAPI | Python,Java,Nodejs,OpenAPI |
| 分布式高可用最小集群数量 | Config*2+Admin*3+Portal\*2+Mysql=8 | Nacos\*3+MySql=4 |
| 配置格式校验 | 支持 | 支持 |
| 通信协议 | HTTP | HTTP |
| 数据一致性 | 数据库模拟消息队列，Apollo定时读消息 | HTTP异步通知 |
| 单机读（tps） | 9000 | 15000 |
| 单机写（tps） | 1100 | 1800 |

**结论：选用Nacos作为配置中心。但有一个缺点不支持权限管理。**

理由：(1) Nacos部署简化，Nacos整合了注册中心、配置中心功能，且部署相比apollo简单，方便管理和监控。

(2) apollo容器化较困难，Nacos有官网的镜像可以直接部署，总体来说，Nacos比apollo更符合KISS原则

1. **实战Demo**

配置加载优先级

当我们加载多个配置文件时，如果存在相同的key时，需要了解下有关配置加载优先级关系问题。

在使用Nacos配置的时候，只要有以下三类配置：

A：通过spring.cloud.nacos.config.shared-dataids定义的共享配置

B：通过spring.cloud.nacos.config.ext-config[n]定义的加载配置

C：通过内部规则（spring.cloud.nacos.config.prefix、spring.cloud.nacos.config.file-extension、spring.cloud.nacos.config.group这几个参数）拼接出来的配置

想用弄清楚这个几个配置的优先级

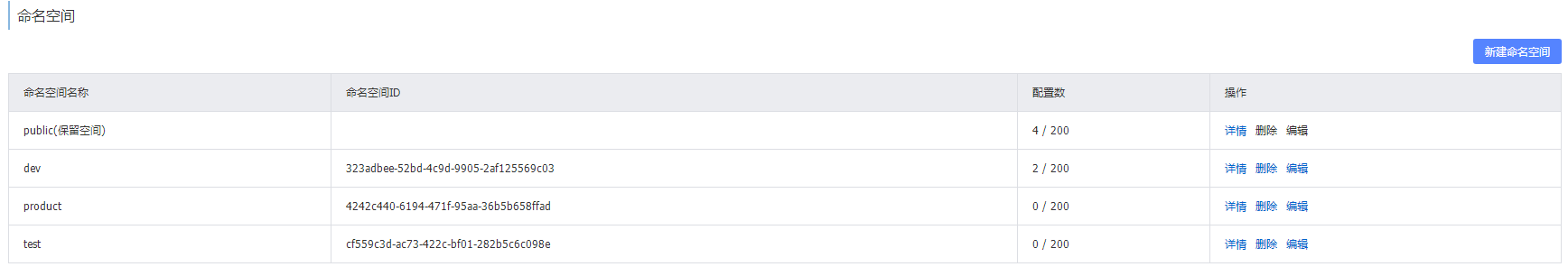
验证明后面加载的配置内容会覆盖之前的配置，所以优先级关系：C > B > A

1.引入依赖

|  |
| --- |
| <dependency>  <groupId>com.alibaba.cloud</groupId>  <artifactId>spring-cloud-alibaba-nacos-config</artifactId>  <version>2.1.0.RELEASE</version>  </dependency> |

|  |
| --- |
| #服务名  spring.application.name=nacos-config  #nacos服务地址  spring.cloud.nacos.config.server-addr=127.0.0.1:8848  #应用程序环境的namespace，默认public dev  #spring.cloud.nacos.config.namespace=323adbee-52bd-4c9d-9905-2af125569c03  spring.cloud.nacos.config.ext-config[0].data-id=application.yml  spring.cloud.nacos.config.ext-config[0].group=nacos-config  spring.cloud.nacos.config.ext-config[0].refresh=true  spring.cloud.nacos.config.ext-config[1].data-id=system.properties  spring.cloud.nacos.config.ext-config[1].group=nacos-config  spring.cloud.nacos.config.ext-config[1].refresh=true |

2.在nacos控制台添加命名空间



3.新建配置



4.添加bootstrap.properties

|  |
| --- |
| #服务名  spring.application.name=nacos-config  #nacos服务地址  spring.cloud.nacos.config.server-addr=127.0.0.1:8848  #应用程序环境的namespace，默认public dev  #spring.cloud.nacos.config.namespace=323adbee-52bd-4c9d-9905-2af125569c03  spring.cloud.nacos.config.ext-config[0].data-id=application.yml  spring.cloud.nacos.config.ext-config[0].group=nacos-config  spring.cloud.nacos.config.ext-config[0].refresh=true  spring.cloud.nacos.config.ext-config[1].data-id=system.properties  spring.cloud.nacos.config.ext-config[1].group=nacos-config  spring.cloud.nacos.config.ext-config[1].refresh=true |

1. **参考资料**

https://www.jianshu.com/p/afd7776a64c6

<https://www.jianshu.com/p/d78fa8412cbf>