

笔记: {chenqian}

1): Spring Cloud Alibaba介绍:

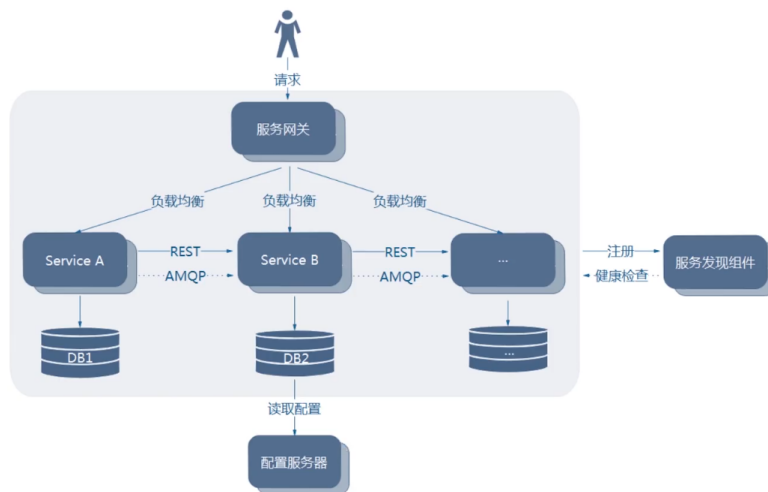
Spring Cloud Alibaba是Spring Cloud的一个子项目,功能主要有以下:

功能	产品	备注
服务限流降级	Sentinel	开源组件
服务注册与发现	Nacos ANS	开源组件 商业组件
分布式配置管理	Nacos ACM	开源组件 商业组件
消息驱动能力	Spring Cloud Stream RocketMQ	开源组件
分布式事务	Seata	开源组件, 不能用于生产(目前0.6.1), 1.0.0之后才可用于生产
阿里云对象存储	OSS	商业组件
分布式任务调度	SchedulerX	商业组件
阿里云短信服务	SMS	商业组件

2): 服务发现-Nacos

官方文档: <https://nacos.io/zh-cn/docs/what-is-nacos.html>

Nacos是一个服务发现组件,同时也是一个配置服务器。



Nacos快速开始

- 下载Nacos,地址:<https://github.com/alibaba/nacos/releases/tag/1.0.1>
- 搭建Nacos Server

Linux/Unix/Mac

启动命令(standalone代表着单机模式运行,非集群模式):

```
1 sh startup.sh -m standalone
```

如果您使用的是ubuntu系统,或者运行脚本报错提示[[符号找不到,可尝试如下运行:

```
1 bash startup.sh -m standalone
```

Windows

启动命令:

```
1 startup.cmd
```

或者双击startup.cmd运行文件。

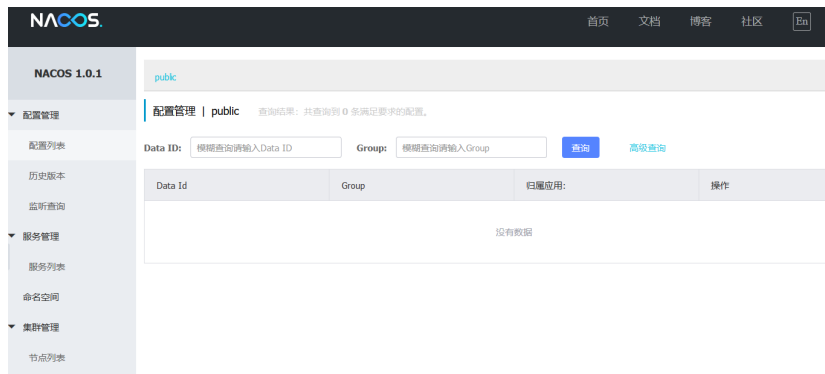
- 启动界面:

```
D:\nacos-server-1.0.1\nacos\bin>startup.cmd

Nacos

Nacos 1.0.1
Running in stand alone mode, All function modules
Port: 8848
Pid: 19344
Console: http://localhost:8848/nacos/index.html
https://nacos.io
```

- 登录Nacos管理控制台,用户名密码: nacos/nacos



Nacos Spring Cloud 快速开始

1. 添加依赖:

```
1 <dependency>
2   <groupId>org.springframework.cloud</groupId>
3   <artifactId>spring-cloud-starter-alibaba-nacos-config</artifactId>
4   <version>${latest.version}</version>
5 </dependency>
```

注意版本,版本对应关系参考: <https://github.com/alibaba/spring-cloud-alibaba/wiki/%E7%89%88%E6%9C%AC%E8%AF%B4%E6%98%8E>

2. 添加注解:

官方给出来的文档中,提供了添加注解的说明. 早期版本需要添加注解,现如今这个注解不是必须添加.

- ii. 通过 Spring Cloud 原生注解 `@EnableDiscoveryClient` 开启服务注册发现功能:

```
@SpringBootApplication
@EnableDiscoveryClient
public class NacosProviderApplication {
```

3. 添加配置

spring:

cloud:

nacos:

discovery:

指定nacos server的地址

server-addr: localhost:8848

application:

服务名称尽量用-, 不要用_, 不要用特殊字符

name: user-center

4. 管理界面查看服务注册情况

服务详情

配置管理

配置列表

历史版本

部署操作

服务管理

服务列表

命名空间

集群管理

节点列表

服务名:

分群:

保护策略:

元数据:

服务路由类型:

集群: DEFAULT

编辑配置

下线

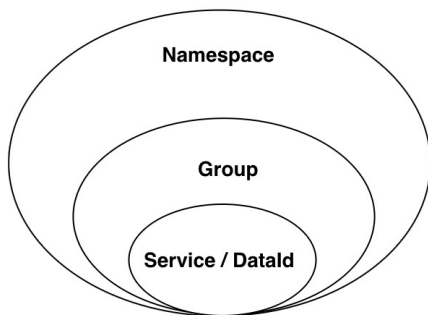
IP	端口	应用实例	权重	健康状态	元数据	操作
102.168.1.102	8080	true	1	true	reserved.register.source=SPRING_CLOUD	<div>编辑</div> <div>下线</div>

3): Nacos领域模型

数据模型

Nacos 数据模型 Key 由三元组唯一确定, Namespace默认是空串, 公共命名空间 (public), 分组默认是 DEFAULT_GROUP

Nacos data model



服务领域模型



Namespace:默认名称public,主要用来实现不同环境之间的隔离.

Group:默认是DEFAULT_GROUP。

Service:微服务.

Cluster:对指定微服务的一个虚拟划分,默认DEFAULT.例如对某个微服务做集群,以提高容灾能力及性能.

Instance:微服务事例.

配置领域模型

围绕配置，主要有两个关联的实体，一个是配置变更历史，一个是服务标签（用于打标分类，方便索引），由 ID 关联。

public | dev

多出新配置的dev环境

服务列表 | dev 56116141-d837-4d15-8842-94e153bb6cfb

服务名称 查询 创建服务

服务名	分组	集群数目	实例数	健康实例数	操作
user-center	DEFAULT_GROUP	1	1	1	详情 示例代码 删除

服务路由类型:

集群: NJ

集群配置

IP	端口	临时实例	权重	健康状态	元数据	操作
	8080	true	1	true	preserved.register.source=SP	<button>编辑</button> <button>下线</button>

Nacos元数据

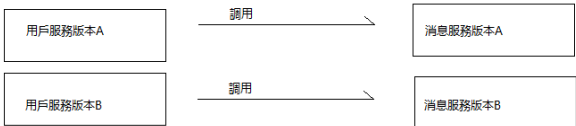
Nacos数据（如配置和服务）描述信息，如服务版本、权重、容灾策略、负载均衡策略、鉴权配置、各种自定义标签 (label)，从作用范围来看，分为服务级别的元信息、集群的元信息及实例的元信息。

作用：

提供描述信息

让微服务调用更加灵活：

如微服务版本控制



用户服务版本A与消息服务版本B的版本不兼容，不能调用

对以上情况,可以使用元数据配置解决.

如何为微服务设置元数据?

1. 管理控制台设置,不做介绍
2. 配置文件制定

```
# cluster-name: NJ
metadata:
  instance: c
  haha: hehe
  version: v1
```

集群: DEFAULT

集群配置

IP	端口	临时实例	权重	健康状态	元数据	操作
192.168.1.109	8080	true	1	true	haha=hehe instance=c preserved.register.source=SPRING_CLOUD version=v1	<button>编辑</button> <button>下线</button>

如何使用Namespace实现环境之间的隔离？
如何使用Cluster实现相同机房的优先调用？
如何使用元数据实现以上提到版本之间控制？
请期待下一篇Ribbon笔记.....