

Spring Cloud Alibaba：Nacos 作为注册中心和配置中心使用

Spring Cloud Alibaba 致力于提供微服务开发的一站式解决方案，Nacos 作为其核心组件之一，可以作为注册中心和配置中心使用，本文将对其用法进行详细介绍。

Nacos简介

Nacos 致力于帮助您发现、配置和管理微服务。Nacos 提供了一组简单易用的特性集，帮助您快速实现动态服务发现、服务配置、服务元数据及流量管理。

Nacos 具有如下特性：

- 服务发现和服务健康监测：支持基于DNS和基于RPC的服务发现，支持对服务的实时的健康检查，阻止向不健康的主机或服务实例发送请求；
- 动态配置服务：动态配置服务可以让您以中心化、外部化和动态化的方式管理所有环境的应用配置和服务配置；
- 动态 DNS 服务：动态 DNS 服务支持权重路由，让您更容易地实现中间层负载均衡、更灵活的路由策略、流量控制以及数据中心内网的简单DNS解析服务；
- 服务及其元数据管理：支持从微服务平台建设的视角管理数据中心的所有服务及元数据。

使用Nacos作为注册中心

安装并运行Nacos

- 我们先从官网下载Nacos，这里下载的是 `nacos-server-1.1.4.zip` 文件，下载地址：<https://github.com/alibaba/nacos/releases>
- 配置 `JAVA_HOME` 环境变量，不配置会导致无法运行Nacos；

```
1 JAVA_HOME=D:\developer\env\Java\jdk1.8.0_91Copy to clipboardErrorCopied
```

- 解压安装包，直接运行 `bin` 目录下的 `startup.cmd` ；
- 运行成功后，访问 <http://localhost:8848/nacos> 可以查看Nacos的主页，默认账号密码都是 `nacos`。



创建应用注册到Nacos

我们通过改造consul-user-service和consul-ribbon-service来演示下服务注册与发现的功能，主要是将应用原来的Consul注册中心支持改为Nacos注册中心支持。

- 创建nacos-user-service模块和nacos-ribbon-service模块；
- 如果要使用Spring Cloud Alibaba 的组件都需要在pom.xml中添加如下的配置；

```
1      <dependencyManagement>
2          <dependencies>
3              <dependency>
4                  <groupId>com.alibaba.cloud</groupId>
5                  <artifactId>spring-cloud-alibaba-dependencies</artifactId>
6                  <version>2.1.0.RELEASE</version>
7                  <type>pom</type>
8                  <scope>import</scope>
9              </dependency>
10         </dependencies>
11     </dependencyManagement>Copy to clipboardErrorCopied
```

- 修改相关依赖，把原来的Consul注册发现的依赖改为Nacos的：

```
1     <dependency>
2         <groupId>com.alibaba.cloud</groupId>
3         <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
4     </dependency>Copy to clipboardErrorCopied
```

- 修改配置文件application.yml，将Consul的注册发现配置改为Nacos的：

```
1     server:
2         port: 8206
3     spring:
4         application:
5             name: nacos-user-service
```

```

6     cloud:
7       nacos:
8         discovery:
9           server-addr: localhost:8848 #配置Nacos地址
10    management:
11      endpoints:
12        web:
13          exposure:
14            include: '*'Copy to clipboardErrorCopied

```

- 运行两个nacos-user-service和一个nacos-ribbon-service，在Nacos页面上可以看到如下信息：

The screenshot shows the Nacos 1.1.4 web interface. The top navigation bar includes links for '首页', '文档', '博客', '社区', and a language switcher 'En'. The left sidebar has a 'NACOS 1.1.4' header and a menu with '配置管理' (Configuration Management) and '服务管理' (Service Management). Under '服务管理', '服务列表' (Service List) is highlighted. The main content area shows the 'public' namespace. It includes input fields for '服务名称' (Service Name) and '分组名称' (Group Name), a '隐藏空服务' (Hide Empty Services) toggle, and buttons for '查询' (Query) and '创建服务' (Create Service). Below these is a table listing services:

服务名	分组名称	集群数目	实例数	健康实例数	触发保护阈值	操作
nacos-ribbon-service	DEFAULT_GROUP	1	1	1	false	详情 示例代码 删除
nacos-user-service	DEFAULT_GROUP	1	2	2	false	详情 示例代码 删除

负载均衡功能

由于我们运行了两个nacos-user-service，而nacos-ribbon-service默认会去调用它的接口，我们调用nacos-ribbon-service的接口来演示下负载均衡功能。

多次调用接口：<http://localhost:8308/user/1>，可以发现两个nacos-user-service的控制台交替打印如下信息。

```

1  2019-11-06 14:28:06.458 INFO 12092 --- [nio-8207-exec-2]
   c.macro.cloud.controller.UserController : 根据id获取用户信息，用户名称为: macroCopy to
   clipboardErrorCopied

```

使用Nacos作为配置中心

我们通过创建nacos-config-client模块，并在Nacos页面中添加配置信息来演示下配置管理的功能。

创建nacos-config-client模块

- 在pom.xml中添加相关依赖：

```
1 <dependency>
2     <groupId>com.alibaba.cloud</groupId>
3     <artifactId>spring-cloud-starter-alibaba-nacos-config</artifactId>
4 </dependency>
5 <dependency>
6     <groupId>com.alibaba.cloud</groupId>
7     <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
8 </dependency>Copy to clipboardErrorCopied
```

- 添加配置文件application.yml，启用的是dev环境的配置：

```
1 spring:
2   profiles:
3     active: devCopy to clipboardErrorCopied
```

- 添加配置文件bootstrap.yml，主要是对Nacos的作为配置中心的功能进行配置：

```
1 server:
2   port: 9101
3 spring:
4   application:
5     name: nacos-config-client
6   cloud:
7     nacos:
8       discovery:
9         server-addr: localhost:8848 #Nacos地址
10      config:
11        server-addr: localhost:8848 #Nacos地址
12        file-extension: yaml #这里我们获取的yaml格式的配置Copy to
    clipboardErrorCopied
```

- 创建ConfigClientController，从Nacos配置中心中获取配置信息：

```
1 /**
2  * Created by macro on 2019/9/11.
3  */
4 @RestController
5 @RefreshScope
6 public class ConfigClientController {
7
8     @Value("${config.info}")
9     private String configInfo;
10
11     @GetMapping("/configInfo")
12     public String getConfigInfo() {
13         return configInfo;
14     }
15 }Copy to clipboardErrorCopied
```

在Nacos中添加配置

- 我们先来讲下Nacos中的dataid的组成格式及与SpringBoot配置文件中的属性对应关系：

```
1 ${spring.application.name}-${spring.profiles.active}.${spring.cloud.nacos.config.file-extension}Copy to clipboardErrorCopied
```

- 比如说我们现在要获取应用名称为 `nacos-config-client` 的应用在 `dev` 环境下的 `yaml` 配置，dataid如下：

```
1 nacos-config-client-dev.yamlCopy to clipboardErrorCopied
```

- 按照以上dataid添加如下配置：

```
1 config:
2   info: "config info for dev"Copy to clipboardErrorCopied
```

- 填写配置示意图：

The screenshot shows the Nacos web interface for creating a new configuration. The left sidebar has '配置列表' (Configuration List) highlighted with a red box and labeled '1'. The main area is titled '新建配置' (New Configuration). The 'Data' field contains 'nacos-config-client-dev.yaml' and is highlighted with a red box and labeled '2'. The 'Group' is 'DEFAULT_GROUP'. The 'Format' is 'YAML', highlighted with a red box. The 'Content' field contains the YAML configuration: 'config: info: \"config info for dev\"', highlighted with a red box and labeled '4'. At the bottom right, the '发布' (Publish) button is highlighted with a red box and labeled '5'.

- 启动nacos-config-client，调用接口查看配置信息：<http://localhost:9101/configInfo>

```
1 config info for devCopy to clipboardErrorCopied
```

Nacos的动态刷新配置

我们只要修改下Nacos中的配置信息，再次调用查看配置的接口，就会发现配置已经刷新，Nacos和Consul一样都支持动态刷新配置。当我们在Nacos页面上修改配置并发布后，应用会刷新配置并打印如下信息

```
1 2019-11-06 14:50:49.460 INFO 12372 --- [-localhost_8848]
   trationDelegate$BeanPostProcessorChecker : Bean
   'org.springframework.cloud.autoconfigure.ConfigurationPropertiesRebinderAutoConfig
   uration' of type
   [org.springframework.cloud.autoconfigure.ConfigurationPropertiesRebinderAutoConfig
   uration$$EnhancerBySpringCGLIB$$395f8e] is not eligible for getting processed by
   all BeanPostProcessors (for example: not eligible for auto-proxying)
2 2019-11-06 14:50:49.608 INFO 12372 --- [-localhost_8848]
   c.a.c.n.c.NacosPropertySourceBuilder : Loading nacos data, dataId: 'nacos-
   config-client-dev.yaml', group: 'DEFAULT_GROUP'
3 2019-11-06 14:50:49.609 INFO 12372 --- [-localhost_8848]
   b.c.PropertySourceBootstrapConfiguration : Located property source:
   CompositePropertySource {name='NACOS', propertySources=[NacosPropertySource
   {name='nacos-config-client-dev.yaml'}, NacosPropertySource {name='nacos-config-
   client.yaml'}]}
4 2019-11-06 14:50:49.610 INFO 12372 --- [-localhost_8848]
   o.s.boot.SpringApplication : The following profiles are active: dev
5 2019-11-06 14:50:49.620 INFO 12372 --- [-localhost_8848]
   o.s.boot.SpringApplication : Started application in 0.328 seconds
   (JVM running for 172.085)
6 2019-11-06 14:50:49.638 INFO 12372 --- [-localhost_8848]
   o.s.c.e.event.RefreshEventListener : Refresh keys changed: [config.info]Copy
   to clipboardErrorCopied
```

参考资料

Spring Cloud Alibaba 官方文档: <https://github.com/alibaba/spring-cloud-alibaba/wiki>

使用到的模块

```
1 springcloud-learning
2 |—— nacos-config-client -- 用于演示nacos作为配置中心的nacos客户端
3 |—— nacos-user-service -- 注册到nacos的提供User对象CRUD接口的服务
4 |—— nacos-ribbon-service -- 注册到nacos的ribbon服务调用测试服务
```