题目标签

- 题目难度：简单
- 知识点标签：分布式，nacos，微服务
- 课程知识点：springcloud 上下篇
- 课程时间：30分钟

方案背景

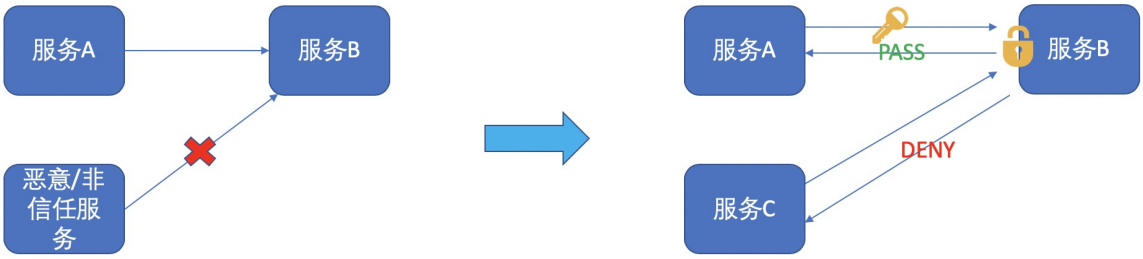
分布式系统中需要对未知的或者不受信任的请求或者服务的来源、请求进行识别和拒绝。权限控制一般分为两个阶段：身份识别（Authentication）和权限识别（Authorization）。身份认证主要确定访问者的身份，权限识别则判断这个访问者是否有对应资源的权限。

那么在分布式中如何在根本上控制身份和权限的识别尤为重要。

场景重现

在Nacos的场景中，配置管理的权限控制指的是设置某个配置能否被某个用户读写，这个比较好理解，没有权限的用户旧无法读取或者写入对应的配置。服务发现的权限控制指的是用户是否有权限进行某个服务的注册或者订阅，这里需要注意的是服务发现的权限控制只能够控制用户是否可以从Nacos获取到服务的地址或者在Nacos上修改服务的地址。但是如果已经获取到了服务的地址，Nacos无法在服务真正调用时进行权限控制，这个时候的权限控制需要由服务框架来完成。

正常应该如下图



常见实现方式

认证（Authentication）

- 用户名+密码
- Cookie（只适用于浏览器）
- Session
- Token（JWT，Oauth，LDAP，SAML，OpenID）
- AK/SK

鉴权（Authorization）

- ACL： 规定**资源**可以被哪些**主体**进行哪些操作；
- DAC： 规定**资源**可以被哪些**主体**进行哪些操作 同时，**主体**可以将**资源**的权限，授予其他**主体**；
- MAC：a. 规定**资源**可以被哪些类别的**主体**进行哪些**操作** b. 规定**主体**可以对哪些等级的**资源**进行哪些**操作** 当一个**操作**，同时满足a与b时，允许**操作**；

- RBAC： a. 规定**角色**可以对哪些**资源**进行哪些**操作** b. 规定**主体**拥有哪些**角色**当一个操作，同时满足a与b时，允许**操作**；
- ABAC： 规定哪些**属性**的**主体**可以对哪些**属性**的**资源**在哪些**属性**的情况下进行哪些**操作**

Nacos权限控制使用

安装Nacos 1.2.0

1. 部署包准备。可以直接下载安装包：https://github.com/alibaba/nacos/releases/tag/1.2.0，也可以将Nacos master分支clone下来进行源码编译：

```
mvn -Prelease-nacos -Dmaven.test.skip=true clean install -U
```

1. 安装包解压，然后使用distribution/nacos-mysql.sql进行数据库初始化，主要是新增了users, roles, permissions三张表，standalone模式使用distribution/schema.sql进行初始化。
2. Server端打开权限控制开关。修改con/application.properties内容：

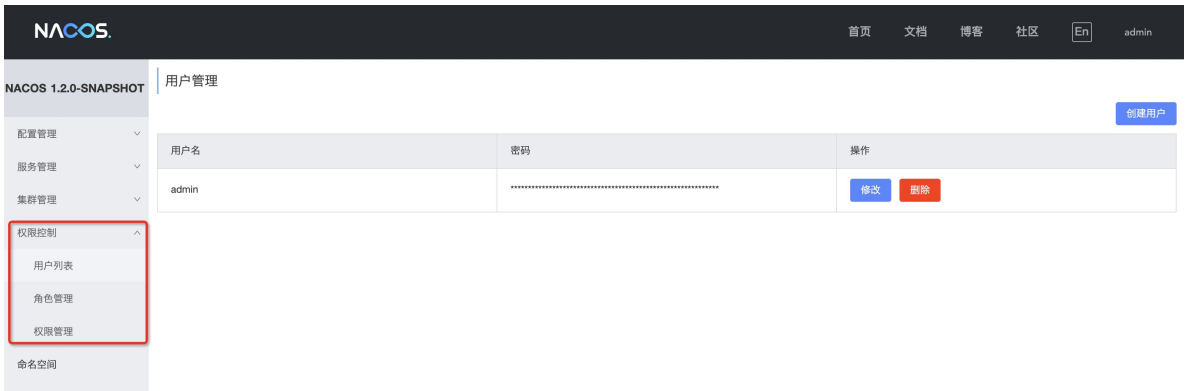
```
nacos.core.auth.enabled=true
```

这个开关采用了热加载模式，无需重启Server即可生效。因此当权限控制功能使用有异常时，可以直接回滚到不鉴权的模式。

注意： Nacos 1.2.0里登录和鉴权是绑定关系，而由于这个开关的默认值为false，因此默认启动时，是没有登录界面的，这点请读者注意。

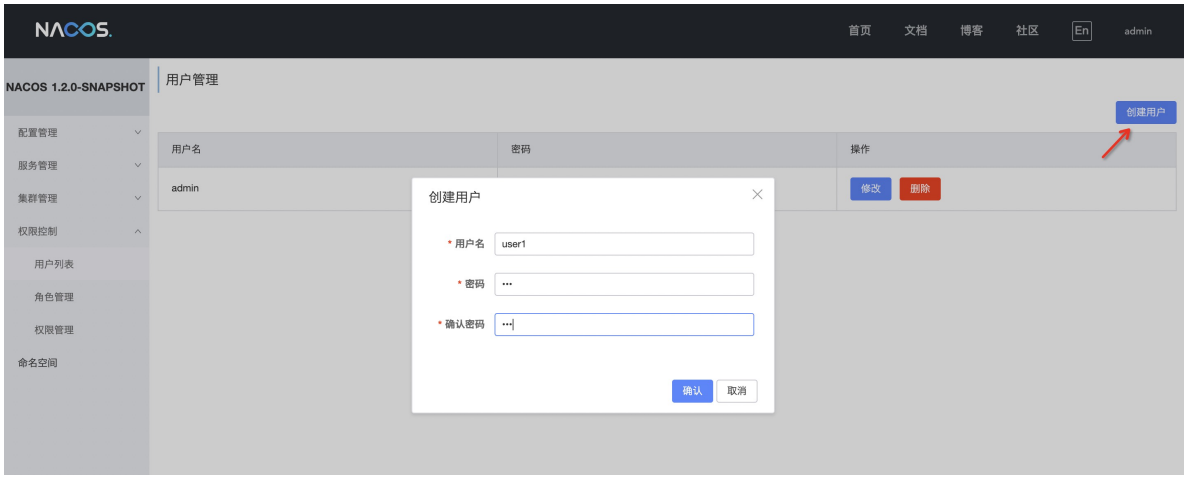
使用权限控制

1. 使用管理员账号登录Nacos控制台（如果页面提示错误，可以清空浏览器缓存刷新页面）：



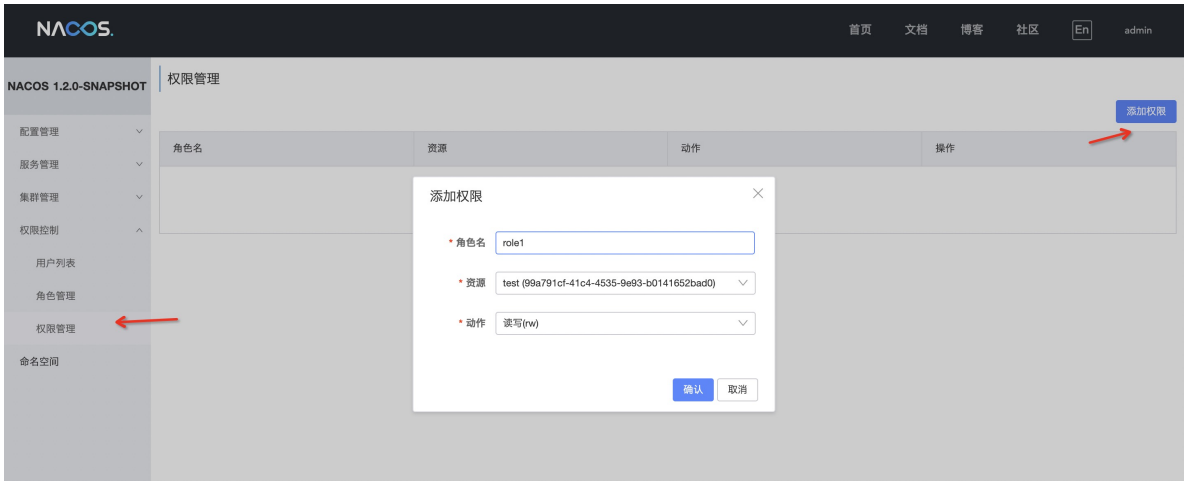
可以看到，左侧边栏增加了一个父菜单和三个子菜单，分别用于权限控制里的用户创建、角色创建以及权限管　理。这个菜单栏只会在管理员登录的时候显示，也就意味着只有管理员才能进行权限的管理和分配。

1. 管理用户。点击“用户列表”，进入用户管理页面，可以进行用户的创建、修改和删除：



1. 管理角色。因为Nacos的自带的权限是基于角色来进行分配的，因此需要给创建好的用户绑定一些角色：

1. 管理权限。角色创建好以后，就可以给这个角色赋予特定的权限了：

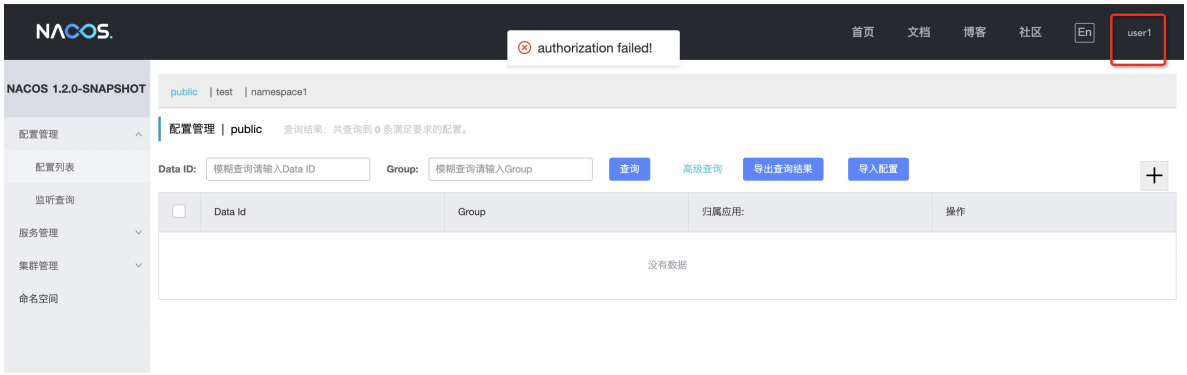


在“添加资源”对话框里，可以选择绑定的角色，命名空间资源以及对应的动作类型，例如在上图中，我们给角色role1绑定命名空间test的读写权限。然后又因为刚刚我们是将user1绑定到了role1上，那么user1这个用户就可以对test这个命名空间的资源进行读写操作了。

1. 使用user1登录控制台。点击控制台右上角，退出admin账号，然后用刚才创建的user1进行登录：

image.png

如上图所示，首先是左侧的权限管理菜单消失了，因为当前用户不是管理员。其次是会弹出一个鉴权失败的提示框。不用担心，这个提示框意思是user1没有public命名空间的读权限，所以会弹出，但是不影响我们将命名空间切换到test：



如上图所示，我们可以看到test命名空间的配置数据了，下面我们再来介绍客户端的使用。

1. 首先依赖最新的nacos 1.2.0客户端，然后在初始化时添加如下代码：

```
Properties properties = new Properties();
properties.put(PropertyKeyConst.NAMESPACE, "99a791cf-41c4-4535-9e93-
b0141652bad0");
properties.put(PropertyKeyConst.SERVER_ADDR, "127.0.0.1:8848");
// 配置用户名：
properties.put(PropertyKeyConst.USERNAME, "user1");
// 配置密码：
properties.put(PropertyKeyConst.PASSWORD, "pwd1");
ConfigService iconfig = NacosFactory.createConfigService(properties);
```

注意事项

因为nacos的权限控制当前并不稳定，所以使用时需要注意，service版本也要使用1.2.0以及以上的。而在springcloud中alibaba的版本也要使用最新的2.2.3的。不然会没有这个用户和密码属性。