

1, 基于docker安装nacos

1)拉取最新版本的nacos

```
>docker pull nacos/nacos-server
```

2)查看镜像

```
>docker images
```

3) 启动nacos,我这里做了端口映射, 映射端口为8091

```
> docker run --env MODE=standalone --name nacos -d -p 8091:8848 nacos/nacos-server
```

4) 访问nacos,默认用户和密码均为nacos, 访问http://127.0.0.1:8091/nacos

2, 测试服务注册发现, 和创建服务提供者

1) 创建microservice-nacos-provider工程, 作为服务提供者

2) 创建服务提供者

3) 创建配置文件bootstrap.yml,因为加载顺序的问题, 此处必须得是bootstrap.yml,而不是application.yml, 否则nacos config会启动失败

4) 创建服务提供的具体业务逻辑

5) 在nacos的控制台创建配置文件nacos-service-dev.yaml



6)分别用8091, 8092两个端口启动microservice-nacos-provider生产者项目, 创建集群, 在nacos管理界面, 查看服务列表是否注册成功;

3, 创建服务消费者

1) 创建microservice-nacos-customer工程

2) 创建消费者启动类, 在启动类上加上@EnableFeignClients注解, 用于开启openFeign客户端

3) 创建消费者NacosCustomerServer类, 用户调用生产者接口

4) 创建服务消费者的bootstrap.yml

5) 创建openFeign客户端，NacosServiceFeign

4, 启动消费者，并调用http://127.0.0.1:9001/getTestResult验证结果

--注：1,具体代码参考：<https://github.com/birkhoff-liu/spring-cloud/tree/master/nacos>

2, 在bootstrap.yml文件中配置nacos注册中心地址时，注意结果要以"/"结束，

正确地址：http://127.0.0.1:8001/

NacosDiscoveryProperties.java代码片段如下：

```
186 metadata.put(PreservedMetadataKeys.REGISTER_SOURCE, "SPRING_CLOUD");
187 if (secure) {
188     metadata.put("secure", "true");
189 }
190
191 serverAddr = Objects.toString(serverAddr, nullDefault: "");
192 if (serverAddr.lastIndexOf( str: "/" ) != -1) {
193     serverAddr = serverAddr.substring(0, serverAddr.length() - 1);
194 }
195 endpoint = Objects.toString(endpoint, nullDefault: "");
196 namespace = Objects.toString(namespace, nullDefault: "");
197 logName = Objects.toString(logName, nullDefault: "");
198
```