部署rest服务

- 我使用S2I的方式部署rest服务，主要是为了演示用jar和war的方式处理spring boot都可以，差别不大
- 参考文档：https://blog.codecentric.de/en/2016/03/deploy-spring-boot-applications-openshift/
- oc new-app codecentric/springboot-maven3-centos~https://github.com/caishu/gs-sample.git 这个是在linux上执行的；注意用developer用户执行（如果不确认自己是哪个用户，可以用oc whoami看看自己是谁；或者直接oc login -u developer再登录一次）
- 这步会用springboot-maven3-centos这个s2i镜像，创建一个openshift的服务
- oc expose sec gs-sample 。这步是给刚才部署的服务添加一个route
- 这时候从mac浏览器访问 http://gs-sample-myproject.127.0.0.1.nip.io/greeting，就能看到返回的json串了。这里，注意这个域名，是从mac访问linux时候用的。这个域名需要解析成127.0.0.1，如果解析错误，可以在mac的/etc/hosts里边加一个记录
- 查看gs-sample服务的详情：https://127.0.0.1:8443/console/project/myproject/browse/services/gs-sample?tab=details 。这里可以看到Hostname: gs-sample.myproject.svc 这个域名是openshift内部pod之间访问时候用的。这个需要记下来（实际这个是自动生成的，规则是service_name+project_name+’svc’），一会创建的另一个消费这个REST的程序里用这个地址

创建消费rest的工程

- 参考这个：https://spring.io/guides/gs/consuming-rest/
- 我从这里开始创建工程的：https://start.spring.io/
- 我选了打包格式是war，另外选择了“web”和”actuator”
- 我用的工程代码在这里：https://github.com/caishu/rs-consumer.git
- 注意QuoteController.java的22行，用的地址是刚才部署的服务的内部域名；端口是8080
- 另外注意代码里我添加了WEB-INF/jboss-web.xml这个文件。这个文件是把这个war映射到/的访问；否则wildfly自动部署war会用war文件名做uri的路径
- 这一步的验证办法是，mvn clean package看到BUILD SUCCESS就可以
- 另外注意这里我还定义了一个新的rest，访问这个时候，会调用刚才部署的greeting