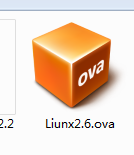
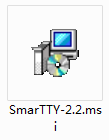
springboot+dubbo+Zookeeper多实例分布式项目

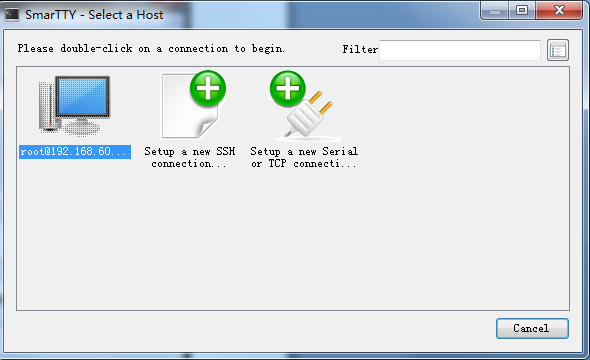
首先，我们先进行liunx环境搭建，我们使用的是Docker镜像环境来进行支持软件安装。

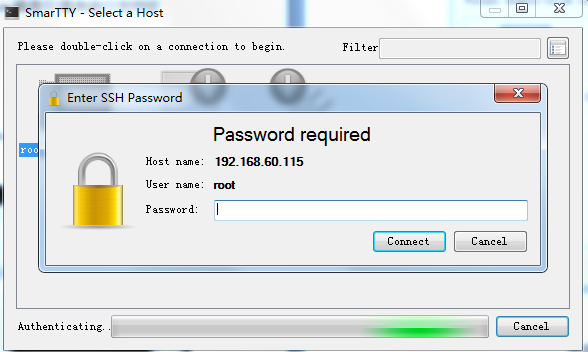
1.首先先安装虚拟机，安装好了之后导入Linux系统，导入完成之后如下图：



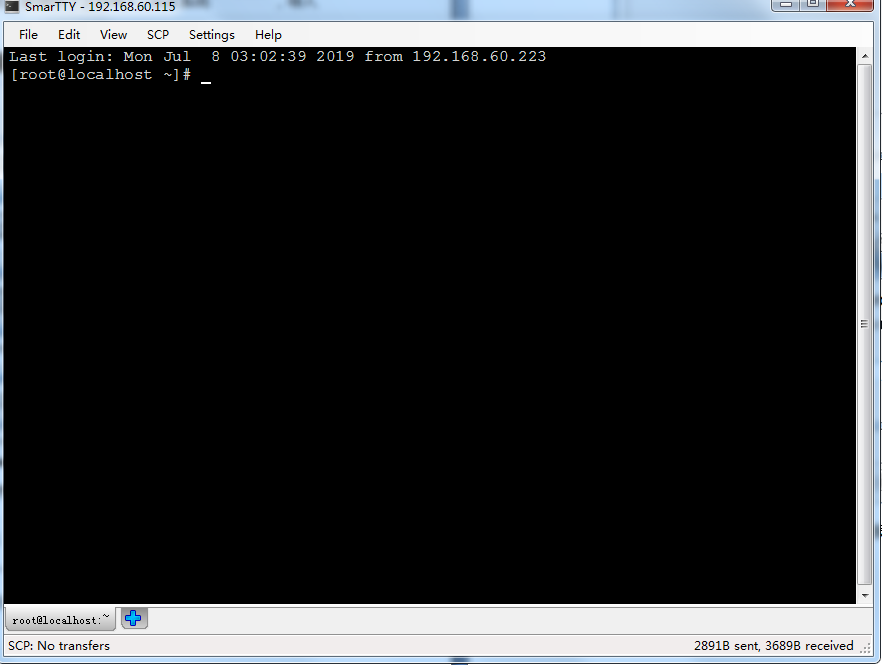
Liunx虚拟机界面（用户名root密码123456）

2.虚拟机安装完成后安装，SmarTTY是一款免费的多标签SSH客户端，支持使用SCP命令随时复制文件和目录。安装完毕之后，输入虚拟机的ip和密码：（如下图：）

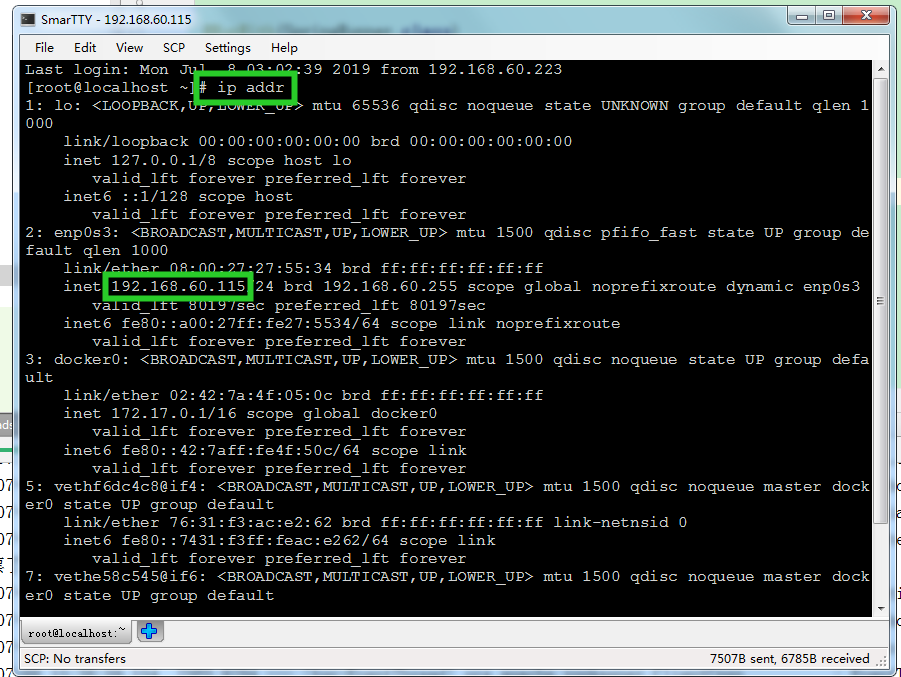




使用客户端连接linux服务器进行命令操作；

连接成功后如下图：

通过SmarTTY我们就可以进行liunx命令来进行操作了；例如使用 ip addr命令可以查看虚拟机ip（如下图）：



3. 在linux虚拟机上安装docker

安装docker命令如下（红字部分为命令）：

1、检查内核版本，必须是3.10及以上

uname ‐r

2、安装docker

yum install docker

3、输入y确认安装

4、启动docker

[root@localhost ~]# systemctl start docker

[root@localhost ~]# docker ‐v

Docker version 1.12.6, build 3e8e77d/1.12.6

5、开机启动docker

[root@localhost ~]# systemctl enable docker

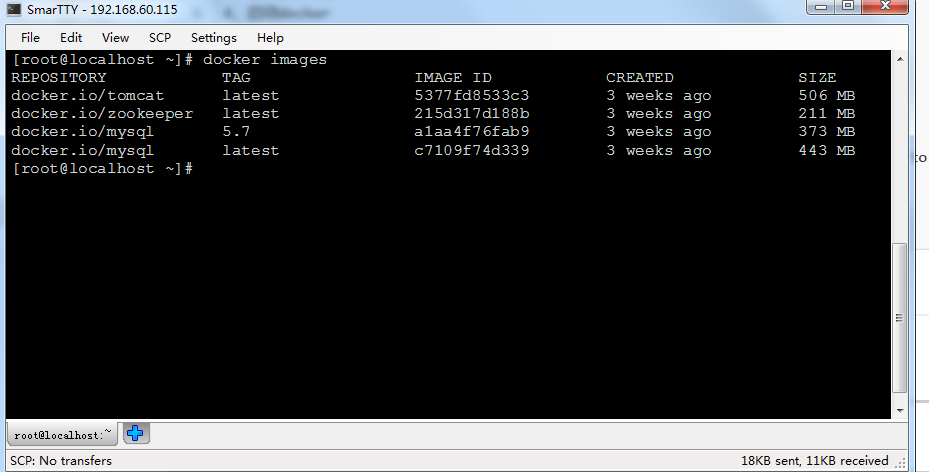
Created symlink from /etc/systemd/system/multi‐user.target.wants/docker.service to

/usr/lib/systemd/system/docker.service.

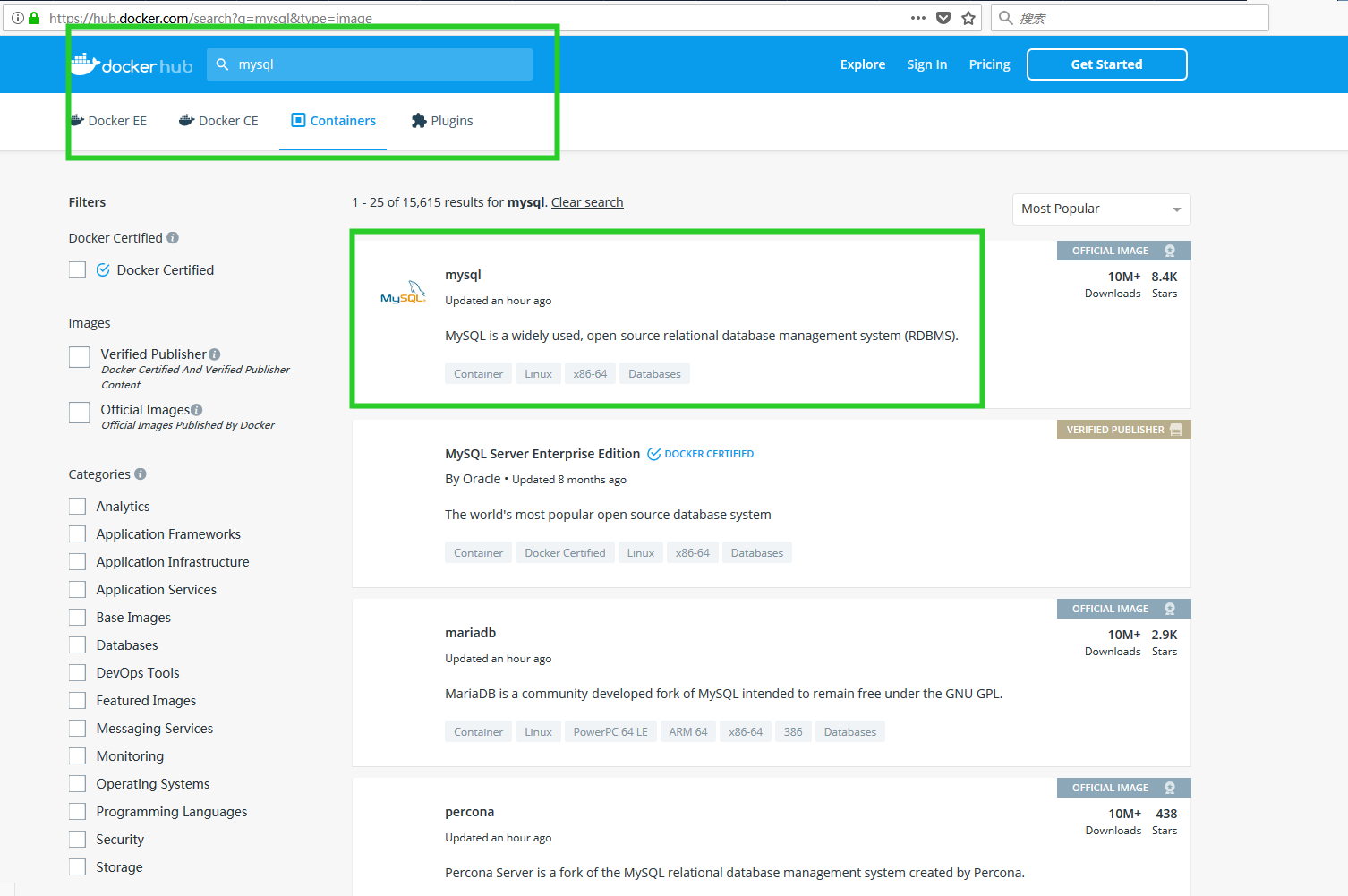
6、停止docker

systemctl stop docker

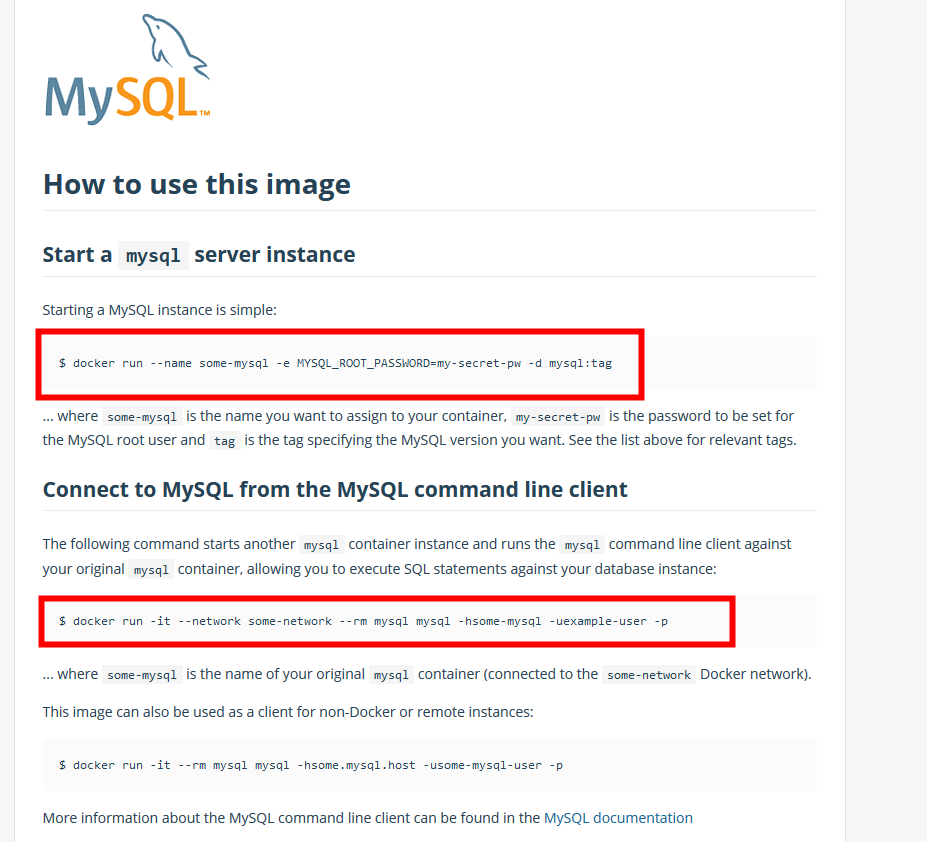
通过使用docker images命令可查看到安装软件列表：（下图为安装了一个默认版本的tomcat。默认版本的zookeeper，默认版本的mysql，和一个5.7版本的mysql）



需要那个软件直接可以去<https://hub.docker.com/>查找，下图为网站使用方法截图：这里以mysql为例：



点击进入sql之后，进入sql安装实例，如下图：



第一个命令为简单的链接到虚拟机mysql，第2个命令为带网络映射的虚拟机mysql命令，通过第2个命令我们可以在使用远程Navicate进行网络链接，实现远程mysql的链接，

[root@localhost ~]# docker run ‐p 3306:3306 ‐‐name fw-mysql03 ‐e MYSQL\_ROOT\_PASSWORD=123456 ‐d mysql

ad10e4bc5c6a0f61cbad43898de71d366117d120e39db651844c0e73863b9434

如下图：



下面为docker常用的一些命令，这些常用命令可以通过百度很容易查询出来；



Docker容器操作常用命令：

1、搜索镜像

[root@localhost ~]# docker search tomcat

2、拉取镜像

[root@localhost ~]# docker pull tomcat

3、根据镜像启动容器

docker run ‐‐name mytomcat ‐d tomcat:latest

4、docker ps

查看运行中的容器

5、 停止运行中的容器

docker stop 容器的id

6、查看所有的容器

docker ps ‐a

7、启动容器

docker start 容器id

8、删除一个容器

docker rm 容器id

9、启动一个做了端口映射的tomcat

[root@localhost ~]# docker run ‐d ‐p 8888:8080 tomcat

‐d：后台运行

‐p: 将主机的端口映射到容器的一个端口 主机端口:容器内部的端口

10、关闭linux的防火墙

service firewalld status ；查看防火墙状态

service firewalld stop：关闭防火墙

11、查看容器的日志

docker logs container‐name/container‐id

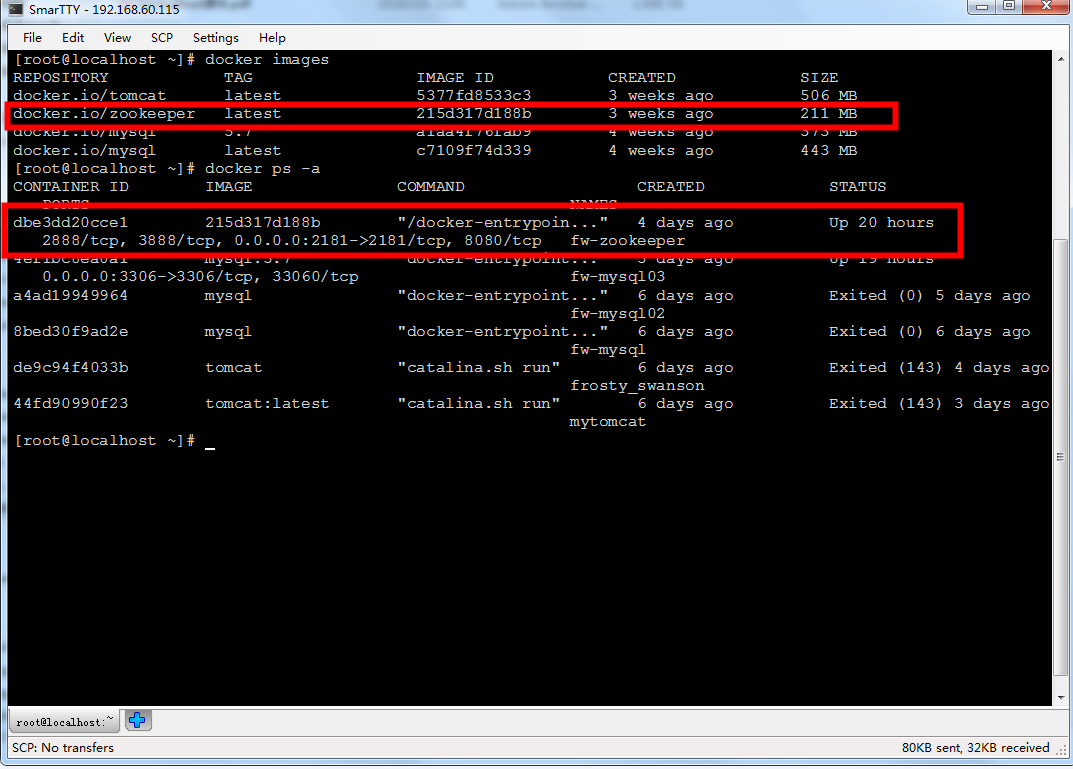
更多命令参看

https://docs.docker.com/engine/reference/commandline/docker/

可以参考每一个镜像的文档

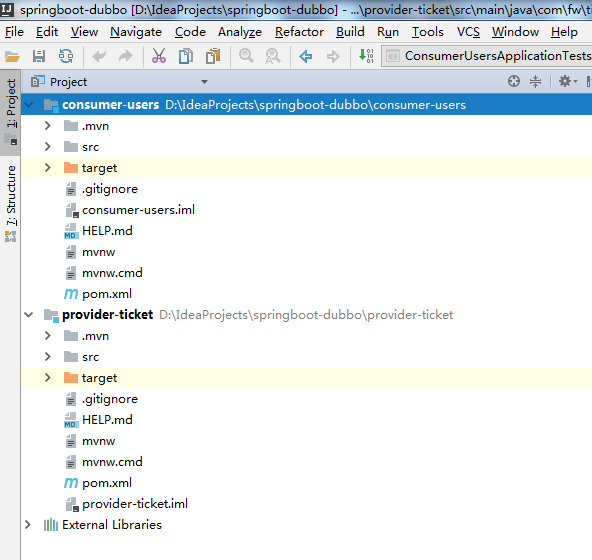


安装好tomcat和mysql测试运行正常级可安装zookeeper，搭建ZooKeeper 作为 Dubbo 的注册中心。安装zookeeper过程和tomcat一样。安装好zookeeper镜像和容器如下图：



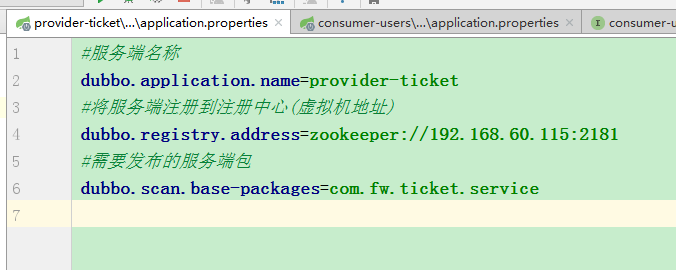
整合springboot+dubbo多实例项目：

1. 首先利用idea搭建两个子项目，分别是users项目和ticket项目，users指用户项目，ticket项目指票务项目。模块案例很简单，用户需要通过票务系统进行购票，才可以看电影。下图为项目列表：

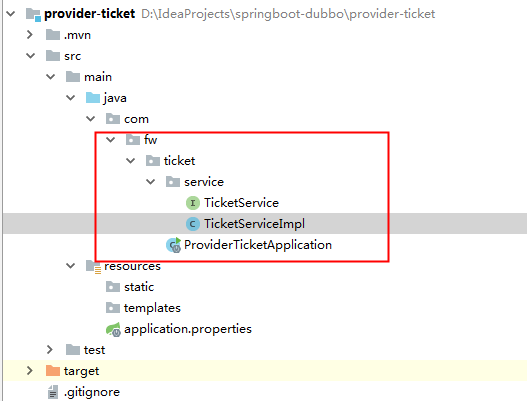


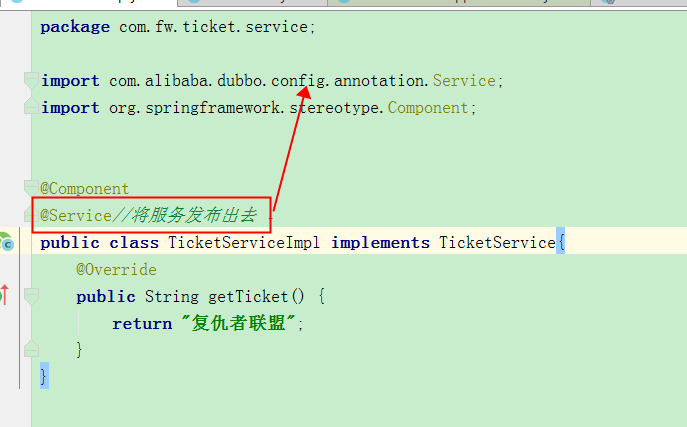
1. 搭建好项目之后在pom文件里添加dubbo支持和引入zookeeper的客户端工具。（以下为项目pom配置文件）

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-parent</**artifactId**>  
 <**version**>2.1.6.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
 <**groupId**>com.fw</**groupId**>  
 <**artifactId**>provider-ticket</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**name**>provider-ticket</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**properties**>  
 <**java.version**>1.8</**java.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>com.alibaba.boot</**groupId**>  
 <**artifactId**>dubbo-spring-boot-starter</**artifactId**>  
 <**version**>0.1.0</**version**>  
 </**dependency**>  
 *<!--引入zookeeper的客户端工具-->  
 <!-- https://mvnrepository.com/artifact/com.github.sgroschupf/zkclient -->* <**dependency**>  
 <**groupId**>com.github.sgroschupf</**groupId**>  
 <**artifactId**>zkclient</**artifactId**>  
 <**version**>0.1</**version**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
</**project**>

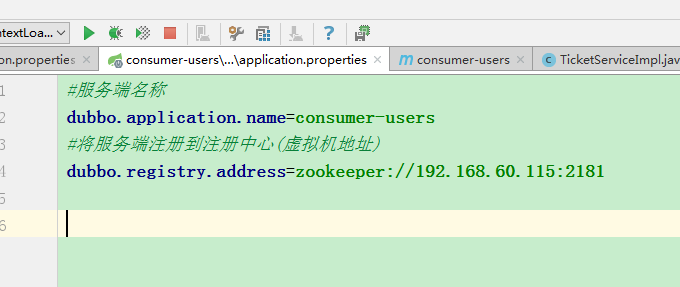
配置完pom文件之后，进行文件配置：

配置完成之后编写我们需要调用的接口，和正常开发一样，编写TicketService和TicketServiceImpl实现类。

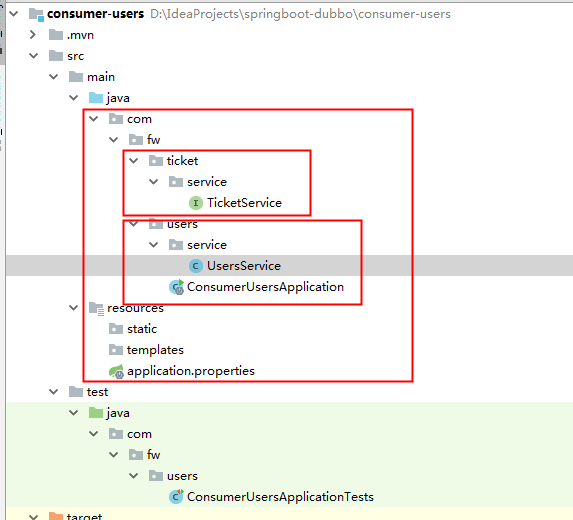


很简单的实现类使用service注解，这个注解是dubbo的注解：

到这一步票务系统暂时完成，下来编写用户系统。

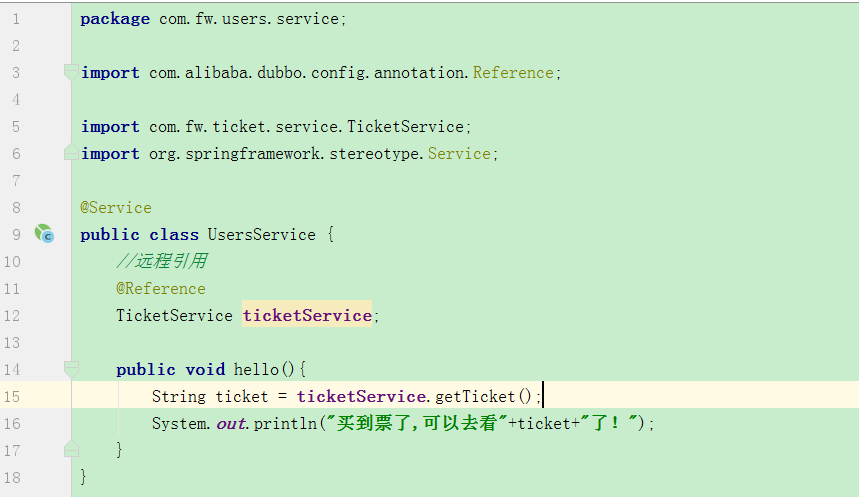


用户的服务配置



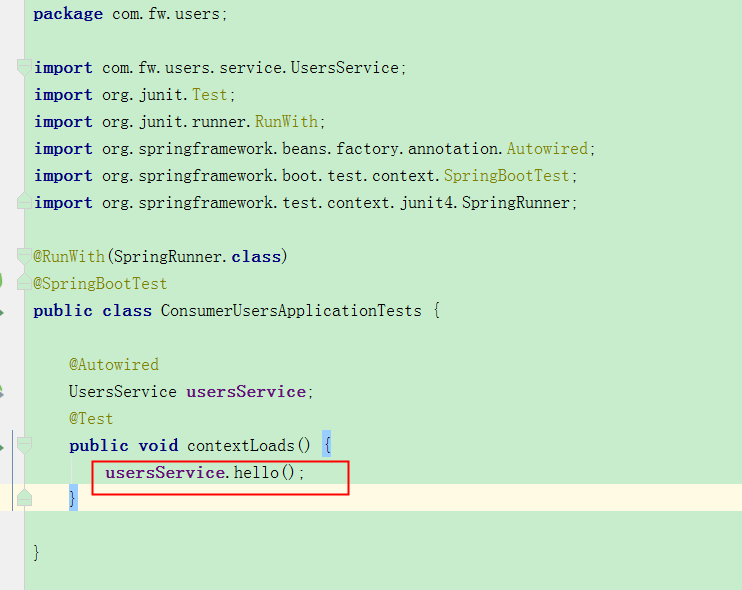
用户端的目录结构，因为用户端需要调用服务，所以将票务系统接口拷贝到用户项目下。

用户代码如下：



通过Reference注解，即可进行多项目远程调用。

编写测试用例：

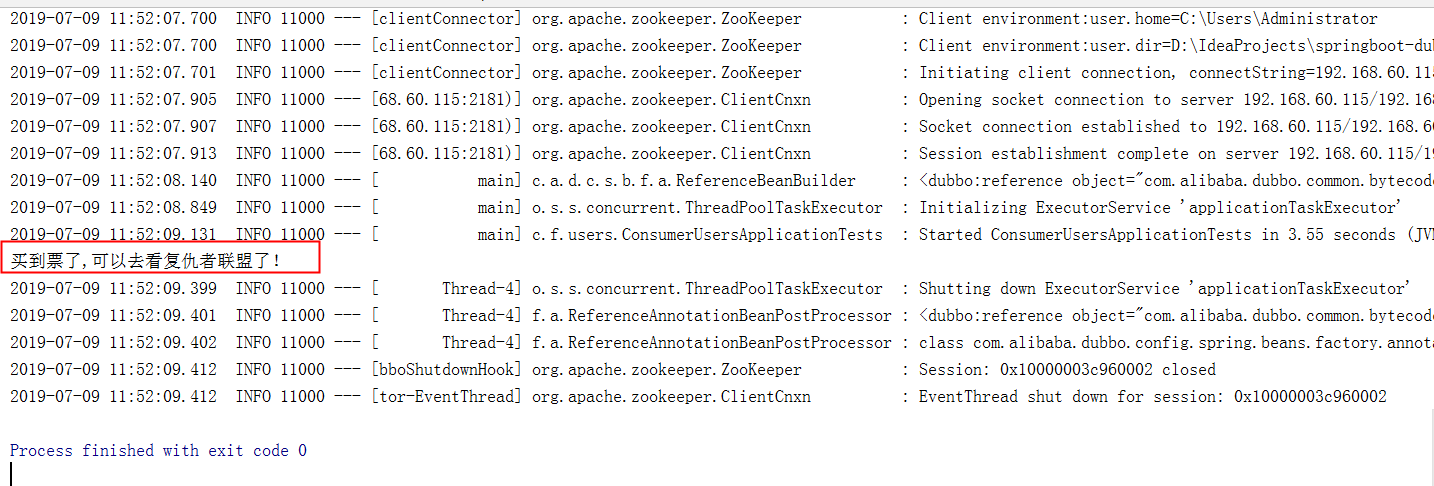


进行用户项目接口调用，并测试。

首先启动服务端的项目服务，然后在启动用户的测试用例：



经过漫长的启动等待，终于可以打压出我们要的结果了。如下图



至此，springboot+dubbo+Zookeeper多实例分布式项目搭建完成。