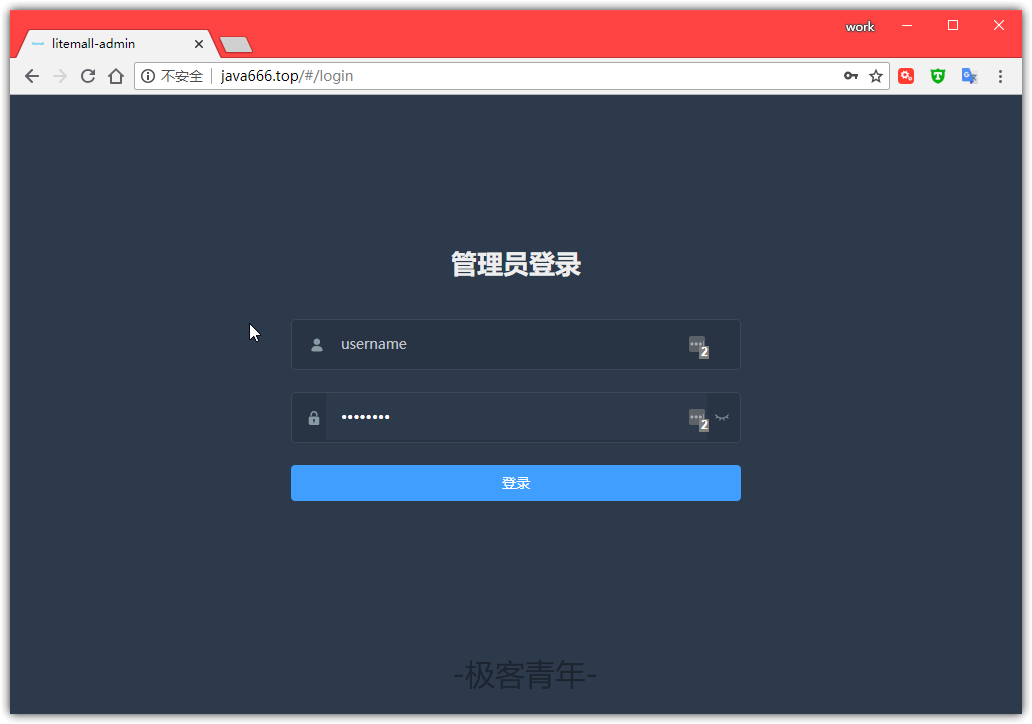
litemall 轻商城开发笔记之我在谷歌云centos 上面踩过的坑和奇葩问题

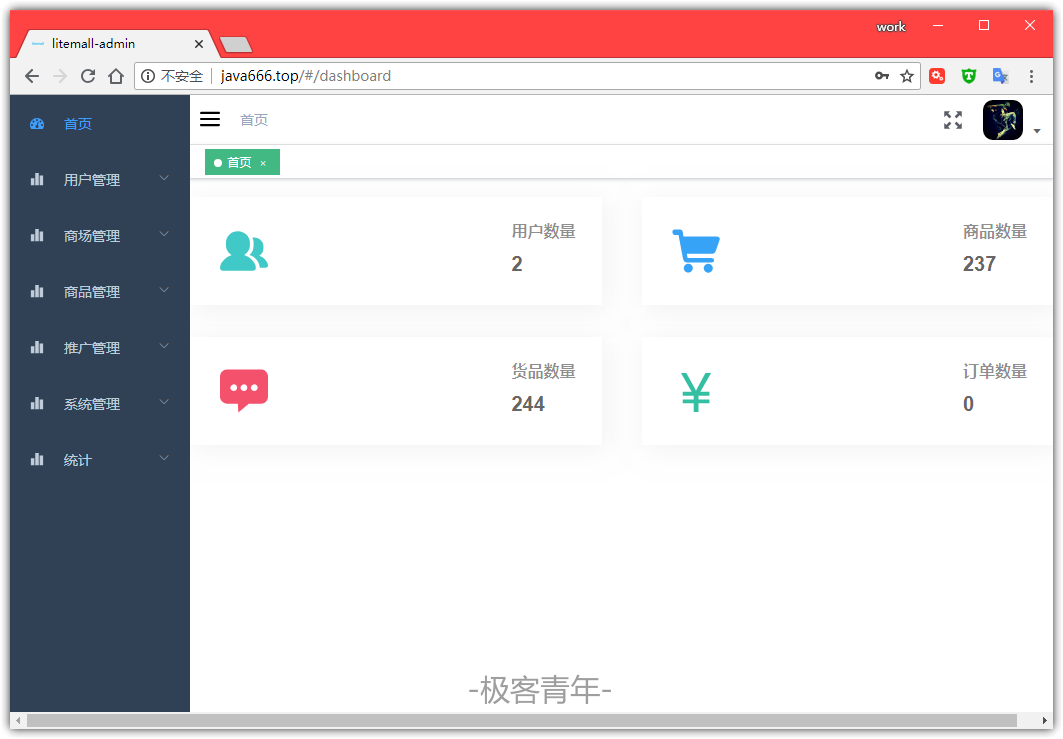
litemall 轻商城开发笔记之我在谷歌云centos 上面踩过的坑和奇葩问题

首先介绍下项目的背景，litemall 项目是码云代码托管平台开源软件页面推荐的项目，无意之间看到的，逛了一下他的演示站，我立刻被震撼到了，看过许多开源作品，只有litemall 的web后台管理系统让我眼前一亮，VUE 框架开发的 WEB 页面果然十分的优雅和美观，于是出于职业习惯和强烈的研究欲望，我马上决定把这个项目运行在我那闲置多年的谷歌云服务器上。先看下效果图：

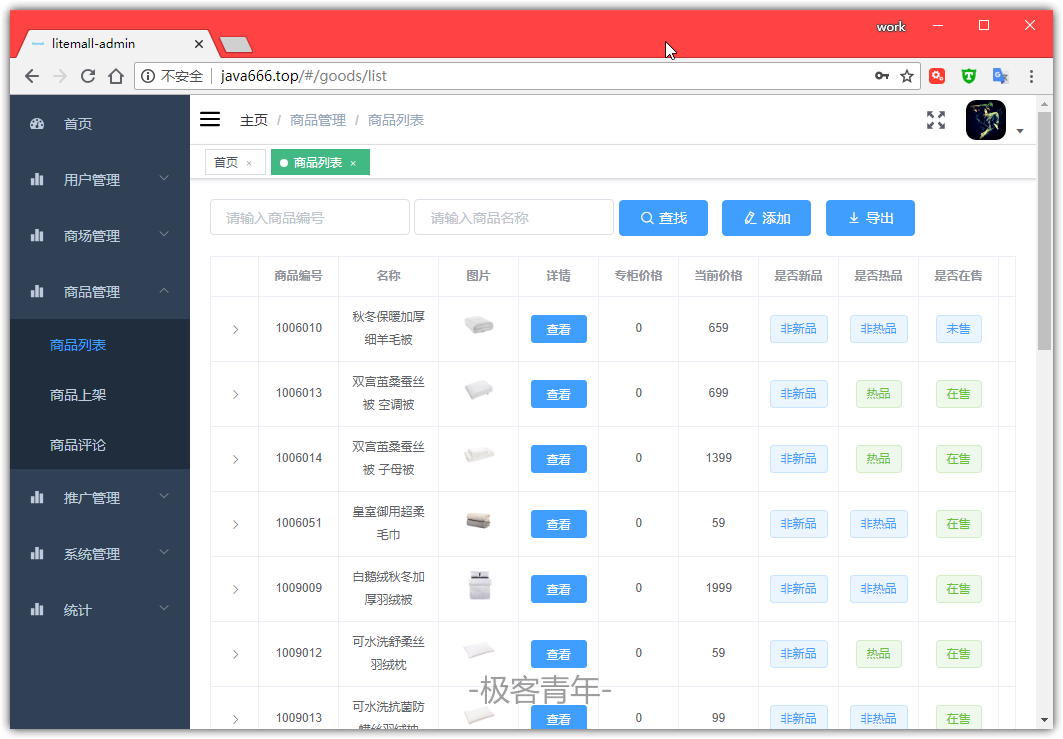
<http://p7fcrq2e4.bkt.clouddn.com/201818012328-chrome20180801_232819.png-sy>



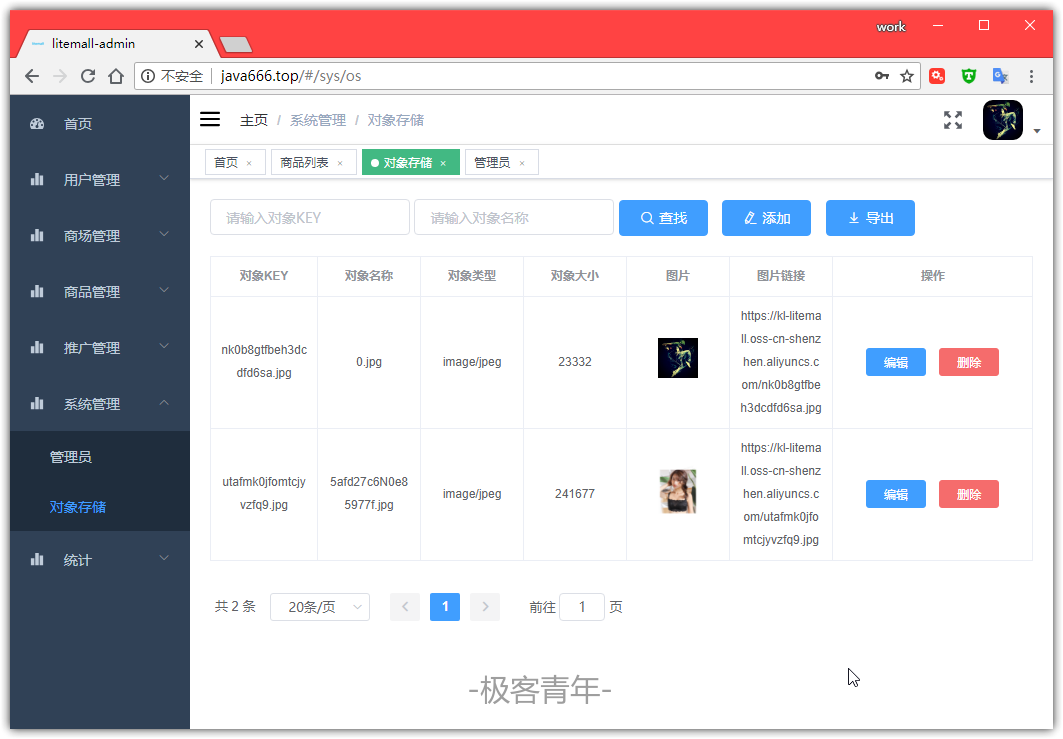
<http://p7fcrq2e4.bkt.clouddn.com/201818012329-chrome20180801_232945.png-sy>



<http://p7fcrq2e4.bkt.clouddn.com/201818012331-chrome20180801_233129.png-sy>



<http://p7fcrq2e4.bkt.clouddn.com/201818012332-chrome20180801_233246.png-sy>



1-关于一年免费试用谷歌云搭建科学上网服务器的文章我推荐这个，写得非常详细：<https://www.wmsoho.com/google-cloud-platform-ssr-bbr-tutorial/>

原版 & 魔改版 Google BBR 拥塞控制算法一键安装脚本：<https://www.vultrcn.com/5.html>

<http://p7fcrq2e4.bkt.clouddn.com/201818012308-chrome20180801_230818.png-sy>



2-有了服务器和科学上网环境之后，新建虚拟机和配置网络防火墙之类的此处略过一万字。本文以谷歌云centos 6为例，其他系统仅供参考。提一下谷歌云默认的云虚拟机无法root 远程登录，ssh 密钥登录方式见推荐文章。可以先通过网站的内置控制台进入云虚拟机更改以下代码，即可实现远程root登录：

2.1-以非root用户进入控制台后，

sudo su - root

进入root账户

passwd root

设置root账户的密码

然后修改SSH配置文件

vim /etc/ssh/sshd\_config

找到PermitRootLogin和PasswordAuthentication，参考如下设置：

PermitRootLogin yes //默认为no，需要开启root用户访问改为yes

PasswordAuthentication yes //默认为no，改为yes开启密码登陆

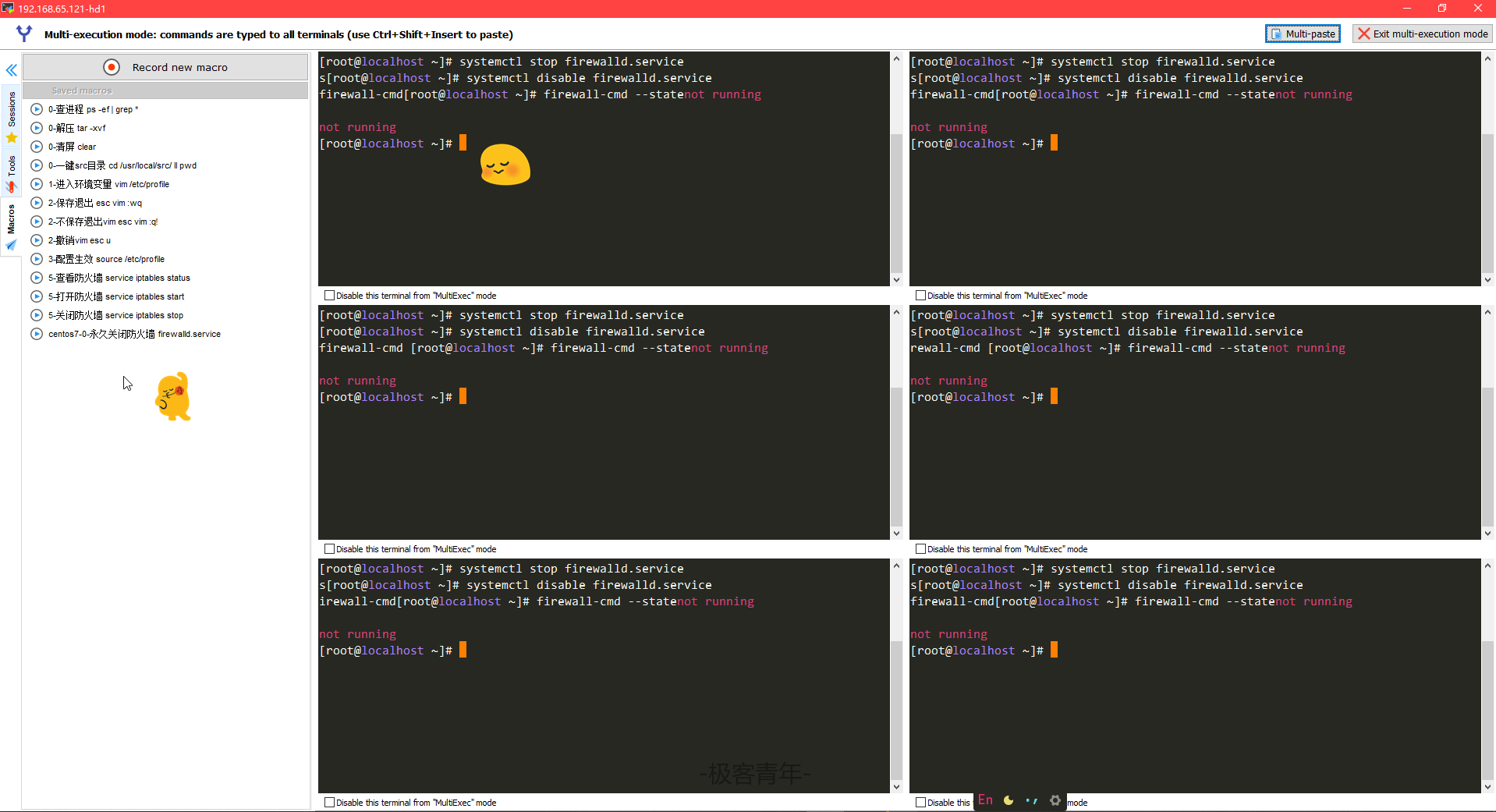
重启SSH服务使修改生效

/etc/init.d/ssh restart

3-接下来，快速部署LNM（Linux+Nginx+Mysql）环境。

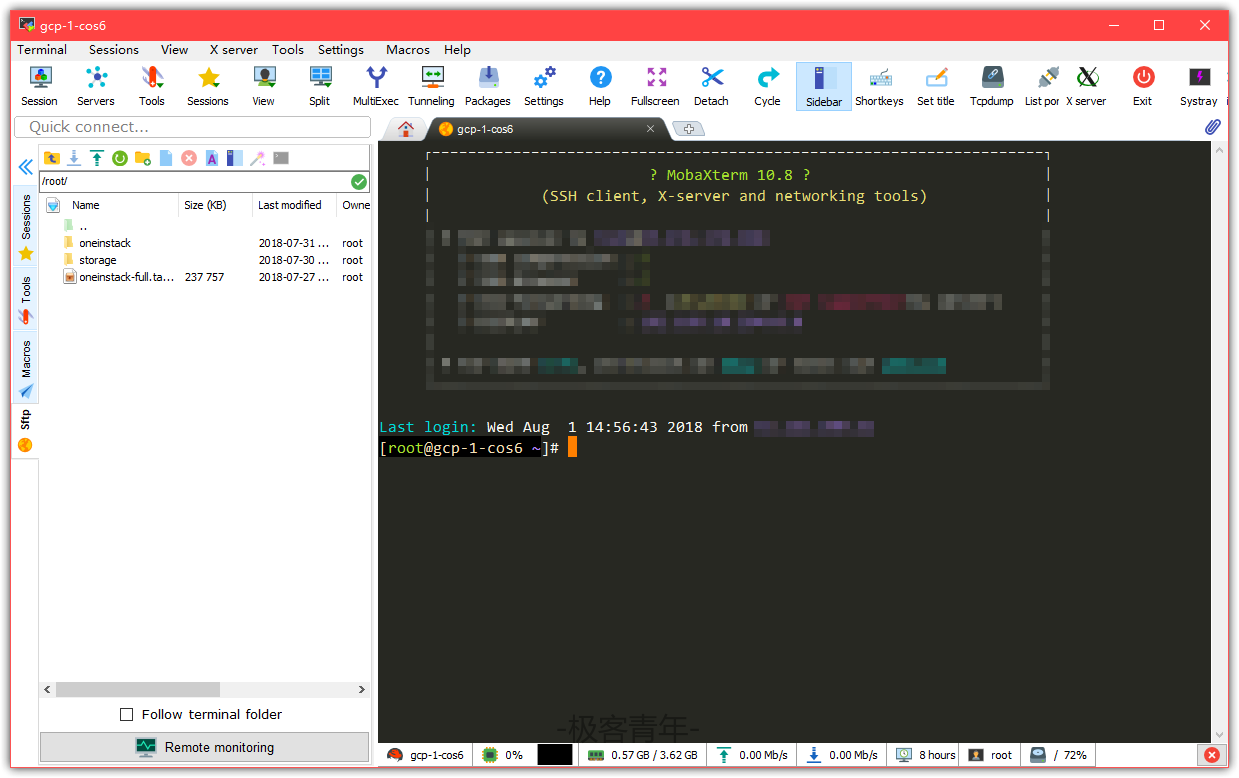
3.1-本文使用远程终端神器mobaxterm 来管理服务器，至于这个软件有多强大，看图：

<http://p7fcrq2e4.bkt.clouddn.com/201818020414-20180802_041416.png-sy>



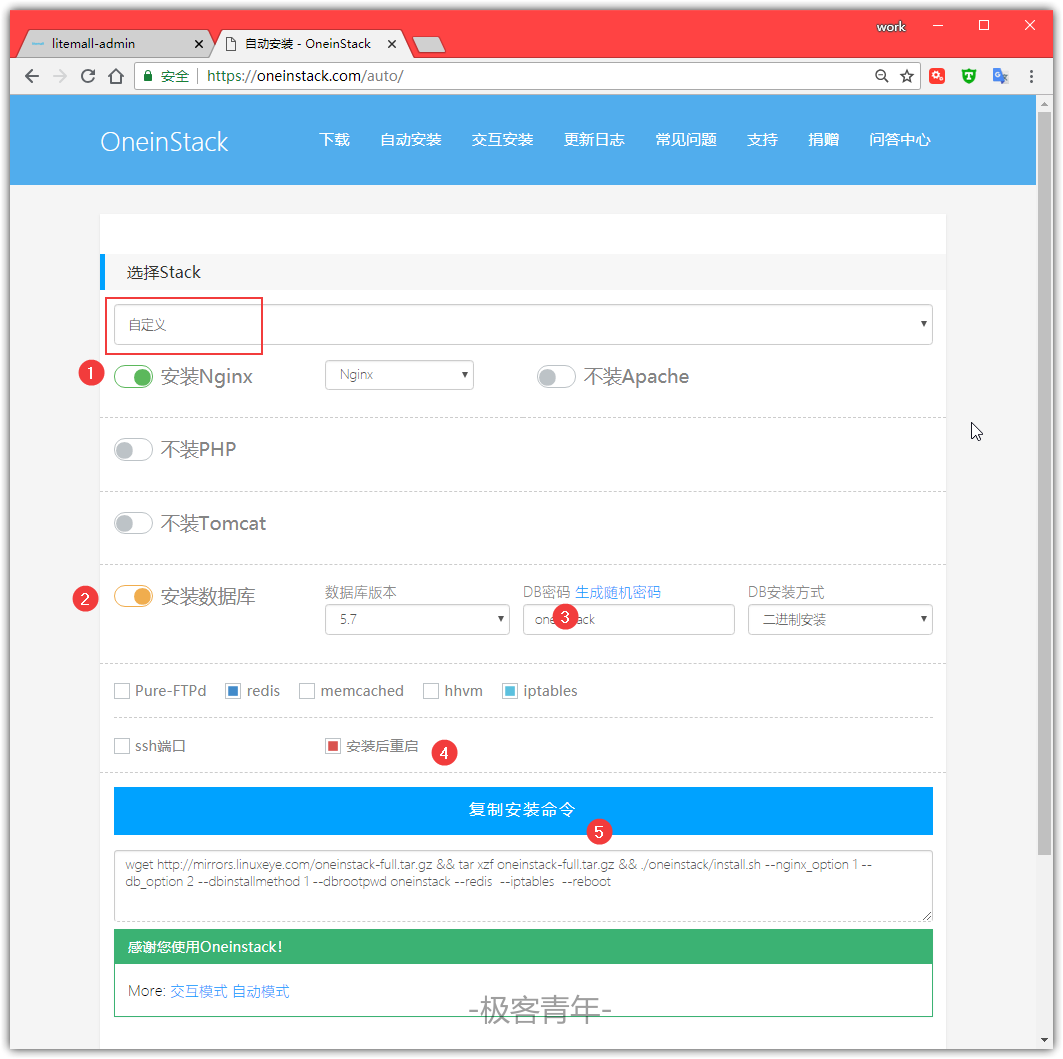
登入新建的云虚拟机：

<http://p7fcrq2e4.bkt.clouddn.com/201818012335-MobaXterm20180801_233533.png-sy>



3.2-打开oneinstack网站：<https://oneinstack.com/auto/>，使用一键部署命令，只需要20分钟左右就可以快速完成LNM环境部署。

<http://p7fcrq2e4.bkt.clouddn.com/201818012342-chrome20180801_234205.png-sy>



我连代码都给你准备好了（一行），redis 可选，你自己决定，如果不装tomcat ，你需要自己安装jdk 1.8：

wget http://mirrors.linuxeye.com/oneinstack-full.tar.gz && tar xzf oneinstack-full.tar.gz && ./oneinstack/install.sh --nginx\_option 1 --db\_option 2 --dbinstallmethod 1 --dbrootpwd oneinstack --redis --iptables --reboot

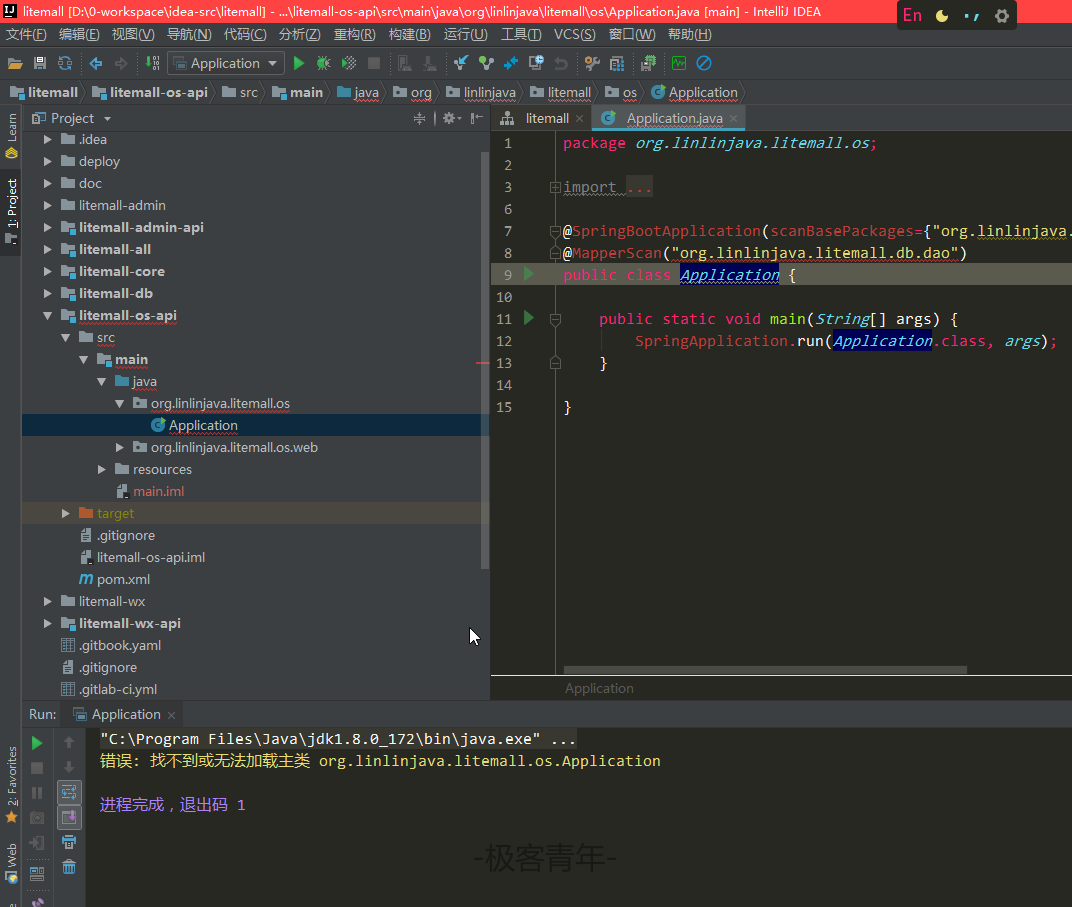
重启之后，mysql 和 nginx 已经默认启动，此时访问你的云端虚拟主机ip 应该可以访问到nginx 页面，注意1：如果无法访问80端口，请依次检查---云主机防火墙---云平台防火墙---端口占用----实在不行重启。

4-准备项目代码，修改配置文件并打包。

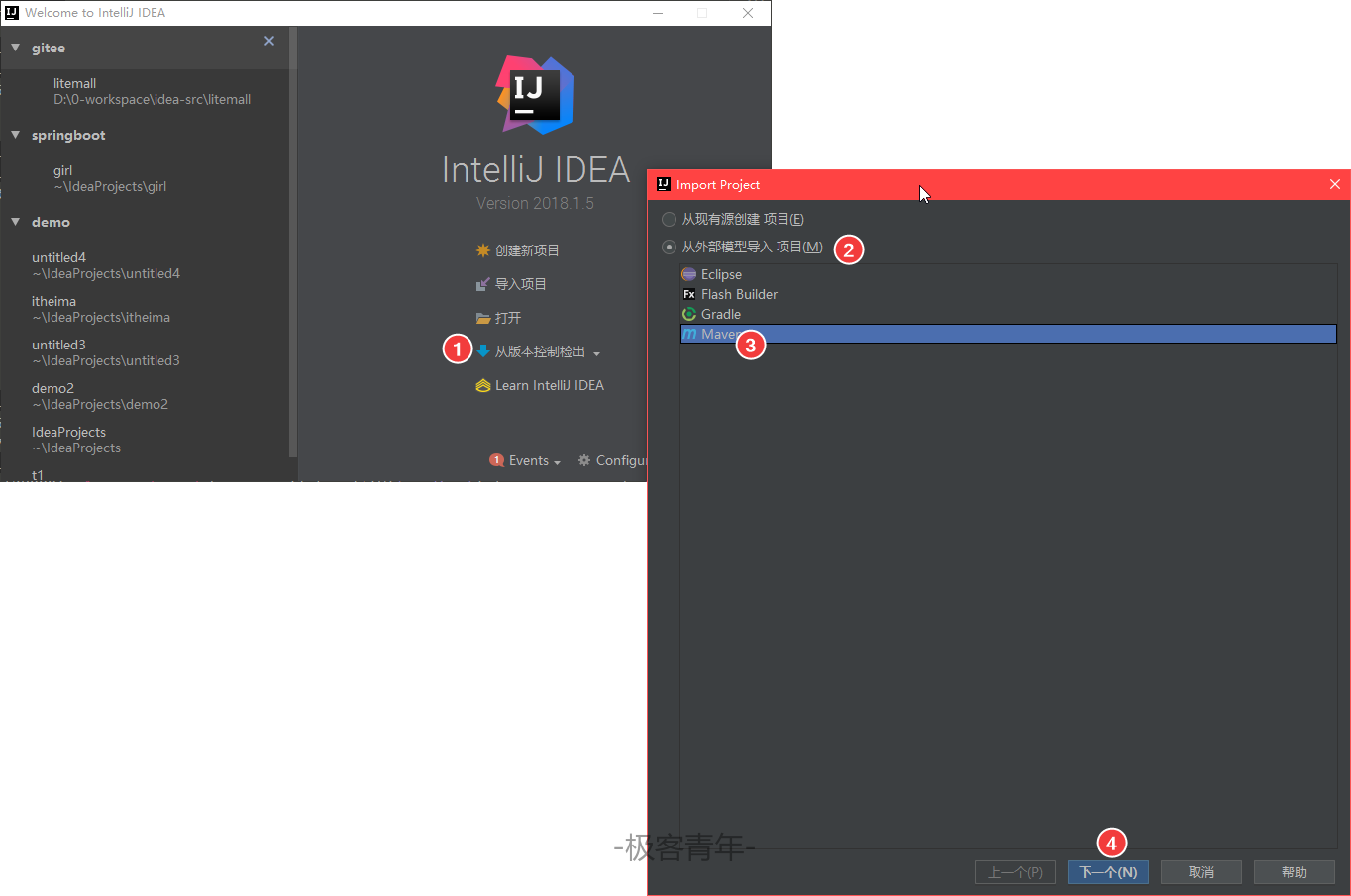
4.2-下载litemall 开源项目的代码到本地，码云：<https://gitee.com/jkqn/litemall/blob/v0.8.0>

推荐下载标签版本，如果git 克隆到本地则是下载默认主分支，至于idea 切换到标签版本我不太熟。也可以使用IDEA 编辑器git 插件克隆，注意2：克隆后导入方式选择maven，否则项目无法编译。

<http://p7fcrq2e4.bkt.clouddn.com/201818020419-20180802_041947.png-sy>

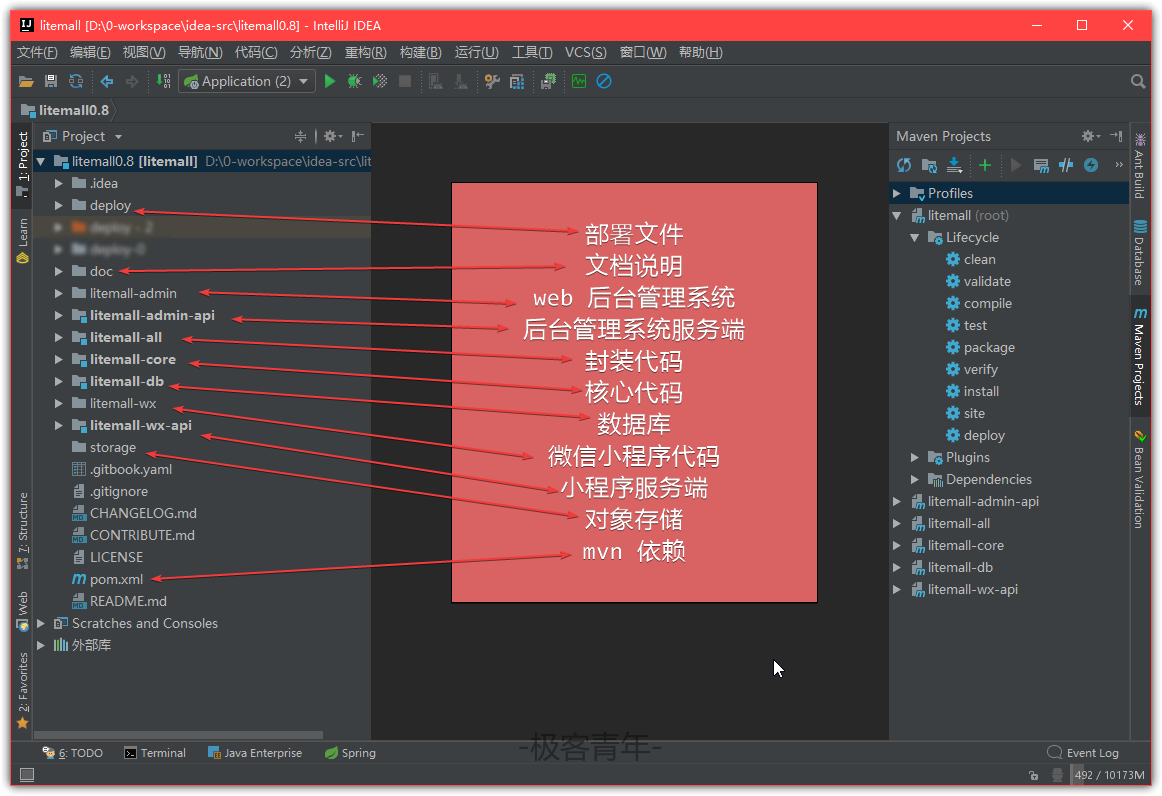


<http://p7fcrq2e4.bkt.clouddn.com/201818020058-20180802_005800.png-sy>

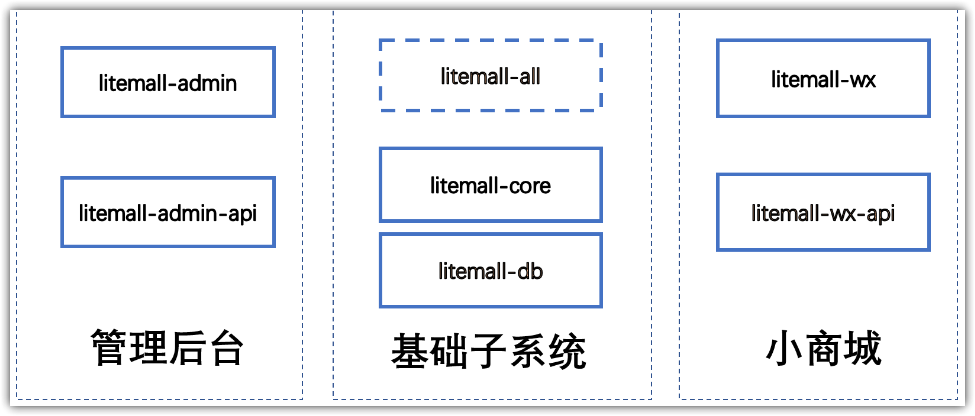


导入后效果如下：

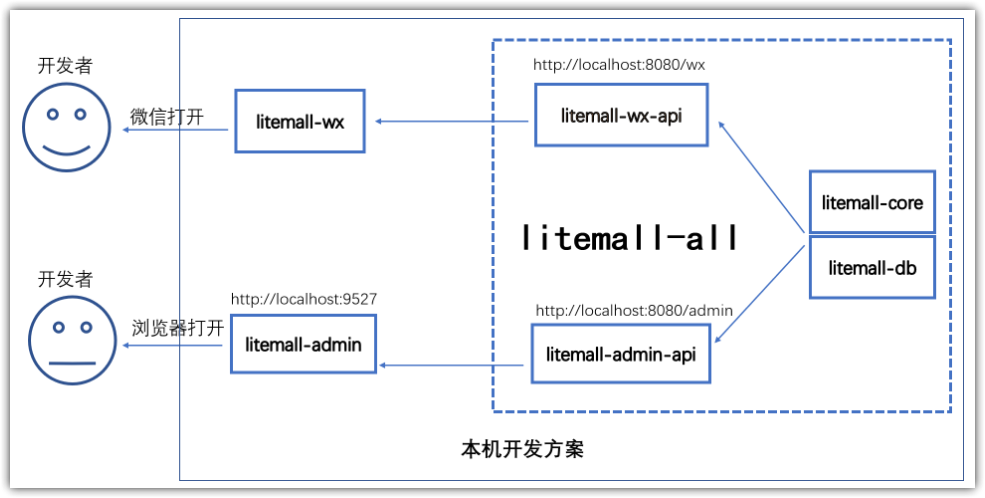
<http://p7fcrq2e4.bkt.clouddn.com/201818020006-idea6420180802_000623.png-sy>



<http://p7fcrq2e4.bkt.clouddn.com/201818020015-chrome20180802_001511.png>



<http://p7fcrq2e4.bkt.clouddn.com/201818020016-chrome20180802_001617.png>



当前开发阶段的方案：

MySQL数据访问地址jdbc:mysql://localhost:3306/litemall

litemall-wx-api后台服务地址http://localhost:8080/wx，数据则来自MySQL

litemall-admin-api后台服务地址http://localhost:8080/admin,数据则来自MySQL

litemall-admin前端访问地址http://localhost:9527, 数据来自litemall-admin-api

litemall-wx没有前端访问地址，而是直接在微信小程序工具上编译测试开发，最终会部署到微信官方平台（即不需要自己部署web服务器），而数据则来自litemall-wx-api。

部署阶段可参考以上配置修改相应地址。本地开发测试过程如下：

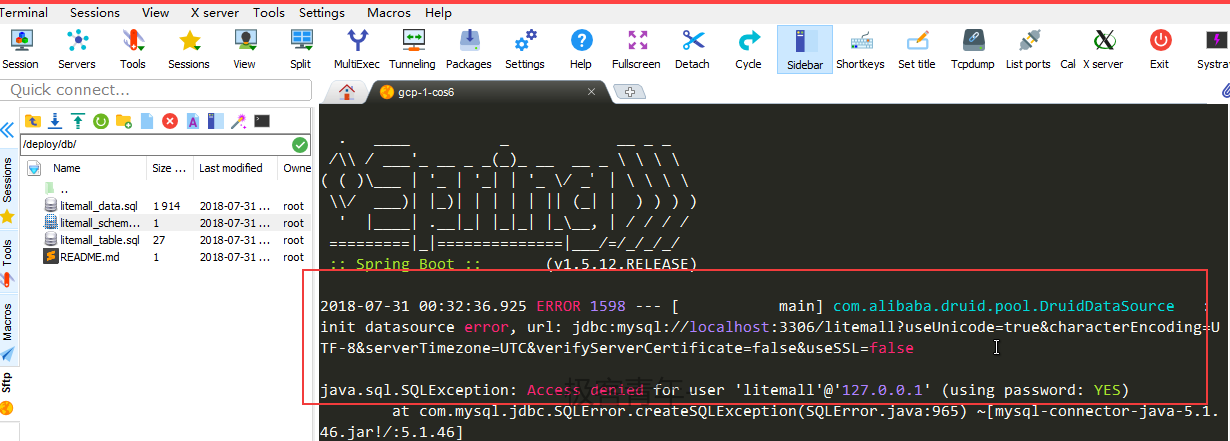
4.2-数据库环境设置过程如下：

本机安装MySQL，创建数据库、用户权限、数据库表和测试数据。

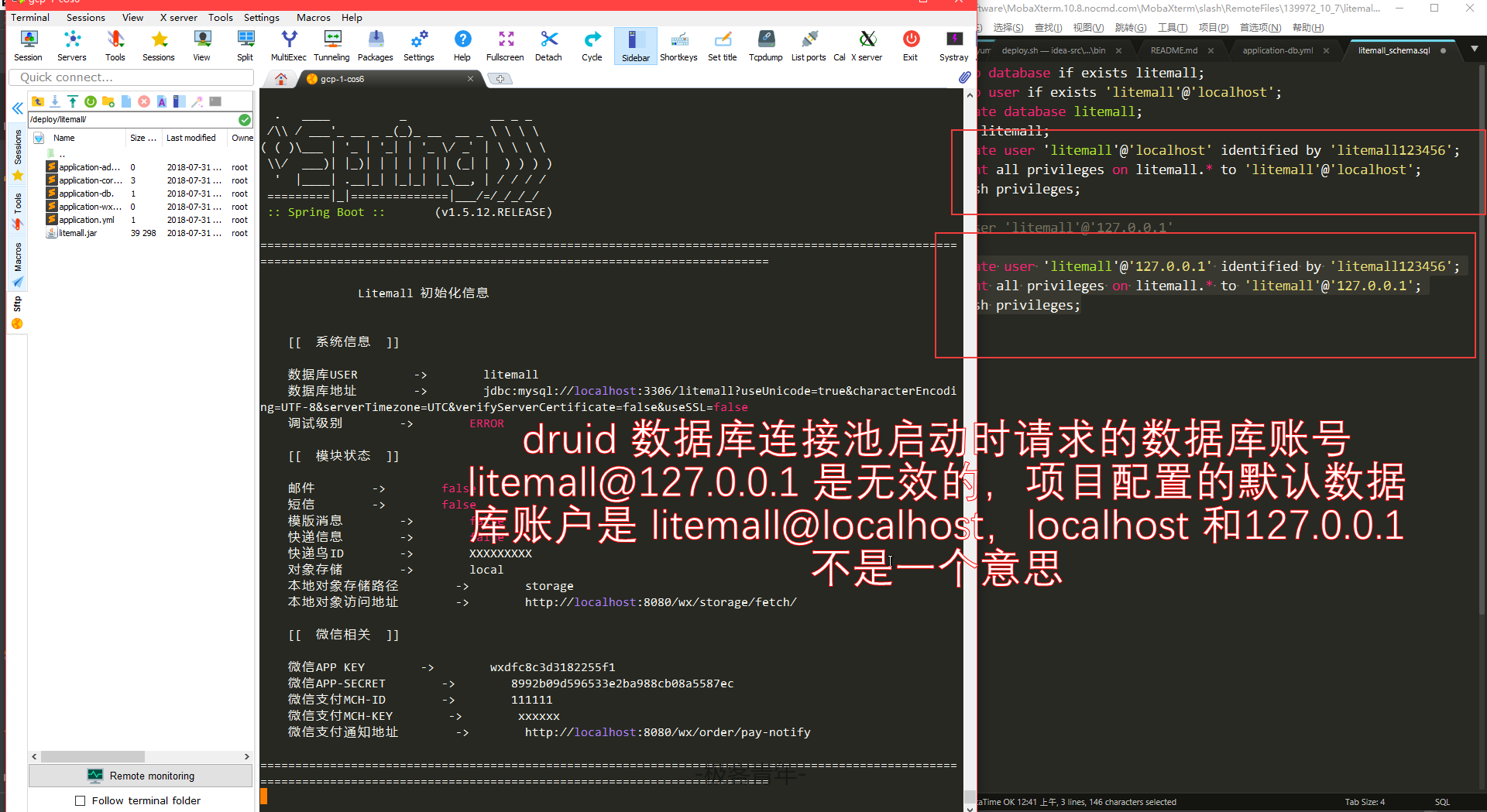
数据库文件在litemall-db/sql文件夹中，其中litemall\_schema.sql创建数据库和用户权限， litemall\_table.sql则创建表，litemall\_data.sql则是测试数据。

注意3：如果数据库连接异常，本地localhost 和 127.0.0.1 不相同，必须一一对应。

<http://p7fcrq2e4.bkt.clouddn.com/201818020425-20180802_042514.png-sy>



<http://p7fcrq2e4.bkt.clouddn.com/201818020425-20180802_042551.png-sy>



4.3-接下来先IDEA 双击执行maven 菜单下的litemall根节点的clean，再双击install 安装依赖库。

4.4-IDEA编辑器在litemall-all模块的Application类 右键Run Application.main()方式运行该模块，运行时会自动编译再运行。启动成功后可以看到，litemall-wx-api等模块多了target文件夹，里面是编译出的文件。打开浏览器，输入

http://localhost:8080/wx/index/index

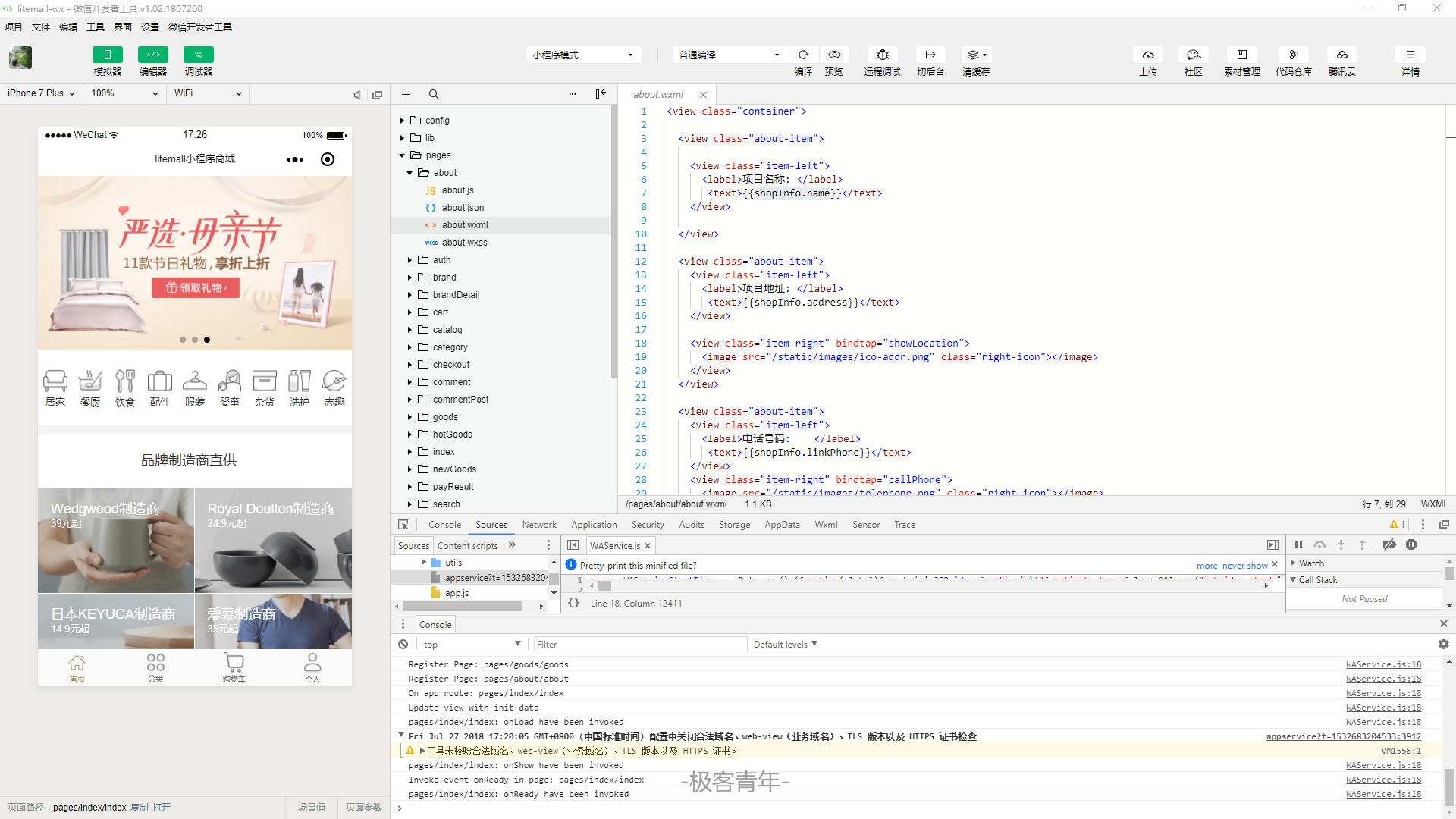
http://localhost:8080/admin/index/index

如果出现JSON数据，则litemall-all模块运行正常。

4.5-微信小程序开发测试：

安装微信小程序开发工具，导入本项目的litemall-wx模块文件夹，编译前，请确定litemall-all模块已经运行，litemall-wx模块的config文件夹中的api.js已经设置正确的后台数据服务地址；点击编译，如果出现数据和图片，则运行正常。详情查看项目文档。

<http://p7fcrq2e4.bkt.clouddn.com/201818020430-20180802_043013.png-sy>



4.6-Vue开发环境部署：

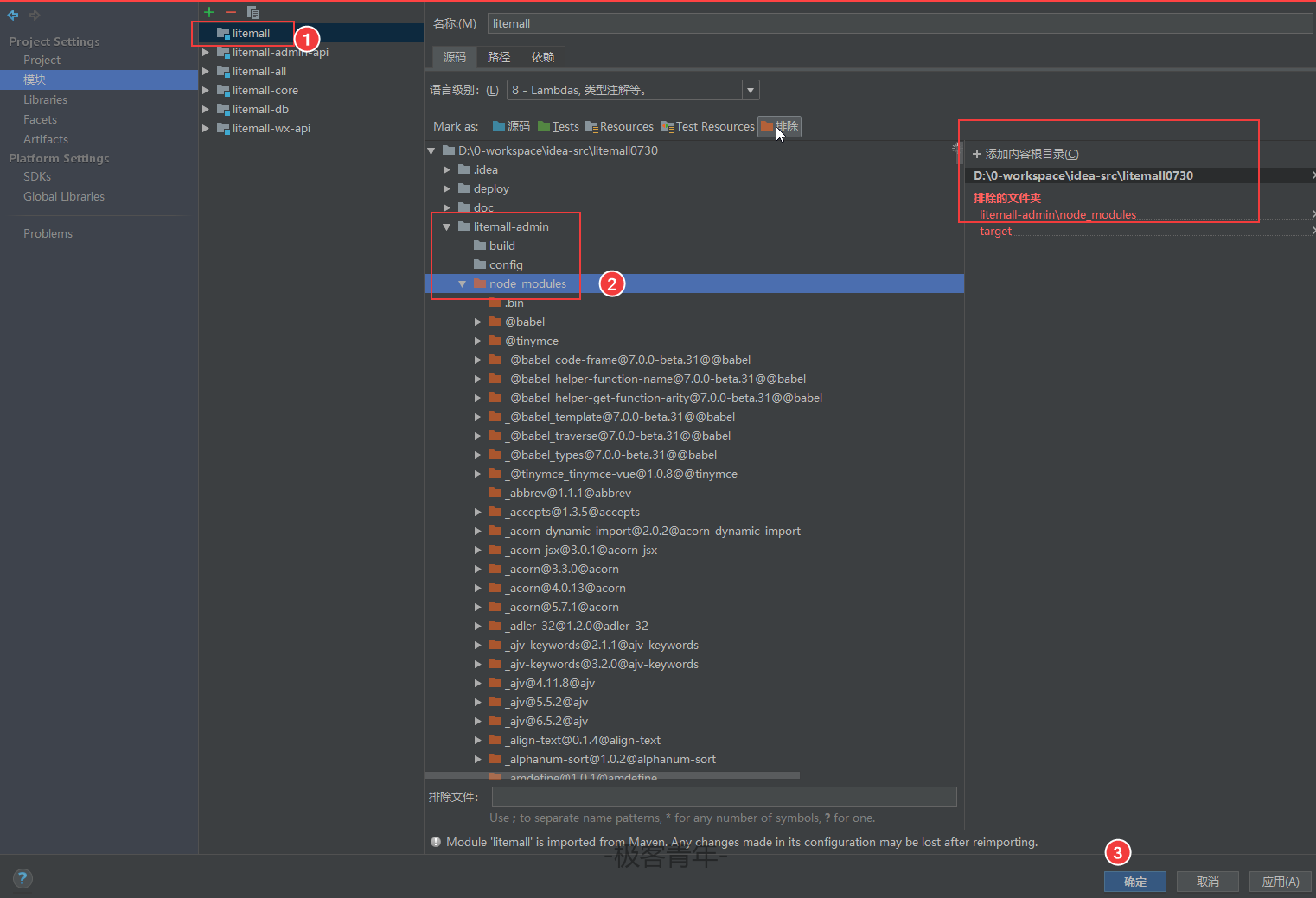
安装nodejs，安装依赖库：

cd litemall/litemall-admin

npm install -g cnpm --registry=https://registry.npm.taobao.org

注意4：为了防止cnpm 产生的文件太多导致IDEA 索引卡死，强烈建议先在litemall-admin 下面手动新建node\_modules 文件夹，并将其排除，避免idea 索引这个目录导致卡死。

<http://p7fcrq2e4.bkt.clouddn.com/201818020431-20180802_043126.png-sy>



接下来：

cd litemall/litemall-admin

cnpm install

编译并运行web 项目

cnpm run dev

然后，打开浏览器，输入http://localhost:9527。 如果出现管理后台登录页面，则表明管理后台的前端运行正常；

请确定litemall-all模块已经运行，然后点击登录，如果能够成功登录，则表明管理后台的前端和后端对接成功，运行正常。

5-项目打包，整理部署文件并上传云主机。

5.1-在主机打包项目到deploy目录：

5.1.1-复制sql 文件到deploy/db目录下：itemall\_schema.sql、litemall\_table.sql、litemall\_data.sql；

5.1.2-编译打包部署环境下的web 文件dist，并将litemall-admin/dist目录复制到deploy下面：

cd ./litemall-admin

cnpm install #已执行过可省略

cnpm run build:dep

5.1.3-执行mvn 菜单的litemall根节点clean 和package，打包，然后把生成的可执行jar包litemall-all/target/litemall-all-\*-exec.jar 复制到/deploy/litemall/litemall.jar；

5.1.4-修改deploy/litemall文件夹下面的\*.yml外部配置文件，当litemall-all模块启动时会 加载外部配置文件，而覆盖默认jar包内部的配置文件。 例如，配置文件中一些地方需要设置成远程主机的IP地址。

注意5：要使用绝对地址，而不要使用相对地址比如localhost。

5.1.5-deploy文件夹结构包含dist网页静态文件，db数据库文件，litemall-all 主程序文件，然后将deploy 上传到云主机根目录，推荐使用mobaxterm直接拖拽；

6-项目部署：修改云主机配置文件并启动项目，本文以centos 6为例。

6.1-确保远程主机环境MySQL和JDK1.8已经安装好，确保云主机平台的全局防火墙和云主机防火墙 iptables 已经允许相应的端口3306 80 8080 等。安装过程参考3.2。

6.2-依次按顺序导入db/litemall\_schema.sql litemall\_table.sql litemall\_data.sql，顺序不可打乱。

6.3-启动springboot服务：

chmod a+x /deploy/litemall/litemall.jar #增加全部用户可执行权限

sudo ln -f -s /deploy/litemall/litemall.jar /etc/init.d/litemall #创建软连接注册服务

sudo service litemall start

注意6：如果提示unable to find java ，而且java -version 显示正确的话，可能的原因是jdk 安装方式不规范，需要建立java命令的软链接到/sbin。

如：ln -s /usr/local/jdk/bin/java /sbin/java

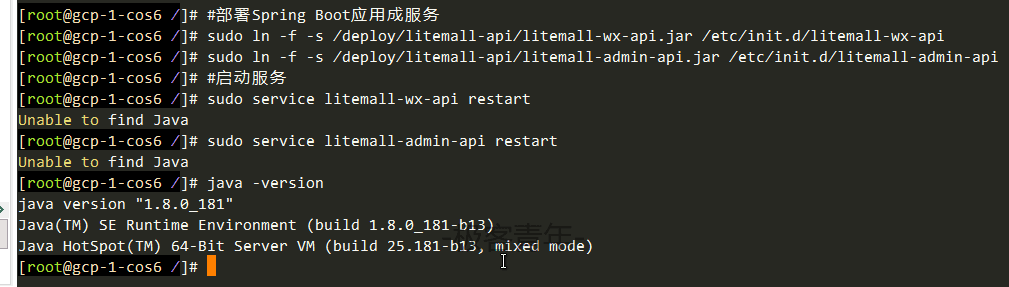
<http://p7fcrq2e4.bkt.clouddn.com/201818020432-20180802_043224.png-sy>



参考来源：<https://www.jianshu.com/p/563497a6e1a7>

注意：7如果还是提示failed to start ，

<http://p7fcrq2e4.bkt.clouddn.com/201818020432-20180802_043255.png-sy>



那就该祭出终极大杀器了，这个问题我找遍了google 才在一个国外网站找到答案：

<http://p7fcrq2e4.bkt.clouddn.com/201818020433-20180802_043322.png-sy>



下面先说一下我是怎么解决这个问题的，因为尝试了所有的办法去启动服务失败了，我就去查看了litemall 服务启动日志/var/logs/litemall.log，我在文件中发现了如下字样：

"start-stop-daemon: unrecognized option '--no-close'"我用google 搜了一下，找到stackoverflow 社区的一个帖子，跟我的案例很相似。案后翻译中文，按照提示执行最关键的一句：

Create a config file /var/appname/appname.conf with the following content

USE\_START\_STOP\_DAEMON=false

果然，程序正确启动了，我激动得流下了感动的眼泪。现在，我们的项目核心jar 包已经运行，由于已经集成tomcat，所以无需再部署tomcat，浏览器访问如下地址，测试能否正常显示。

http://xxx.xxx.xxx.xxx:8080/wx/index/index

http://xxx.xxx.xxx.xxx:8080/admin/index/index

http://xxx.xxx.xxx.xxx:8080/#/login

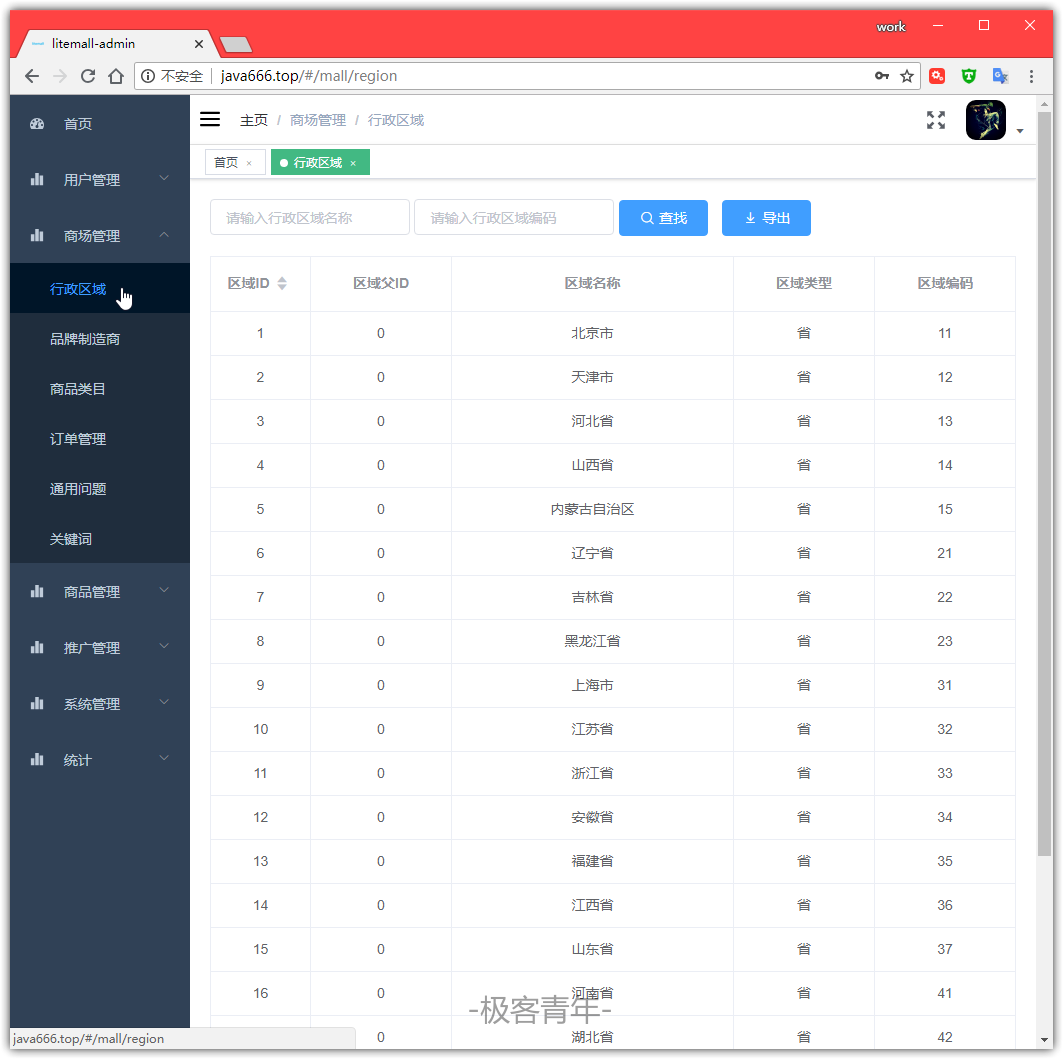
7-修改nginx 并配置域名，ssl 证书可选。

在这个项目的nginx 配置过程，我遭遇了无尽的挫折，明明平时用的很正常的nginx ，偏偏这次无法访问80 端口，后来走投无路的我干脆重启云主机【注意8】，万万没想到，80端口竟然开放了。

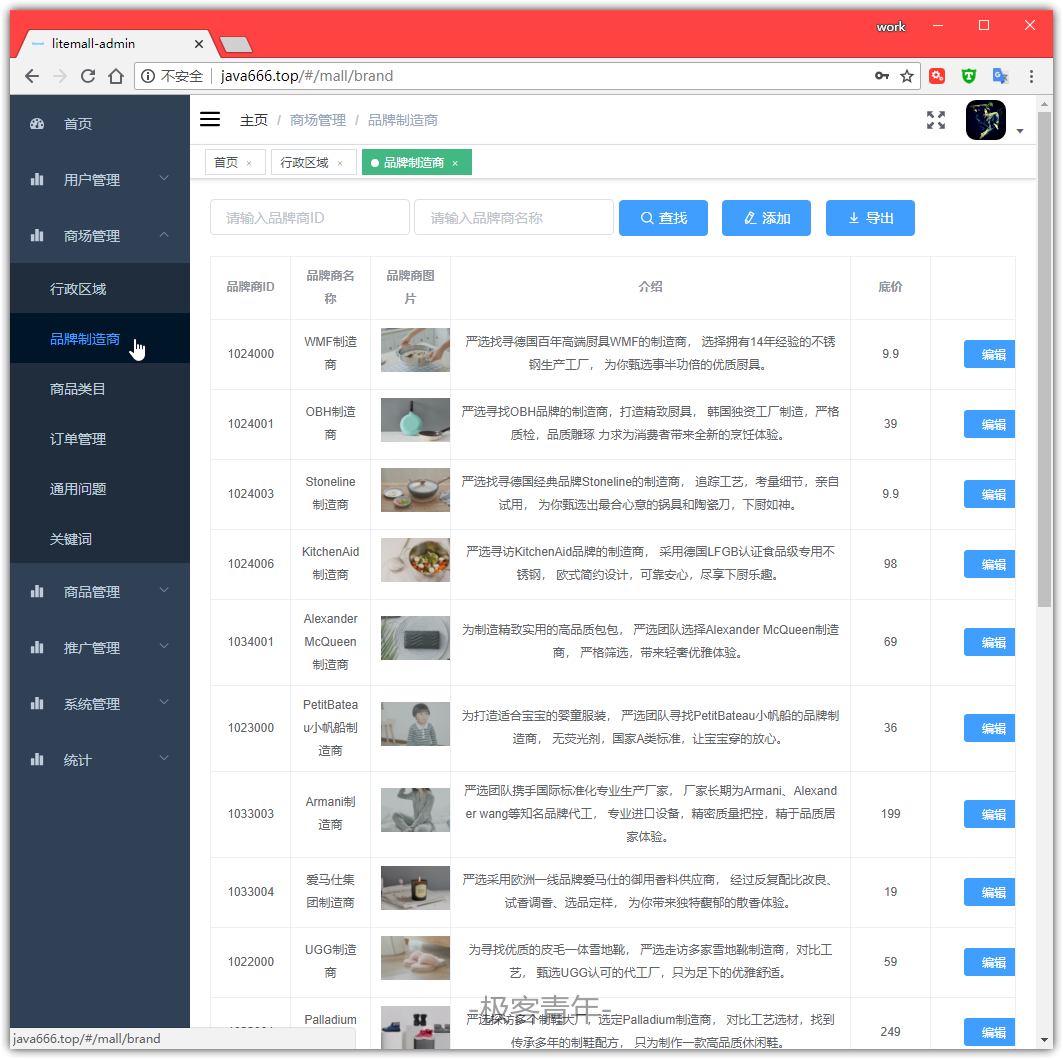
然后修改nginx.conf文件反向代理相应的服务，终于可以通过<java666.top>来访问项目演示啦，此处省略500字。至于ssl 证书配置，请参考文档，反正我配置没成功，哭瞎。

8-测试效果：

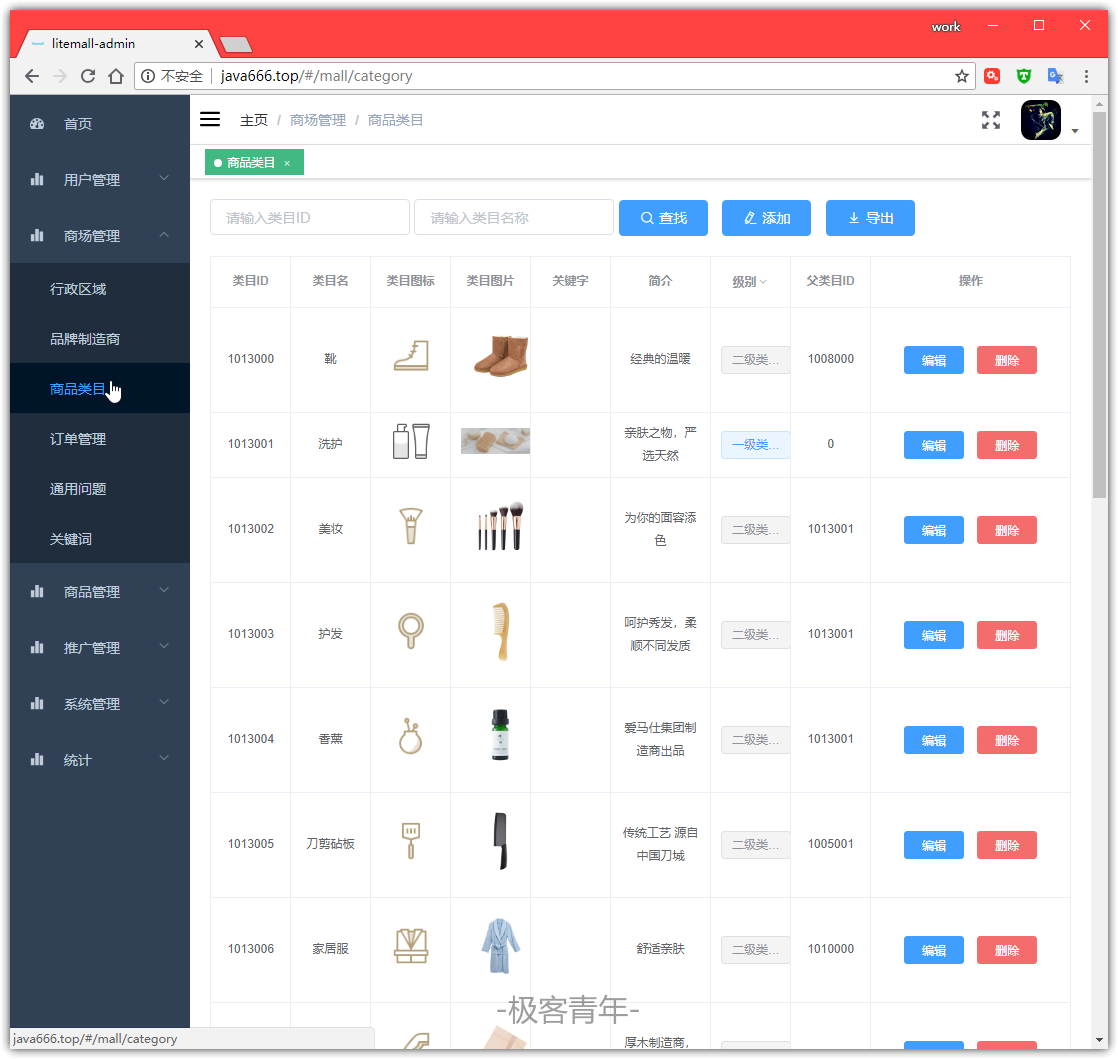
<http://p7fcrq2e4.bkt.clouddn.com/201818020355-chrome20180802_035527.png-sy>



<http://p7fcrq2e4.bkt.clouddn.com/201818020356-chrome20180802_035639.png-sy>



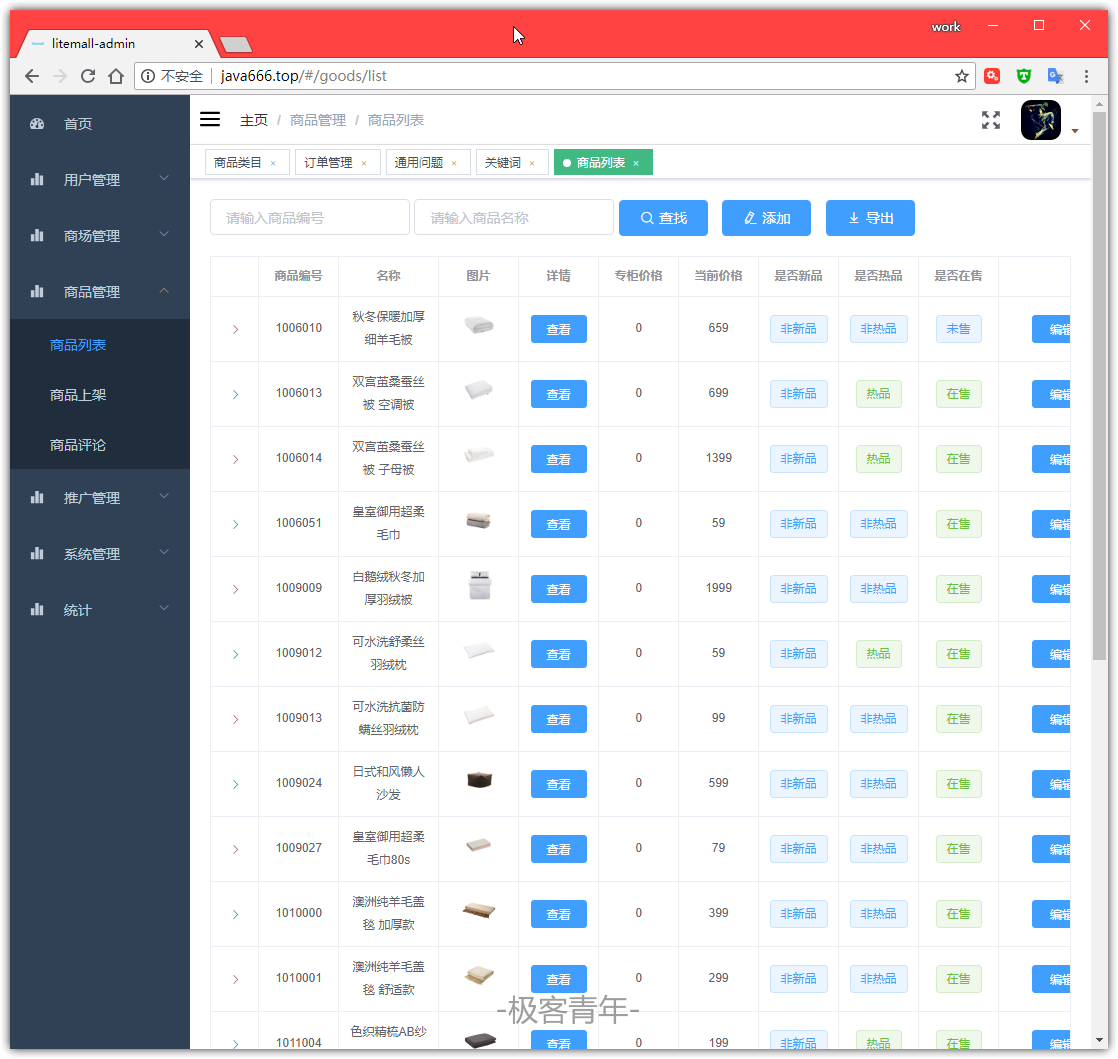
<http://p7fcrq2e4.bkt.clouddn.com/201818020357-chrome20180802_035730.png-sy>



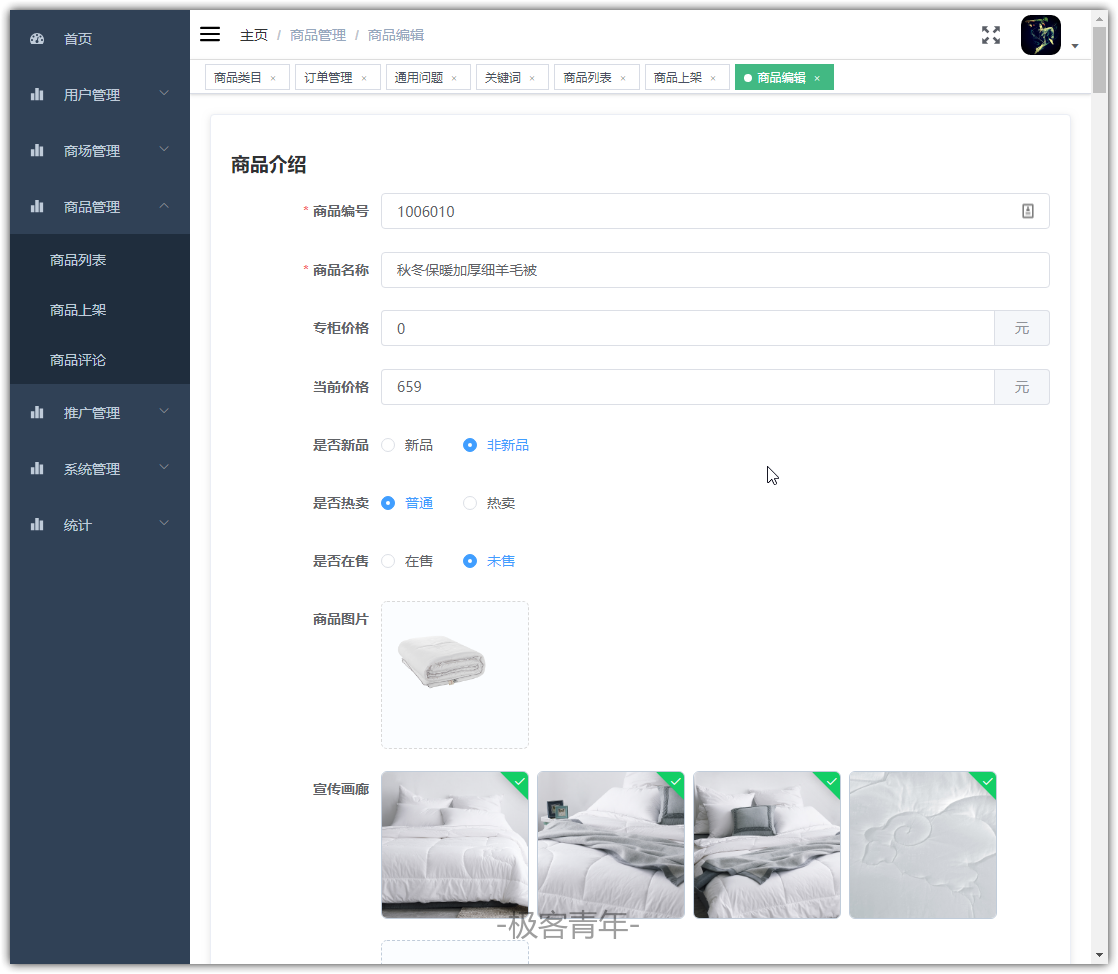
<http://p7fcrq2e4.bkt.clouddn.com/201818020358-chrome20180802_035809.png-sy>



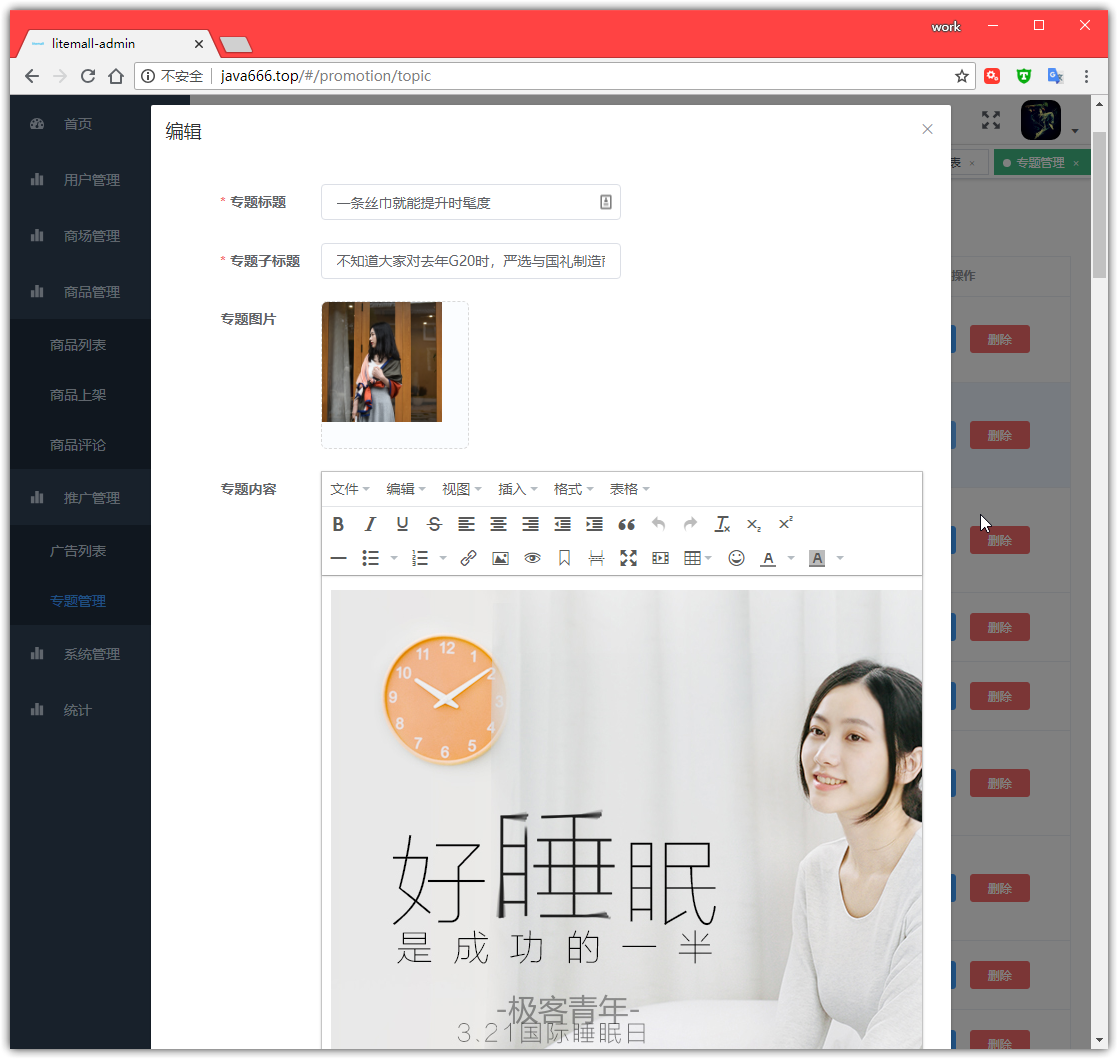
<http://p7fcrq2e4.bkt.clouddn.com/201818020359-chrome20180802_035909.png-sy>



<http://p7fcrq2e4.bkt.clouddn.com/201818020359-chrome20180802_035943.png-sy>



<http://p7fcrq2e4.bkt.clouddn.com/201818020400-chrome20180802_040018.png-sy>



想要快速查找本文的注意事项，请使用ctrl + f 搜索关键字“注意”。。。。