# 线上云服务器前后端分离部署方案

# 部署springboot+vue项目文档（若依ruoyi项目部署步骤）

<https://blog.csdn.net/Dreamboy_w/article/details/104389797>

## 总纲部署

1. .jdk1.8.

<https://www.cnblogs.com/stulzq/p/9286878.html>

1. Redis

安装Redis数据源

wget http://download.redis.io/releases/redis-5.0.4.tar.gz

#### 编译

make && make install

5.将redis-5.0.4目录下的redis.conf拷贝一份到/etc中 cp redis.conf /etc

6、修改etc目录下的redis.conf文件，将文件中的daemonize 将no改为yes（设置成守护进程），注释掉bind127.0.0.1添加bind 0.0.0.0（为后面外网连接做准备）。

这里我遇到个问题，一般修改文件，用vim命令，但我用了之后，redis.conf文件内容展示如下，里面的内容没有展示完全，试了很多方法不行，这个问题有待解决，如果有知道问题根源的同学可留言帮助我。我用了其它的方法修改，用xftp文件将redis.conf文件拷贝到本地桌面上，用notepad++打开进行修改并保存，随后将修改好的redis.conf文件重新上传到/etc目录下。

本地测试是否设置成功：

1、首先进入redis的src目录下 执行./redis-server /etc/redis.conf 开启服务端 （注意后面要加配置文件的路径）

2、再运行客户端 ./redis-cli，有如下所示说明开启成功

远程测试是否设置成功

1、首先在阿里服务器中添加redis安全组

2、配置防火墙 6379端口开放 firewall-cmd --permanent --add-port=6379/tcp

3、进入redis的src目录下 执行./redis-server /etc/redis.conf 开启服务端

3.启动redis 启动redis

1 service redis start

2 或

3 redis-server

停止redis

service redis stop

或

redis-cli shutdown

查看redis版本

1 redis-cli --version

查看redis运行状态

service redis status

查看redis进程

**ps** -ef | **grep** redis

4.设置密码

vi /etc/redis.conf

查找 requirepass foobared 将 foobared 修改为你的密码

5.开放远程登录连接

vi /etc/redis.conf

将bind 127.0.0.1 改成了 bind 0.0.0.0

6.设置为开机自动启动

chkconfig redis **on**

或者

systemctl enable redis.service

1. Mysql. Mysql环境需要
2. <https://blog.csdn.net/qq_40181063/article/details/90713153?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-3&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-3>
3. Maven(我感觉是不要)
4. Node(我感觉也不需要)
5. Nginx(需要）

1、查找是否安装nginx

find -name nginx

2、如果有则删除nginx

yum -y remove nginx

3、下载nginx

wget http://nginx.org/download/nginx-1.13.8.tar.gz

4、解压缩

tar -zxvf nginx-1.13.8.tar.gz

**cd**  nginx-1.13.8

./configure

make

make install

如果不报错就等于安装好了默认再/usr/local/nginx目录下

5、如果报错依赖 pcre模块

yum install pcre pcre-devel -y

6、如果还依赖zlib则继续安装

yum install -y zlib-devel

然后继续 ./configure make make install 这三步

可以通过： whereis nginx  nginx查看安装目录

7、启动nginx

**cd** /usr/local/nginx/sbin

./nginx

8、查看nginx进程

ps -ef | grep nginx

9、关闭nginx

./nginx -s stop

或者 粗暴点 ./nginx -s **quit**

10、重启nginx

./nginx -s **reload**

11、查看nginx配置文件是否正确

./nginx -t

12、Nginx 添加进环境变量

vim /etc/profile

在文件最后一行加上：

export PATH=/usr/local/nginx/sbin:$PATH

13、如果需要加入开机启动：

在 /etc/rc.local 中增加启动代码即可

vi /etc/rc.local

/usr/local/nginx/sbin/nginx

查看全部端口命令：netstat -ntpl

杀死某个进程: kill -9 pid

查看某个日志. Cat 日志（在打包会生存msg）

最后在云服务器开放安全组,开放入站端口（3306，6379，80，8080）这些端口分别为数据库,redis,nginx端口,8080后端端口.为了系统的安全建议把这些端口全部改掉安全点

部署按下面大神一部一个脚印去弄

为了提高并发量,最好设置下Nginx限流,Nginx限流可以参考（开源中国某张限流的第二个） 我没实测过

## 一：部署linux + nginx

1. 部署后台代码

1.1由于本人采用的是idea工具开发，所以在terminal中mvn clean package生成对应的jar包，此jar包可去对应文件所在目录的target查找。

1.2将jar复制到linux之中。

1. 部署前端代码

2.1使用npm run build:prod --report将前端项目打包，将会在前端目录下生成一个dist文件夹。

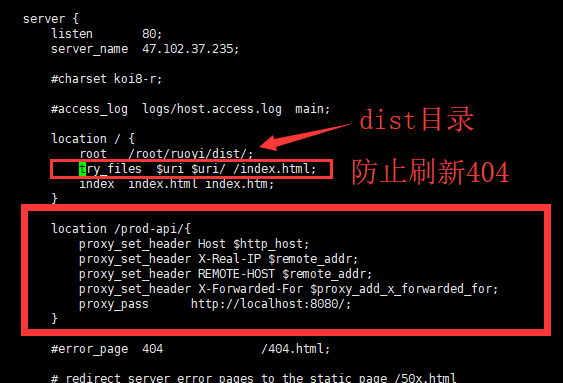
2.2同样将dist复制到linux之中。

3. linux服务器中需要装载redis以及nginx，redis是存放缓存数据，nginx是用于代理前后端服务。

3.1 安装redis并启动。

3.2 安装nginx并修改相关配置。

找到nginx所在目录，并修改conf/nginx.conf文件，修改如下图：



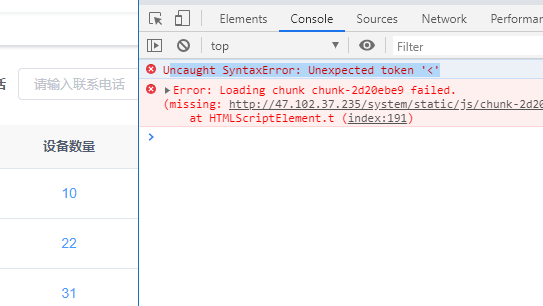
4．启动jar包，并启动nginx，在windows浏览器中直接输入ip即可访问。

4.1jar后台启动，后台启动的话可以进行其他操作，否则将无法进行其他操作。

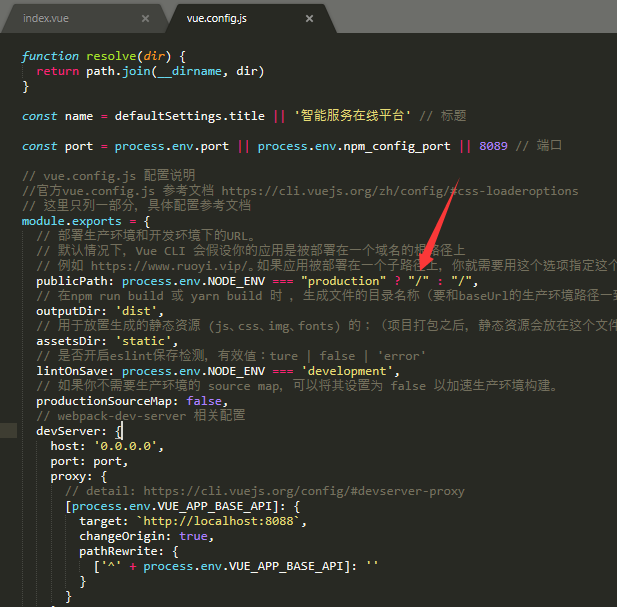
nohup java -jar jarName >msg.log 2>&1 &

msg.log文件是在输入上一行启动命令的当前目录下

5．有的小伙伴可能第一次打包放上去没问题，但是后期修改了一些文件或者菜单等，再部署上去可能会遇到菜单打不开之类的问题，



不要慌，只要修改vue.config.js文件里面的配置就好，如下图，去掉/之前的.即可

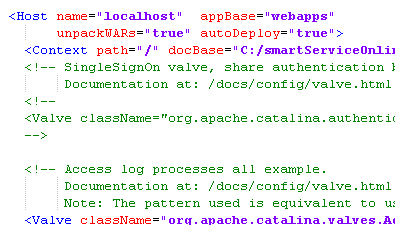


## 二：部署windows + tomcat(无nginx)

1. 修改pom文件，改jar为war:



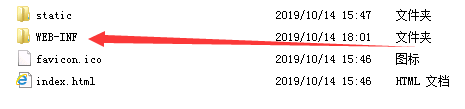
1. 打包成war包之后直接复制到windows中的tomcat的webapps目录下，并修改名称为prod-api，类似部署在linux中需要修改nginx.conf的配置。
2. 将vue项目打包并把dist文件复制到tomcat的webapps目录下。
3. 修改tomcat目录下的conf文件夹下的service.xml:
4. 启动端口为后台的启动端口
5. 在host节点下添加对应的访问路径配置代码



<Context path="/" docBase="dist的绝对路径(也可以写相对于webapps的路径)" reloadable="true" crossContext="true"></Context>

如果不添加此处代码，登录是可以的，但是登陆之后是空白页，因为在本地的运行的项目是类似ip:端口/router地址的，如果不修改，将会导致前面所说的问题。（不知道说的对不对，个人猜想）

1. 启动tomcat,输入ip:端口即可。
2. 以上操作之后刷新页面的话会存在404错误，所以需要在dist目录的文件夹下新建WEB-INF文件夹，并在里面添加web.xml文件:



web.xml文件代码如下：

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"

version="3.1" metadata-complete="true">

<display-name>Router for Tomcat</display-name>

<error-page>

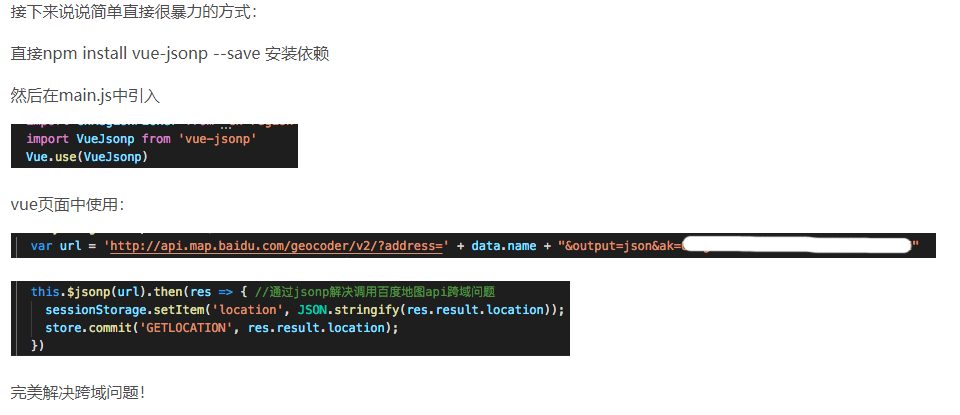
<error-code>404</error-code>

<location>/index.html</location>

</error-page>

</web-app>

**三：解决调用第三方api如百度地图api所存在的跨域问题**



**后端打成jar包跳过测试包为:**

mvn clean package -Dmaven.test.skip=true

**总结下：**

**1.部署的时候我的后台端口是8080端口的,nginx的后台 /prod-api{}里面反向代理的接口要对应上,比如我云上的ip地址为117.25.256.68，那端口要为117.25.256.68:8080**

**坑点;后端的application-druid.yml里面的数据库的主从库,也就是master,savle库的设置,我贴下代码**

**在druid:**

master:  
 url: jdbc:mysql:// **117.25.256.68**:3306/resourcesite?useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertToNull&useSSL=true&serverTimezone=GMT%2B8  
 username: root  
 password: 你的数据库密码

**slave**:  
 url: jdbc:mysql:// **117.25.256.68**:3306/resourcesite?useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertToNull&useSSL=true&serverTimezone=GMT%2B8  
 username: root  
 password: 你的数据库密码

**上面的url,username,password别写,因为我之前在本地跑的没啥问题,部署到服务器上去,一直给我报请求超时100000,很烦人的**

**改成作者的为空就可以了**

**slave**:  
 url:

username:   
 password:

**前端代码**

production：生产环境

development: 开发环境

**修改vue.config.js下面的代码:**

const port = process.env.port || process.env.npm\_config\_port || 80 // nginx对应的端口

80为nginx对应的端口,这里改了,nginx.conf的nginx端口也要响应改变

开发者模式 打包后成生产者模式

//lintOnSave: process.env.NODE\_ENV === 'development'

,

//lintOnSave: process.env.NODE\_ENV ===‘ production’

开发者模式

// [process.env.VUE\_APP\_BASE\_API]: {

// target: `http://localhost:8080`,

// changeOrigin: true,

// pathRewrite: {

// ['^' + process.env.VUE\_APP\_BASE\_API]: ''

// }

// }

开发者模式改成生产者模式

[process.env.VUE\_APP\_BASE\_API]: {

target: `http:// **117.25.256.68**:8080`,

changeOrigin: true,

pathRewrite: {

['^' + process.env.VUE\_APP\_BASE\_API]: ''

}

}

http:// **117.25.256.68**:8080 => 后端的ip对应后端的端口

*config*

// https://webpack.js.org/configuration/devtool/#development

.when(process.env.NODE\_ENV === ‘development’,

*config* => *config*.devtool('cheap-source-map')

)

**换成生产打包**

*config*

// https://webpack.js.org/configuration/devtool/#development

.when(process.env.NODE\_ENV === 'production',

*config* => *config*.devtool('cheap-source-map')

)

*config*

.when(process.env.NODE\_ENV !== ‘development’,

*config* => {

*config*

.plugin('ScriptExtHtmlWebpackPlugin')

开发者换生产者打包

*config*

.when(process.env.NODE\_ENV !== 'production',

*config* => {

*config*

.plugin('ScriptExtHtmlWebpackPlugin')

我记得icon的格式为favication.icon格式的

这里需要改下显示标题的图标,我上传过去没显示出来

我的在vscodeject软件,有个望👀

贴这个代码搜索

<link rel="icon" href="<%= BASE\_URL %>favicon.ico">

这是显示网站图片的代码

这里我不知道为啥我没显示出来

后来我改成把ico成七牛上传就可以,阿里云好像不是别ico

,最终上传到七牛

<link rel="icon" href="七牛图片的地址 ">

科普:icon怎么搞的呢,首先你得有个网站的图标,然后百度搜索图标换成icon，选择16\*16

如果在idea的话

我的是command+shirt+R（window下ctrl+shirt+R（ ）全局搜索,选择全部文件

最后执行打包的工具

**Mysql,redis改成远程服务器117.25.256.68 端口可以改变,建议全部都变,redis加密码要不很不安全**

**祝各位越来越好,我也是小白一个，我vue框架啥也不会,也在这个框架上写出了些东西,我把图片和富文本的图片都上传阿里云了,最后说下,如果在数据库里面回显不出来自己想要的图片都话,记得要对应结果状态码**

**这里要根据你云获取到的地址和状态码来搞的,假设你在网上的人家地址是url,状态吗是data，你也傻傻的拿别人过来,也不会修改的,那就是愚笨,我就试过。要学会f12,如果传过来失败的话,记得在失败的函数之前,打个controle.log(你的值),这个是你假设你阿里云可以上传成功,图片没回显成功的原因**

**后端传过来是一个formData类型,记得把token传过去,可以参考图片上传里面如何上传**

**就这样说吧假设你的图片为**

**This.form.image = 阿里云的地址（name,url）**

**打个比如:{**

**Name:这是阿里云的地址**

**Reponsese:“suceess”. 状态码**

**}**

**常规的写法**

**{**

[**url:这是阿里云的地址**](url:这是阿里云的地址)

**data: 200**

**}**

在云服务器上要学会看日志要 ,http协议很重要.不会的话你很难有很大发展的,至少解决问题这一块就很吃力,我吃过。

向大佬们学习,我是菜鸡