## 将应用部署到linux服务器教程

1. 购买服务器

推荐阿里云，在校学生可以每年领取300元现金券，可以免费购买一台机器一年

[阿里云高校计划\_云工开物\_助力高校科研与教育加速-阿里云 (aliyun.com)](https://university.aliyun.com/?spm=5176.28508143.J_ahRFo5CaAe_asSOaCgS4J.10.6c16154a9BdeoY&scm=20140722.M_10071128.P_157.MO_2278-ID_10071128-MID_10071128-CID_30881-ST_9425-V_1)

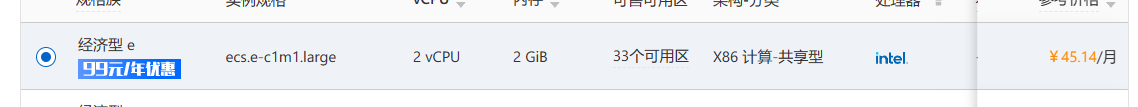


注册账号并登录后，验证学生身份，点击立即领取

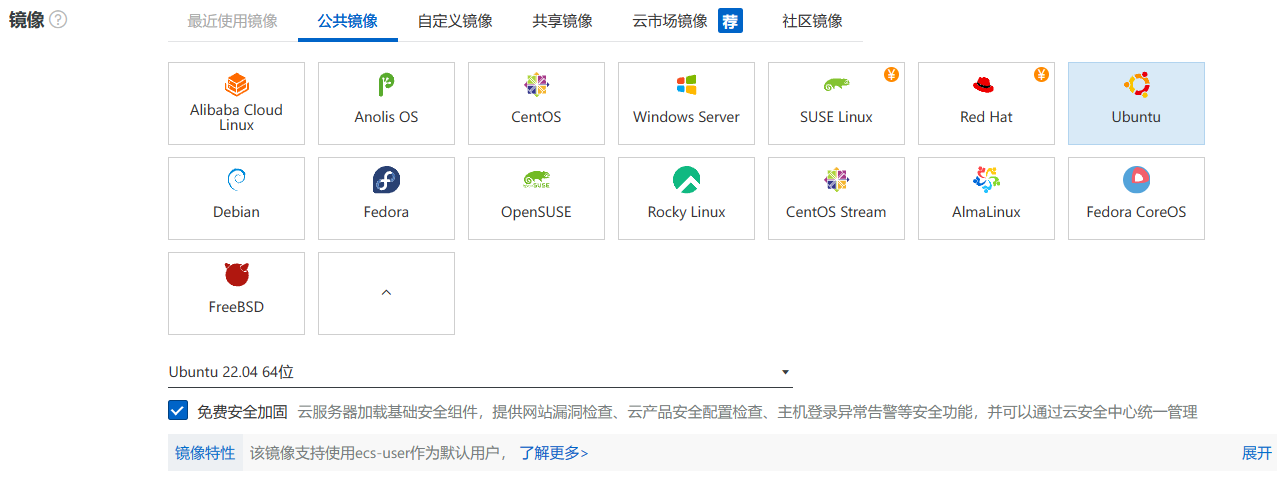
进入服务器购买页面

[云服务器ECS\_云主机\_服务器托管\_计算-阿里云 (aliyun.com)](https://www.aliyun.com/product/ecs?spm=5176.28623341.J_4VYgf18xNlTAyFFbOuOQe.29.24cc4519K1v0qi)

点击立即购买



推荐此配置和镜像，其他配置按需求选择，使用密码登录



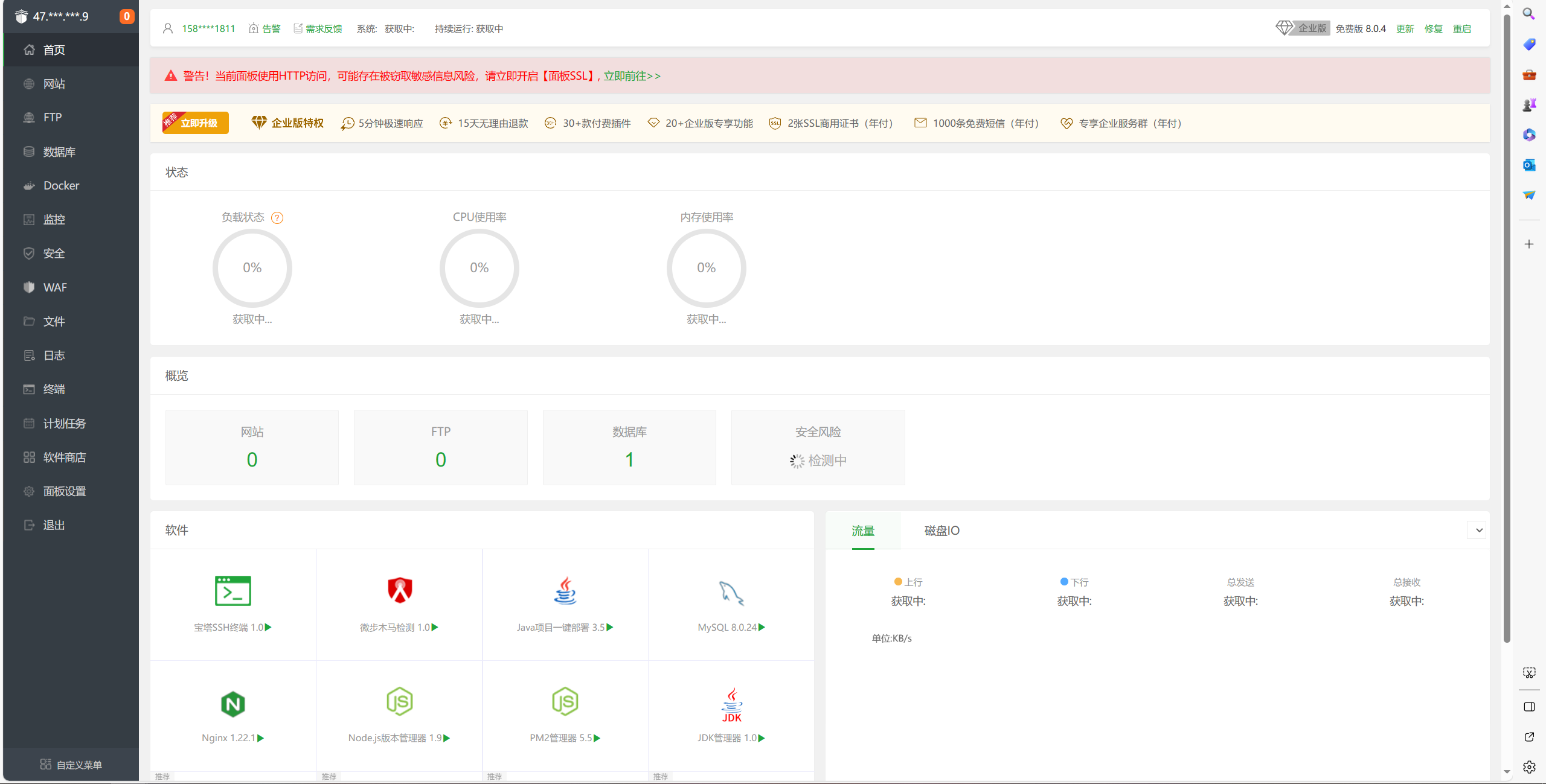
1. 购买完成后进入ecs控制台，点击你购买的服务器，并远程连接

[云服务器管理控制台 (aliyun.com)](https://ecs.console.aliyun.com/#/home)

使用此命令安装宝塔面板，用于后续操作

wget -O install.sh https://download.bt.cn/install/install-ubuntu\_6.0.sh && sudo bash install.sh ed8484bec

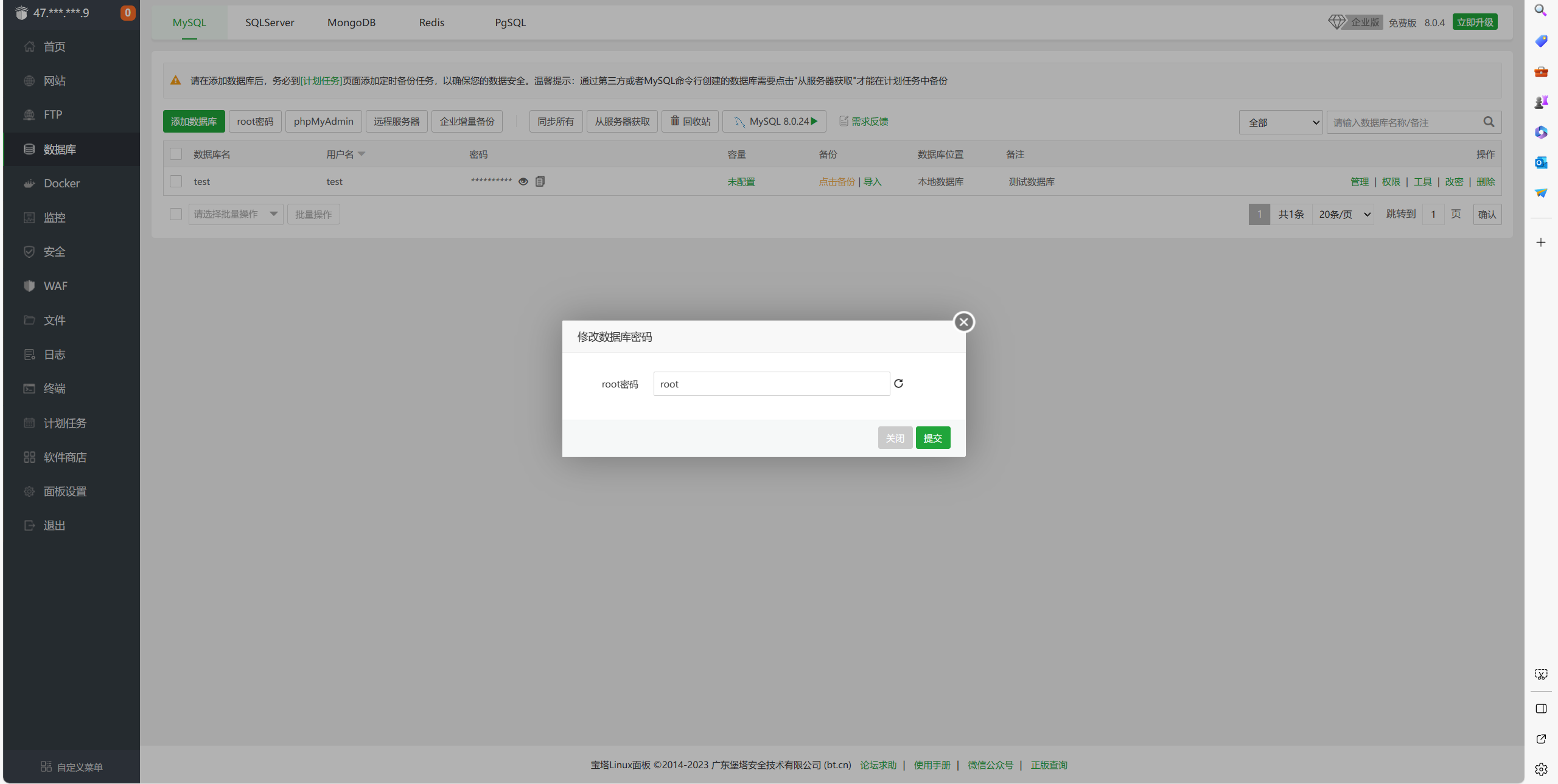
安装完成后，使用默认入口，用户名，密码，登录宝塔面板



点击网站，php项目，安装nginx；使用终端安装jdk1.8，命令如下

sudo apt-get install openjdk-8-jdk

点击数据库，安装mysql8，并重置root密码为root



至此，环境配置完成

在终端使用java -version检查jdk版本，如不为1.8可通过以下方式更改版本

输入

update-alternatives --config java

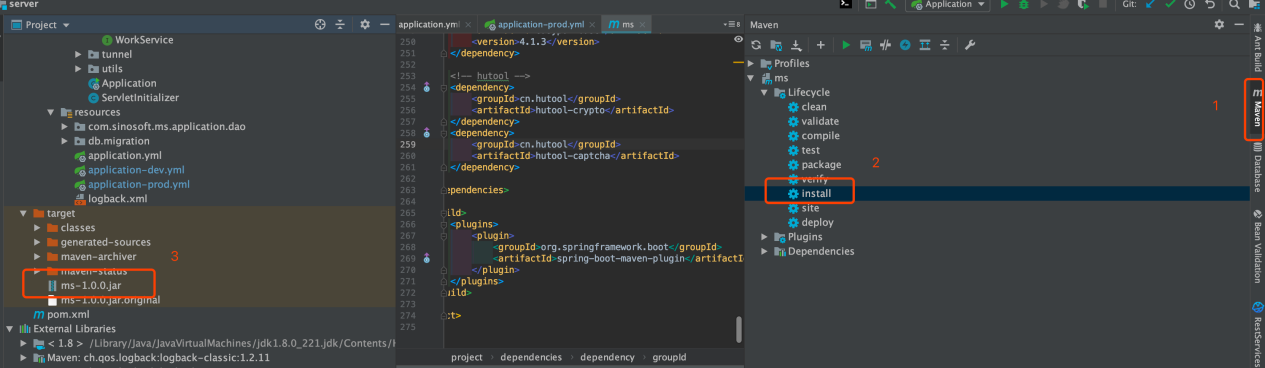
输入jdk1.8的版本对应的数字，如1，2

点击回车

输入java -version检查jdk版本是否已经更换

3. 服务端程序打包

1. 通过idea的maven插件打包



打包完成后会在target目录下生成 jar 文件包

客户端程序打包

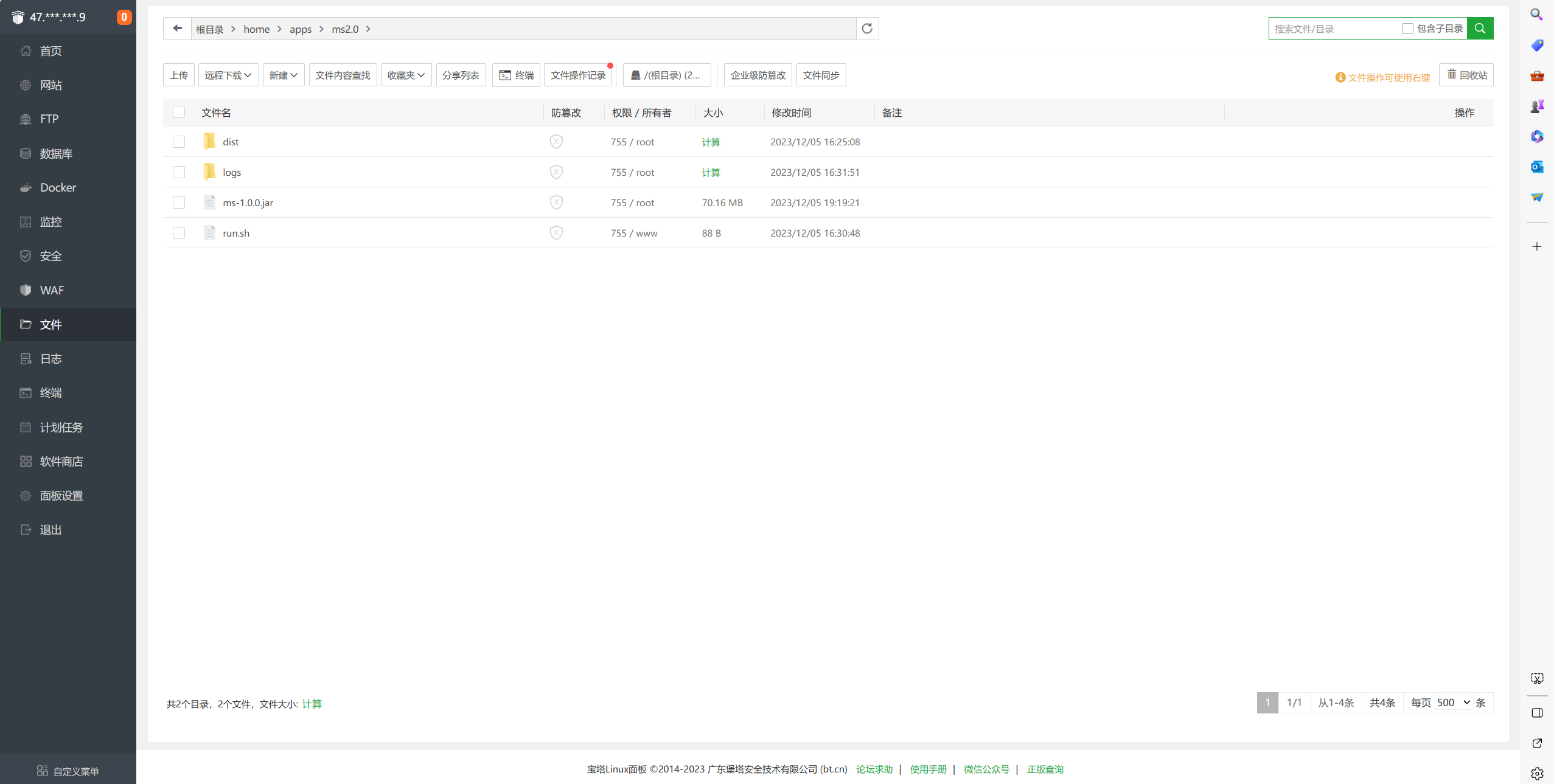
1. 打开package.json文件，如果当前电脑是windows系统，需要修改 scripts 脚本中的build命令为：

set NODE\_ENV=production & webpack --progress --config ./build/webpack.prod.js

1. 在命令行输入 npm run build 进行打包

打包完成后会生成dist目录

3.打开宝塔面板的文件管理将打包好的jar文件和dist目录上传到此路径，如果路径不存在则需要创建。logs为系统生成的日志，不需要理会，run.sh为后端启动命令文件，可以通过直接在终端输入命令启动来代替



1. 在终端上配置mysql8服务器，允许远程登录

终端输入

mysql -u root -p

输入密码

输入

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '你的密码，为root' WITH GRANT OPTION;

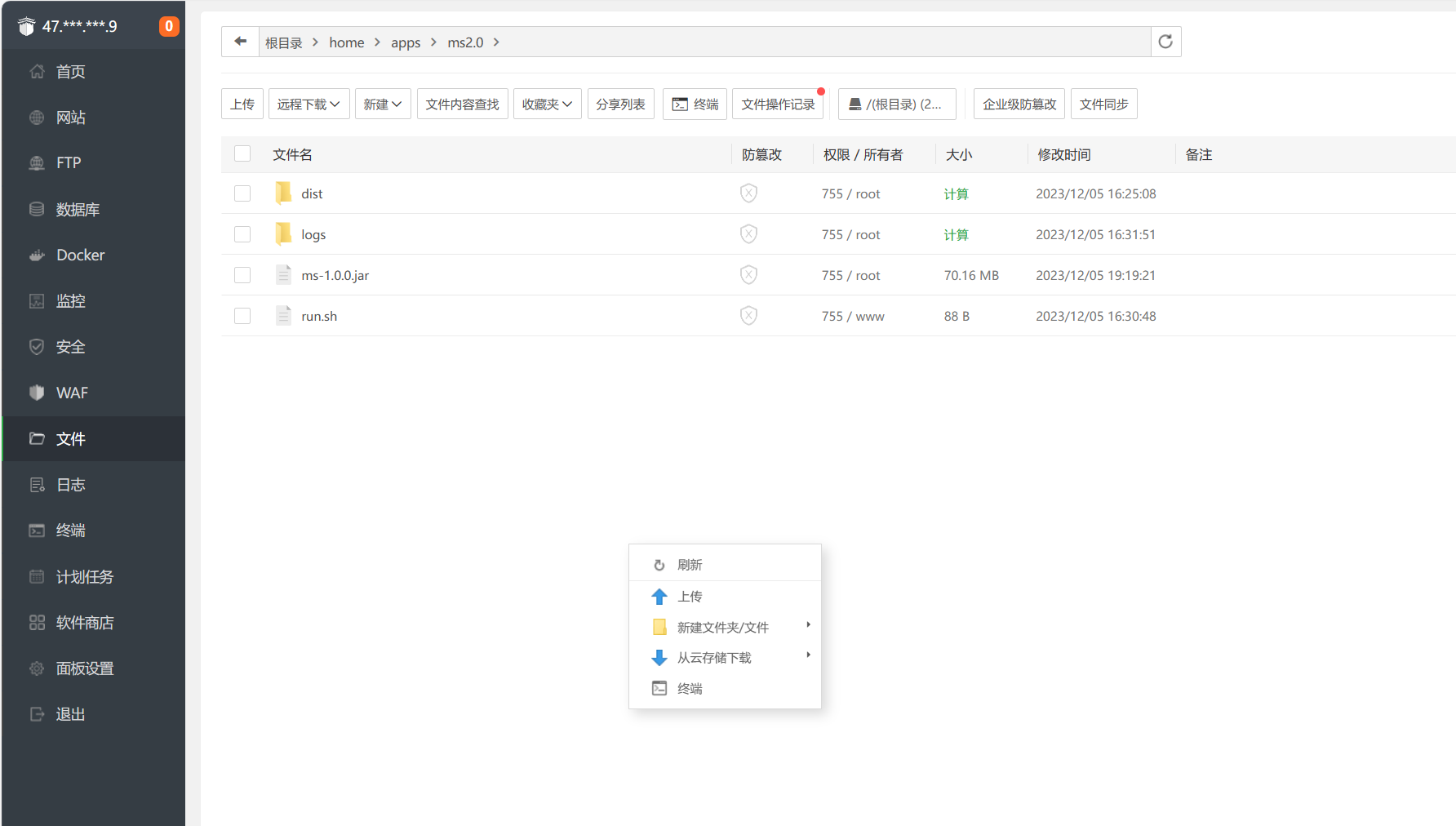
输入

FLUSH PRIVILEGES;

通过navicat连接你的mysql服务器，地址为服务器ip，端口3306，用户名root，密码为root

将数据库脚本在此服务器上运行并刷新，确定数据库已成功创建

1. 右键上面的文件目录，选择终端

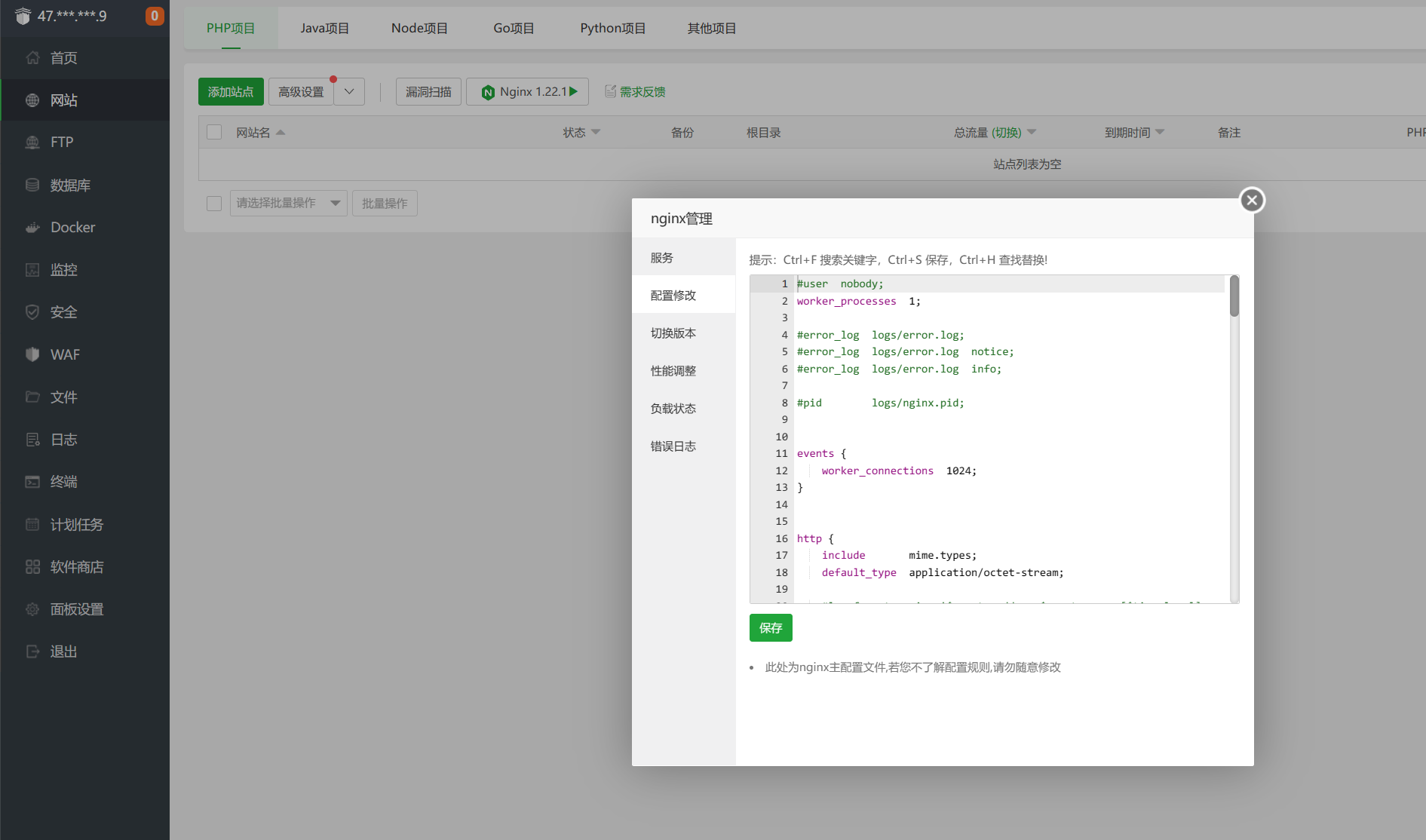


输入

nohup java -jar -Dspring.profiles.active=prod ./ms-1.0.0.jar > /dev/null &

关闭终端

1. 点击网站，php项目，点击nginx，修改配置文件



#user nobody;

worker\_processes 1;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

#pid logs/nginx.pid;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

#log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

# '$status $body\_bytes\_sent "$http\_referer" '

# '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

#access\_log logs/access.log main;

sendfile on;

#tcp\_nopush on;

#keepalive\_timeout 0;

keepalive\_timeout 65;

#gzip on;

server {

listen 8080;

server\_name localhost;

location ^~ /download/ {

root /home/apps/ms2.0/download;

sendfile on;

charset utf-8,gbk;

}

location / {

try\_files $uri /index.html;

root /home/apps/ms2.0/dist;

index index.html index.htm;

}

location /ms {

proxy\_pass http://127.0.0.1:8081/ms;

proxy\_redirect off;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

client\_max\_body\_size 10m;

proxy\_connect\_timeout 90;

proxy\_read\_timeout 90;

proxy\_set\_header Cookie $http\_cookie;

}

}

server {

listen 80;

server\_name localhost;

#charset koi8-r;

#access\_log logs/host.access.log main;

# location /download {

root /home/apps/ms2.0/download;

sendfile on;

autoindex on;

autoindex\_exact\_size off;

autoindex\_localtime on;

charset utf-8,gbk;

# }

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80

#

#location ~ \.php$ {

# proxy\_pass http://127.0.0.1;

#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#

#location ~ \.php$ {

# root html;

# fastcgi\_pass 127.0.0.1:9000;

# fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

# include fastcgi\_params;

#}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

#location ~ /\.ht {

# deny all;

#}

}

# another virtual host using mix of IP-, name-, and port-based configuration

#

#server {

# listen 8000;

# listen somename:8080;

# server\_name somename alias another.alias;

# location / {

# root html;

# index index.html index.htm;

# }

#}

# HTTPS server

#

#server {

# listen 443 ssl;

# server\_name localhost;

# ssl\_certificate cert.pem;

# ssl\_certificate\_key cert.key;

# ssl\_session\_cache shared:SSL:1m;

# ssl\_session\_timeout 5m;

# ssl\_ciphers HIGH:!aNULL:!MD5;

# ssl\_prefer\_server\_ciphers on;

# location / {

# root html;

# index index.html index.htm;

# }

#}

}

保存后重启nginx服务

7，浏览器输入

你的服务器ip:8080进入项目，至此，项目已成功部署到服务器