# IM宝塔系统部署文档

### 本系统使用必须符合国家法律法规 具体法律规定看文档

### 自己准备好服务器和域名 等资料

**PC端、移动端域、后台域名全部添入宝塔网站**

**然后用户端按照下方步骤 部署服务端代码 部署完成后 使用部署文档说明中的nginx配置文件都配置一遍**

**讲解一下我们为了稳定快速提问措施方案，**

**方案一、使用美国服务器，直接部署，但是会比较卡，就是单独一个服务器**

**方案二、最优方案。使用分布式部署方案购买一个美国服务器，解析一个域名，部署服务端，配置一下socket这些，这个只用于请求处理socket购买一个国内服务器 解析两个域名 部署服务端 用户端 管理端 用户端和管理端代码下 ，需要把wss系列的全部替换使用国外服务器域名 其他的全局替换为国内服务器域名即可**

**注：国外服务器同样需要安装宝塔，按照下面的步骤，国外的服务需要链接国内的数据库**

# 文档以centOS为例说明【系统系统 可以按照宝塔官方地址安装】

# 一、宝塔官网地址

[https://www.bt.cn/download/linux.html](https://www.bt.cn/download/linux.html?accessToken=eyJhbGciOiJIUzI1NiIsImtpZCI6ImRlZmF1bHQiLCJ0eXAiOiJKV1QifQ.eyJleHAiOjE2ODM1Mjk5MjEsImZpbGVHVUlEIjoiOTAzME1abTRnWVNwcHF3cCIsImlhdCI6MTY4MzUyOTYyMSwiaXNzIjoidXBsb2FkZXJfYWNjZXNzX3Jlc291cmNlIiwidXNlcklkIjo4OTEyMzk5Nn0.lQ9p_HghduwYQ5LCyqdC4xUqxoz3D30slAyqNivO6Js)

# 二、选择匹配自己系统的直接安装



img0.image

# 三、访问linux服务器安装宝塔

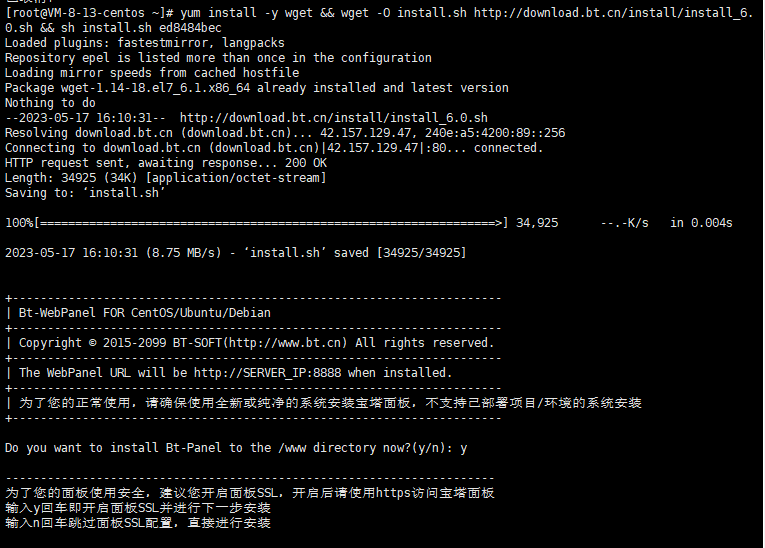
### 1、执行超级权限授权命令

sudo su

### 2、安装centOS服务

yum install -y wget && wget -O install.sh http://download.bt.cn/install/install\_6.0.sh && sh install.sh ed8484bec

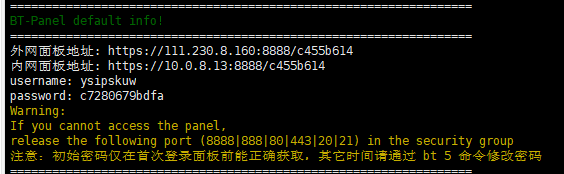
### 3、如图 输入y



img1.image

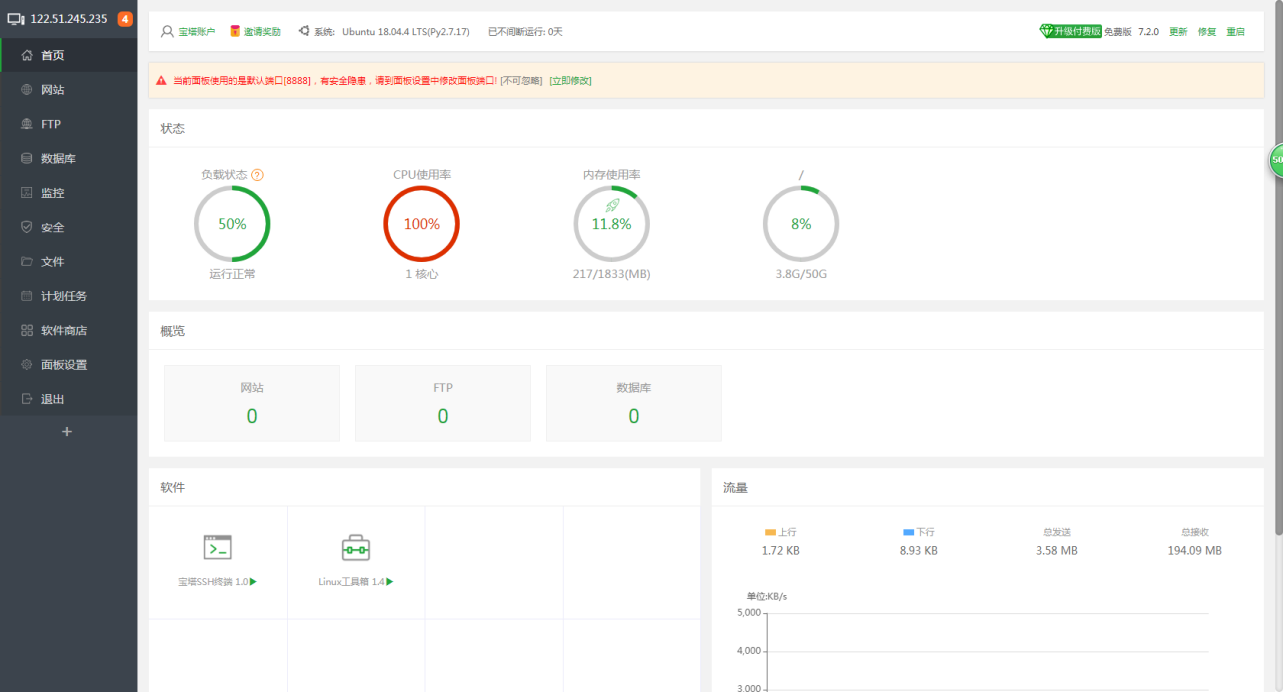
### 4、成功后复制账号密码在浏览器访问

如果无法访问请你自己开启 服务器中的安全组 放开8888端口



img2.image

# 四、登录宝塔界面



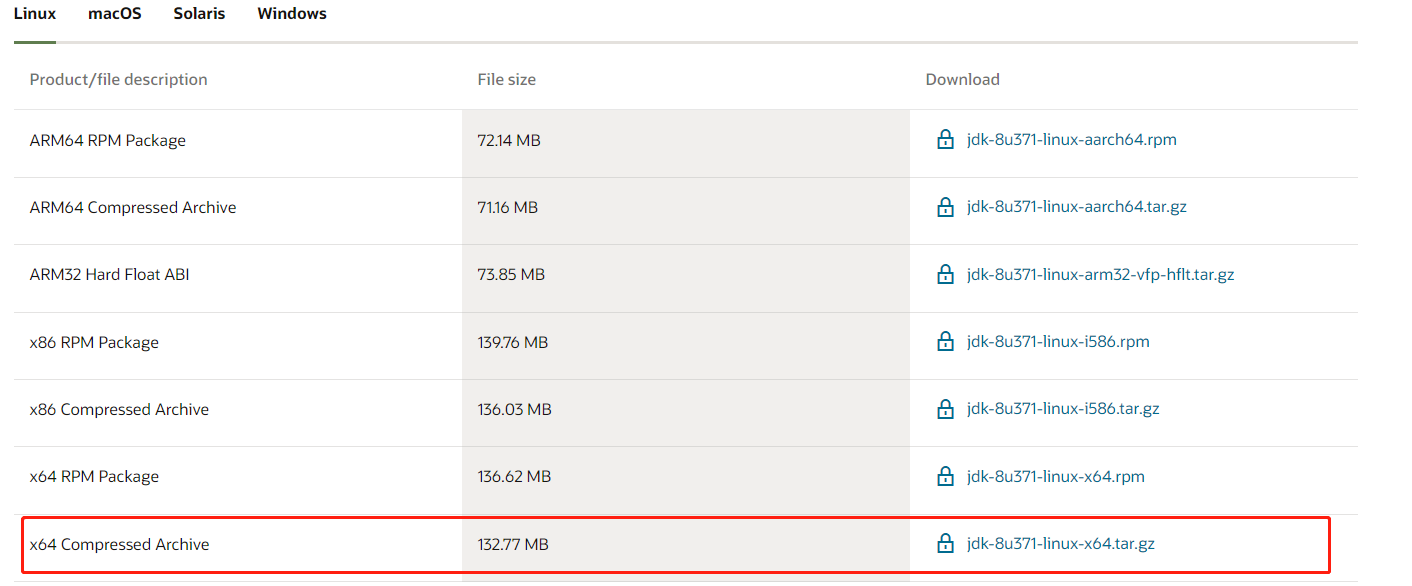
10014.image

# 五、安装基础配置

#### 1.安装JDK 下载jdk8u\_371

检查是否自带JDK #1.检查当前机器是否有自带的JDK rpm -qa |grep java rpm -qa |grep jdk rpm -qa |grep gcj #2.如果没有 则跳至安装步骤，有的话 进行卸载 rpm -qa | grep java | xargs rpm -e –nodeps #3.检测卸载是否成功 java -version #出现一下提示则说明没有安装JDK或者已经卸载成功 -bash: java: command not found

Oracle官网下载：https://www.oracle.com/java/technologies/downloads/#java8



img3.image

上传下载的压缩包 #在服务器上新建Java目录 mkdir /usr/local/java

我是通过SCP命令上传的，你可以按照你想用的任何方式去上传JDK的压缩包到你的服务器上，方式我上面有提示过：

例如使用ftp工具、Xshell的文件上传工具、cmd命令下使用SCP命令上传、宝塔面板上使用上传文件功能、WinSCP工具等等

解压JDK压缩包 tar -zxvf jdk-8u371-linux-x64.tar.gz

配置环境变量 #编辑配置文件 vim /etc/profile #添加JDK配置

jdk配置

export JAVA\_HOME=/usr/local/java/jdk1.8.0\_371 export JRE\_HOME=JAVA\_HOME/bin:PATH export CLASSPATH=JAVA\_HOME/lib:$JRE\_HOME/lib #按esc退出按wq!保存退出

#刷新配置 source /etc/profile

输入命令java -version 查看JDK安装情况

输入命令java 查看JDK安装情况

输入命令javac 查看JDK安装情况

#### 2.CentOS宝塔安装maven

查看是否安装 首先你需要用xshell 或者终端命令

mvn -version

img4.image

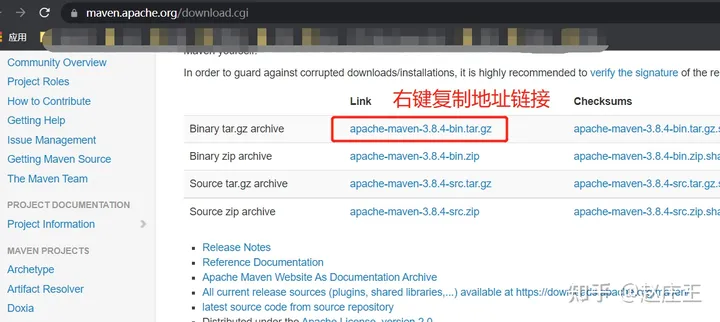
img4.image

下载 Maven [下载地址](https://maven.apache.org/download.cgi)

wget https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz

yum -y install wget

截图的版本为 当前所用



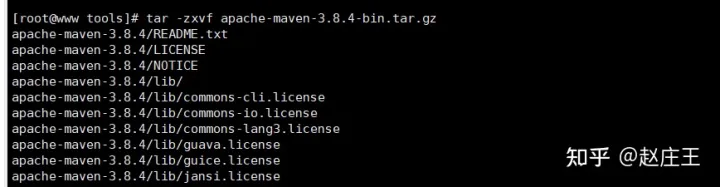
img5.image



img6.image

解压 解压指令 maven压缩包

tar -zxvf apache-maven-3.8.4-bin.tar.gz



img7.image

配置环境变量 首先你需要知道你的文件所在位置

img8.image

img8.image

修改文件

vi /etc/profile 按 a 是获取编辑焦点，输入完之后按 ESC 也就是键盘左上角 ，

shift+: :wq 回车 在文件最后面加入下面配置

#你的maven路径 MAVEN\_HOME=/www/tools/apache-maven-3.8.4

export PATH={PATH}

添加配置

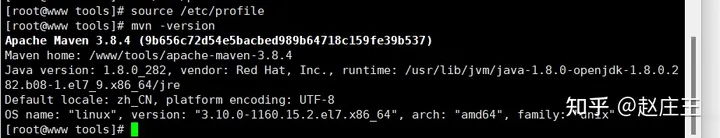


img9.image

重新加载

source /etc/profile

验证是否安装 mvn -version



img10.image

切换国内镜像 如果默认的话，下载会很慢 ，例子为 阿里镜像源 修改settings.xml文件

vi /www/tools/apache-maven-3.8.4/conf/settings.xml

yum 安装 安装路径，我是查找的 /usr/share/maven

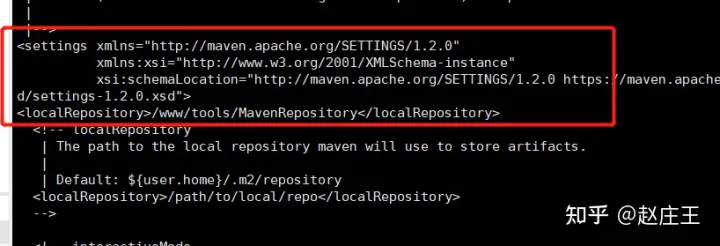
1 设置镜像，阿里

rm -rf /etc/yum.repos.d/\* curl -o /etc/yum.repos.d/Centos-7.repo http://mirrors.aliyun.com/repo/Centos-7.repo curl -o /etc/yum.repos.d/epel-7.repo http://mirrors.aliyun.com/repo/epel-7.repo 1 安装 yum -y install maven 2 切换国内镜像 如果默认的话，下载会很慢 ，例子为 阿里镜像源 修改settings.xml文件

vi /etc/maven/settings.xml 配置资源保存得位置 创建一个文件夹 位置随意，名称随意 ，找到 localRepository (基本在最上边)下面加入代码

代码中的 路径既是你的创建的文件夹名称 MavenRepository

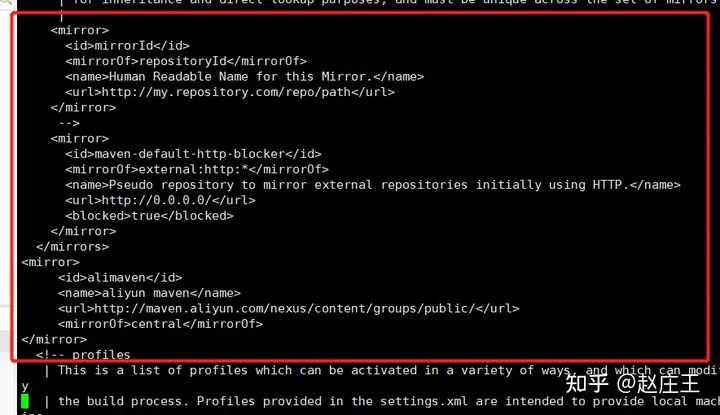
/www/tools/MavenRepository



img11.image

添加镜像 找到 mirrors 标签(基本快到最 底部了)对，在下方添加一下 代码

alimaven aliyun maven http://maven.aliyun.com/nexus/content/groups/public/ central



img12.image

修改关联 jdk 版本 查询你的 jdk版本

whereis java

ls -lrt /usr/bin/java

ls -lrt /etc/alternatives/java

/etc/alternatives/java -> /usr/java/jdk-18/bin/java

在修改你的 jav\_home

export JAVA\_HOME=/usr/java/jdk-18 找到 修改

vi /etc/maven/settings.xml

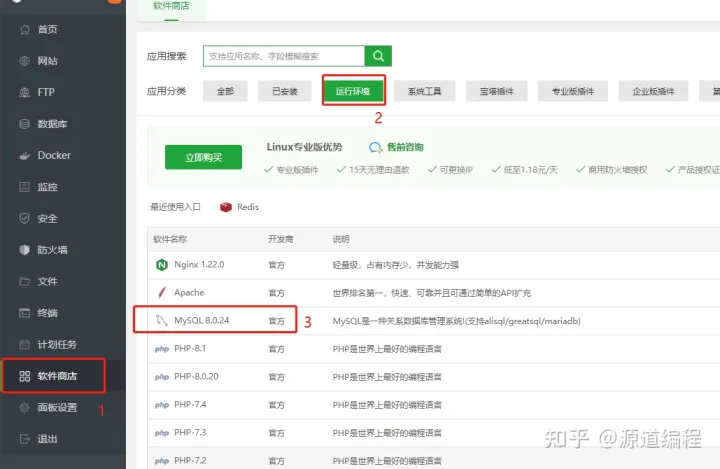
jdk-18 true 18 <maven.compiler.source>18</maven.compiler.source> <maven.compiler.target>18</maven.compiler.target> <maven.compiler.compilerVersion>18</maven.compiler.compilerVersion>

# 六、安装mysql redis nginx mongodb docker nacos seaweedfs 项目需要配置环境

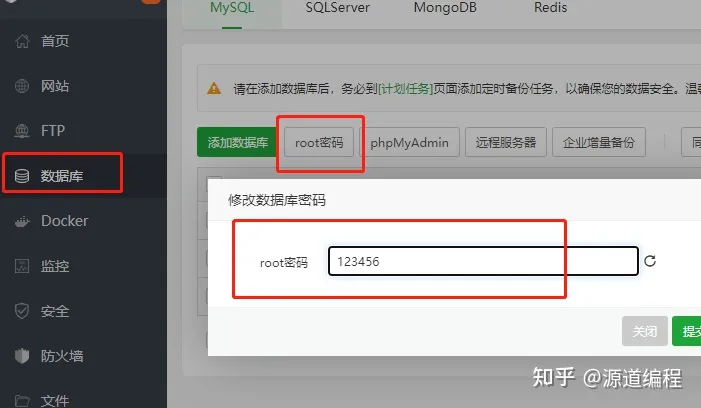
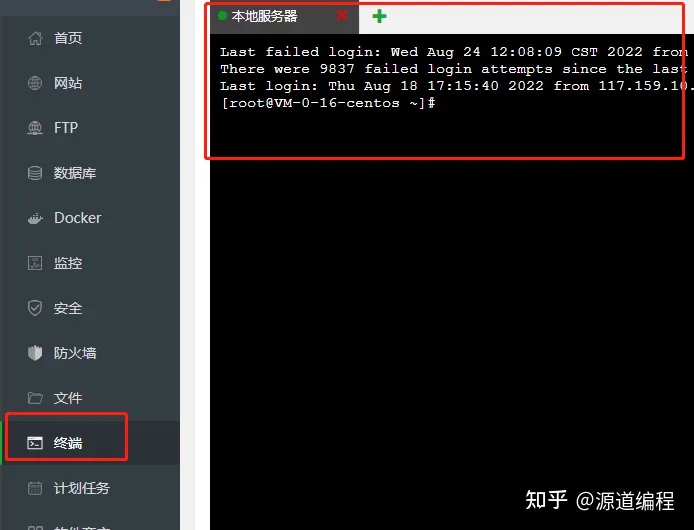
在应用商店安装mysql redis nginx mongodb docker nacos  
注：seaweedfs nacos需要在docker中安装

##### 1.登录到宝塔面板内安装mysql

点击软件商店，选择运行环境，找到mysql，点击后面的安装即可。mysql会自动安装完毕。



配置数据源远程连接：这时候数据库远程还是连不上的，需要命令行进入服务器，设置下才可以。具体步骤如下： 先在宝塔里查看下root用户的密码。复制出来。

 使用终端连接服务器、或者用宝塔里面的终端工具连接也行。 

开启3306端口 1.开启防火墙 systemctl start firewalld

1. 开启3306端口 firewall-cmd –zone=public –add-port=1935/tcp –permanent 命令含义： –zone #作用域 –add-port=1935/tcp #添加端口，

格式为：端口/通讯协议 –permanent #永久生效，没有此参数重启后失效

1. 重启防火墙 systemctl reload firewalld 4.查看端口号 netstat -ntlp //查看当前所有tcp端口· netstat -ntulp |grep 3306 //查看所有3306端口使用情况·

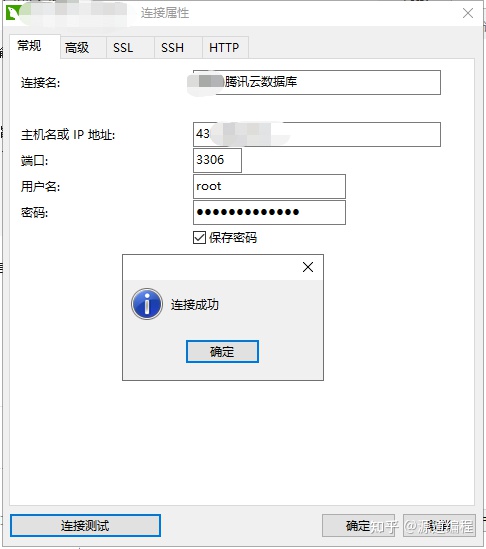
连上服务器后，执行下面步骤即可

【其他方式安装的mysql如果远程连接不上，也可以用下面的方法解决】

#使用root用户进入数据库 输入上面复制的root密码 mysql -u root -p

#使用mysql use mysql

#更新root用户权限，“%”指的是所有地址都可以访问 update user set Host=‘%’ where User=‘root’;

#最后一步，刷新权限即可 flush privileges; 现在即可用Navicat工具来远程连接数据库了。 

注：记得导入后台项目sql文件

##### 2.登录到宝塔面板内安装Nginx



img28.image

在Nginx管理界面，您将会看到有Web服务、配置修改、WAF防火墙、切换版本、负载状态等多个菜单项。切换到Web服务选项卡，在该页面中，您可以停止或重启当前Nginx服务和重新加载Nginx配置文件。  或者，您也可以进SSH输入以下命令来完成重载、重启等操作。 /etc/init.d/nginx start #启动nginx

/etc/init.d/nginx stop #停止nginx

/etc/init.d/nginx restart #重启nginx

/etc/init.d/nginx reload #重载nginx

/etc/init.d/nginx status #查看nginx当前状态

另外，您也可以切换到“配置修改”选项卡，修改当前的Nginx的配置文件，如果需要使当前配置生效，请在Web服务中重载配置文件。

此项目需要修改80 server配置文件如下 注：由于移动端需要进行语音和视频传播 必须要求链接是安全有效的 必须配置证书域名 server { listen 80; server\_name 域名;

listen 443 ssl;  
 #ssl on;  
 ssl\_certificate /etc/letsencrypt/live/域名/fullchain.pem;  
 ssl\_certificate\_key /etc/letsencrypt/live/域名/privkey.pem;  
  
  
 #charset koi8-r;  
  
 #access\_log logs/host.access.log main;  
  
 location / {  
 root 后台前端打包dist目录地址;  
 index index.html index.htm;  
 error\_log logs/f\_error.log;  
 }  
  
 location /api {  
 proxy\_set\_header Host $http\_host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header REMOTE-HOST $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 proxy\_pass http://localhost:端口;  
 }  
   
 location /app {  
 proxy\_set\_header Host $http\_host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header REMOTE-HOST $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 proxy\_pass http://154.204.59.100:8001/api/v1;  
 }  
   
 # 授权接口  
 location /auth {  
 proxy\_pass http://localhost:端口;  
 proxy\_set\_header X-Forwarded-Proto $scheme;  
 proxy\_set\_header X-Forwarded-Port $server\_port;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 proxy\_set\_header Upgrade $http\_upgrade;  
 proxy\_set\_header Connection "upgrade";  
 }  
  
 # WebSocket 服务  
 location /webSocket {  
 proxy\_redirect off;  
 proxy\_pass http://localhost:端口/webSocket;  
 proxy\_http\_version 1.1;  
 proxy\_set\_header Upgrade $http\_upgrade;  
 proxy\_set\_header Connection "upgrade";  
 proxy\_set\_header Host $http\_host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 proxy\_connect\_timeout 60s;  
 proxy\_read\_timeout 86400s;  
 proxy\_send\_timeout 60s;  
 }  
  
 # 头像  
 location /avatar {  
 proxy\_pass http://localhost:端口;  
 }  
  
 # 文件  
 location /file {  
 proxy\_pass http://localhost:端口;  
 }  
   
 error\_page 404 /index.html;  
}  
  
server {  
 listen 80;  
 server\_name 域名;  
  
 listen 443 ssl;  
 #ssl on;  
 ssl\_certificate /etc/letsencrypt/live/域名/fullchain.pem;  
 ssl\_certificate\_key /etc/letsencrypt/live/域名/privkey.pem;  
  
 #charset koi8-r;  
  
 #access\_log logs/host.access.log main;  
  
 location / {  
 root pc端打包dist目录;  
 index index.html index.htm;  
 error\_log logs/f\_error.log;  
 }  
  
 location /api/ {  
 proxy\_pass http://localhost:端口/api;  
 proxy\_connect\_timeout 300; #单位秒  
 proxy\_send\_timeout 300; #单位秒  
 proxy\_read\_timeout 300; #单位秒  
  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 }  
 location /mqtt {  
 proxy\_pass http://localhost:端口;  
 proxy\_http\_version 1.1;  
 proxy\_set\_header Upgrade $http\_upgrade;  
 proxy\_set\_header Connection "upgrade";  
 proxy\_set\_header Host $host;  
 }  
 error\_page 404 /index.html;  
}  
  
server {  
 listen 80;  
 server\_name 域名;  
  
 listen 443 ssl;  
 #ssl on;  
 ssl\_certificate /etc/letsencrypt/live/域名/fullchain.pem;  
 ssl\_certificate\_key /etc/letsencrypt/live/域名/privkey.pem;  
  
 #charset koi8-r;  
  
 #access\_log logs/host.access.log main;  
  
 location / {  
 root 移动端打包dist目录;  
 index index.html index.htm;  
 error\_log logs/f\_error.log;  
 }  
  
 location /api/ {  
 proxy\_pass http://localhost:端口/api;  
 proxy\_connect\_timeout 300; #单位秒  
 proxy\_send\_timeout 300; #单位秒  
 proxy\_read\_timeout 300; #单位秒  
  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 }  
  
 location /mqtt {  
 proxy\_pass http://localhost:端口;  
 proxy\_http\_version 1.1;  
 proxy\_set\_header Upgrade $http\_upgrade;  
 proxy\_set\_header Connection "upgrade";  
 proxy\_set\_header Host $host;   
 }  
 error\_page 404 /index.html;  
}

/etc/init.d/nginx reload #重载nginx

##### 3.登录宝塔面板安装Redis服务：

进入宝塔面板后台管理界面→选择软件管理→运行环境→Redis→点击安装，等待完成。 然后要使用PHP连接Redis的话，还需在PHP设置中安装Redis扩展：

找到/www/server/redis/中的“redis.conf”文件，打开编辑，找到“requirepass foobared”这行，将#注释符去掉 也可不去掉 默认没有密码 。

接下来，打开PHP对应的设置，选择“安装扩展” → redis → 安装。

Redis扩展安装好后，重启即可。

如果您想开启Redis服务，则可在宝塔面板中的Redis管理界面，停止或重启当前Redis服务和重新加载Redis配置文件。

亦可以进SSH输入以下命令来完成重载、重启等操作。

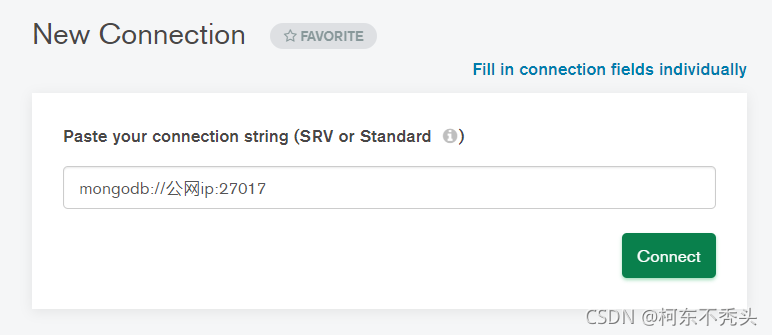
/etc/init.d/redis start#启动redis

/etc/init.d/redis stop#停止redis

/etc/init.d/redis restart#重启redis

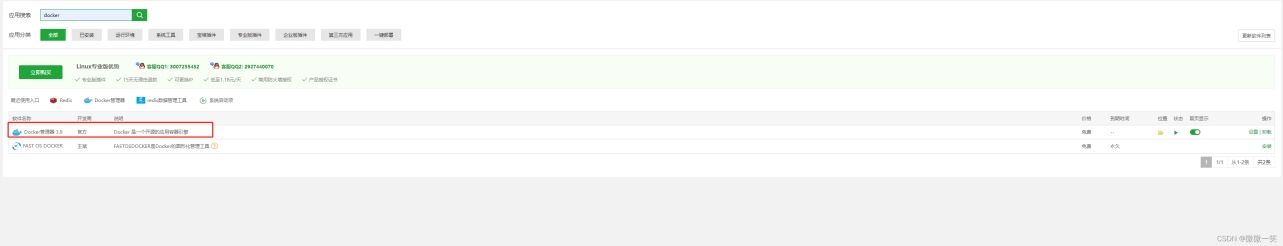
/etc/init.d/redis reload#重载redis

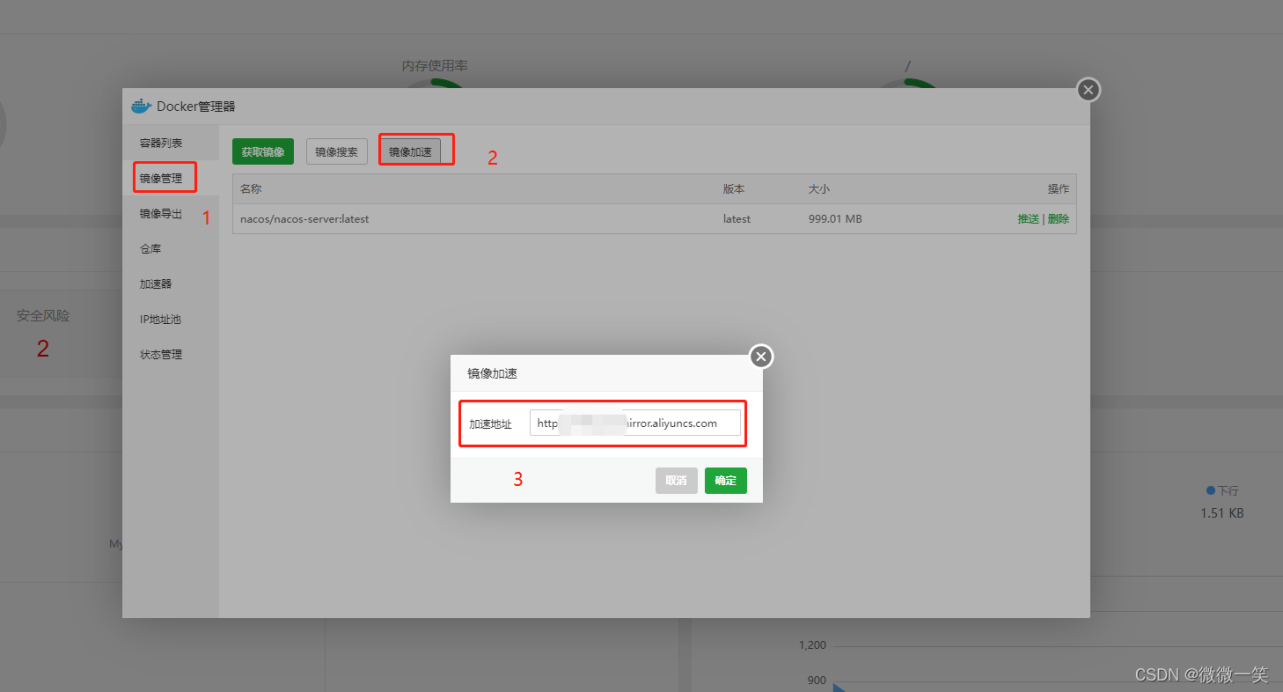
##### 4.登录宝塔安装mongodb

1.点击软件商店，搜索mongodb进行安装 2.找到安装好的mongodb进行设置，将配置文件bindIp改为0.0.0.0 img15.image 阿里云服务器安全组入方向开放端口27017 img16.image 防火墙打开端口27017 img15.image 浏览器输入http://公网ip:27017 出现“It looks like you are trying to access MongoDB over HTTP on the native driver port.”说明配置成功 二、使用MongoDB Compass可视化工具连接数据库 无密码连接 

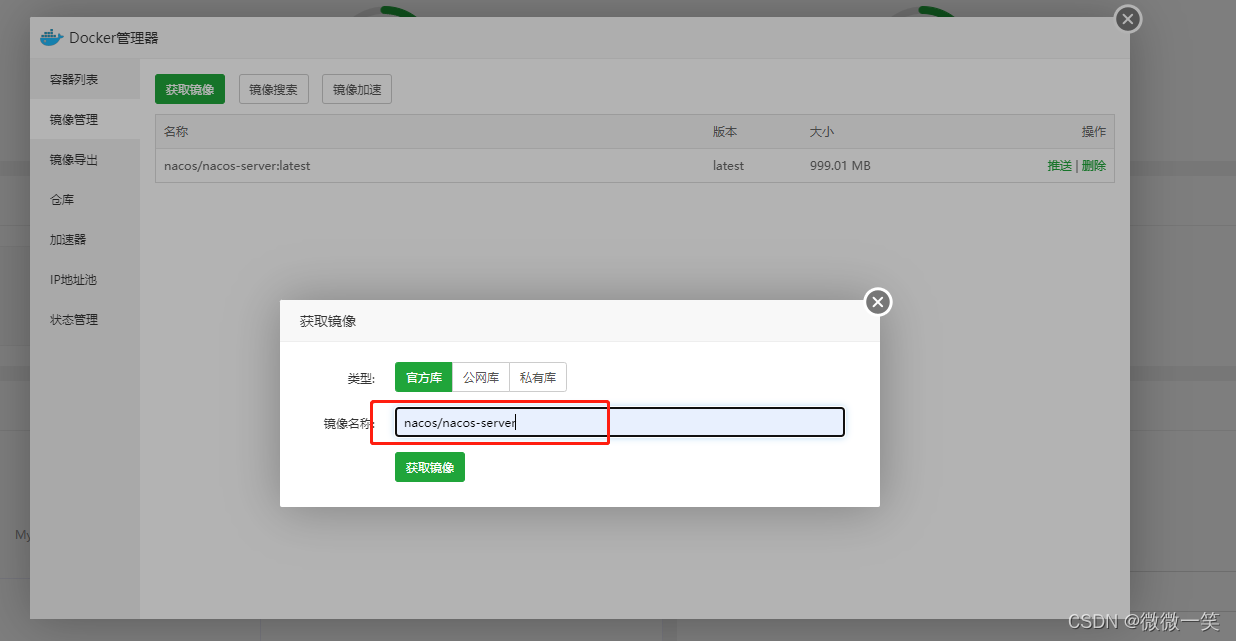
注：记得导入移动端后台数据库文件到mongodb

##### 5.登录宝塔安装docker

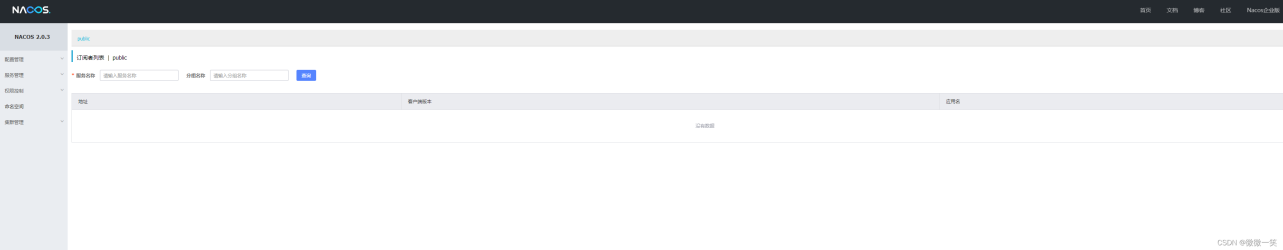
在商店搜索docker 

打开docker管理器 使用阿里云的镜像加速服务: https://cr.console.aliyun.com/cn-hangzhou/instances/mirrors 

##### 6.获取nacos镜像

  容器下载完毕, 不知道怎么安装没关系, 看下边 ↓ 启动nacos镜像 打开宝塔面板上的终端执行以下命令: 命令中已设置了开机自启动

docker run  
–name nacos -d  
-p 8848:8848  
–privileged=true  
–restart=always  
-e JVM\_XMS=256m  
-e JVM\_XMX=256m  
-e MODE=standalone  
-e PREFER\_HOST\_MODE=hostname  
nacos/nacos-server:latest

安装成功 在浏览器上输入访问地址：http://IP地址:8848/nacos，测试是否安装成攻 账号和密码默认是: nacos/nacos 

注：记得导入nacos配置文件和修改项目nacos配置文件修改ip和端口

##### 7.安装seaweedf：

方式一、docker部署seaweedf 总的来说需要两个文件，如下： img30.image 1、compose.yml

需要注意：network\_mode 模式、ip为宿主机ip

version: ‘3.3’

services: master: network\_mode: “host” image: chrislusf/seaweedfs # use a remote image command: “master -ip=172.0.16.143 -ip.bind=172.0.16.143 -metricsPort=9324” volume: network\_mode: “host” image: chrislusf/seaweedfs # use a remote image command: ‘volume -mserver=“172.0.16.143:9333” -ip.bind=172.0.16.143 -port=8080 -metricsPort=9325’ depends\_on: - master filer: network\_mode: “host” image: chrislusf/seaweedfs # use a remote image command: ‘filer -master=“172.0.16.143:9333” -ip.bind=172.0.16.143 -metricsPort=9326’ tty: true stdin\_open: true depends\_on: - master - volume s3: network\_mode: “host” image: chrislusf/seaweedfs # use a remote image command: ‘s3 -filer=“172.0.16.143:8888” -ip.bind=172.0.16.143 -metricsPort=9327’ depends\_on: - master - volume - filer webdav: network\_mode: “host” image: chrislusf/seaweedfs # use a remote image command: ‘webdav -filer=“172.0.16.143:8888”’ depends\_on: - master - volume - filer

prometheus:

image: prom/prometheus:v2.21.0

ports:

​ 9000:9090

volumes:

​ ./prometheus:/etc/prometheus

command: –web.enable-lifecycle –config.file=/etc/prometheus/prometheus.yml

depends\_on:

​ s3

2、install.sh docker-compose -f seaweedfs-compose.yml -p seaweedfs up

方式二、手动安装 SeaWeedfs安装与启动 Linux安装SeaWeedFS mkdir -p /myapp/seaweedfs cd /myapp/seaweedfs 访问Releases · seaweedfs/seaweedfs · GitHub下载linux\_amd64.tar.gz文件,使用XShell工具下载到/myapp/seaweedfs路径。

解压tar.gz 文件：

tar -xzvf linux\_amd64.tar.gz 生成一个weed可执行文件，用于操作master和volume等。 创建数据卷根目录以及各个数据卷目录、master数据目录和filter目录

mkdir -p /myapp/seaweedfs/data/master mkdir -p /myapp/seaweedfs/data/vola mkdir -p /myapp/seaweedfs/data/volb mkdir -p /myapp/seaweedfs/data/volc mkdir -p /myapp/seaweedfs/data/vold mkdir -p /data/weedfs\_data/filer SeaWeedfs启动 启动SeaWeedfs可使用nohup命令进行后台运行并且将日志输出到文件中。

启动mster，-defaultReplication=“001”表示相同机架存储一份副本，而且如果一个卷服务器没在相同的rack开启一个vol就会报错:

nohup ./weed -v=3 master -mdir=/myapp/seaweedfs/data/master -defaultReplication=“001” >> /myapp/seaweedfs/data/master/wdfsmaster.log & 启动volume，两个卷服务器放在同一个机架:

nohup ./weed -v=3 volume -port=8081 -dir=/myapp/seaweedfs/data/vola -mserver=localhost:9333 -rack=rack1 >> /myapp/seaweedfs/data/vola/wefsvola.log & ​ nohup ./weed -v=3 volume -port=8082 -dir=/myapp/seaweedfs/data/volb -mserver=localhost:9333 -rack=rack2 >> /myapp/seaweedfs/data/volb/wefsvolb.log & ​ nohup ./weed -v=3 volume -port=8083 -dir=/myapp/seaweedfs/data/volc -mserver=localhost:9333 -rack=rack3 >> /myapp/seaweedfs/data/volc/wefsvolc.log & ​ nohup ./weed -v=3 volume -port=8084 -dir=/myapp/seaweedfs/data/vold -mserver=localhost:9333 -rack=rack1 >> /myapp/seaweedfs/data/vold/wefsvold.log & 开启filer服务器:

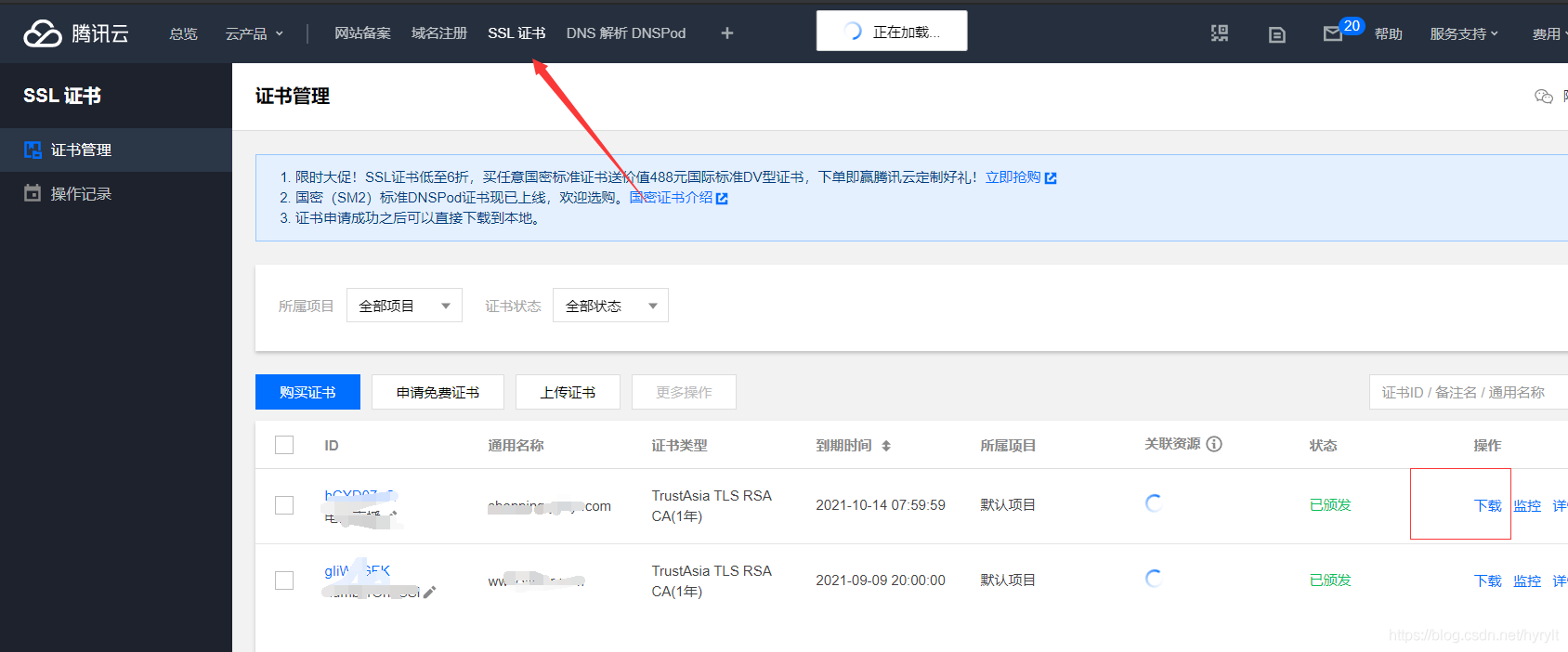
nohup ./weed -v=3 filer -port=8888 -master=localhost:9333 >> /myapp/seaweedfs/data/filer/filer.log & 开启挂载服务器：

nohup ./weed mount -filer=localhost:8888 -dir=/myapp/seaweedfs/data/filermount/data -filer.path=/ >> /myapp/seaweedfs/data/filermount/wefsmount.log &

# 七、配置域名证书

宝塔安装的nginx环境 想要手动为站点域名配置 SSL 证书操作

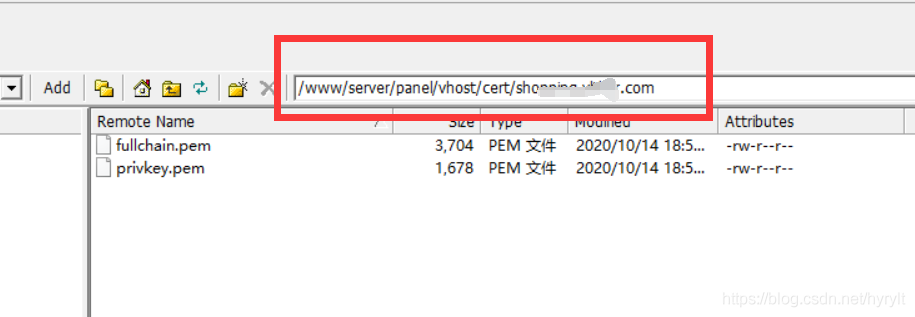
1.在购买域名的地方申请SSL证书 ,(免费版也可以)

2.

3.申请域名购买的证书只能申请的域名使用

4.下载SSL证书文件到本地

5.将SSL证书配置到服务器对应位置,此处为宝塔安装环境下的证书放置位置

6.

7.在/www/server/nginx/conf/vhost 下创建这个域名的访问配置文件

8.参考nginx中配置server写法 改成自己证书地址即可

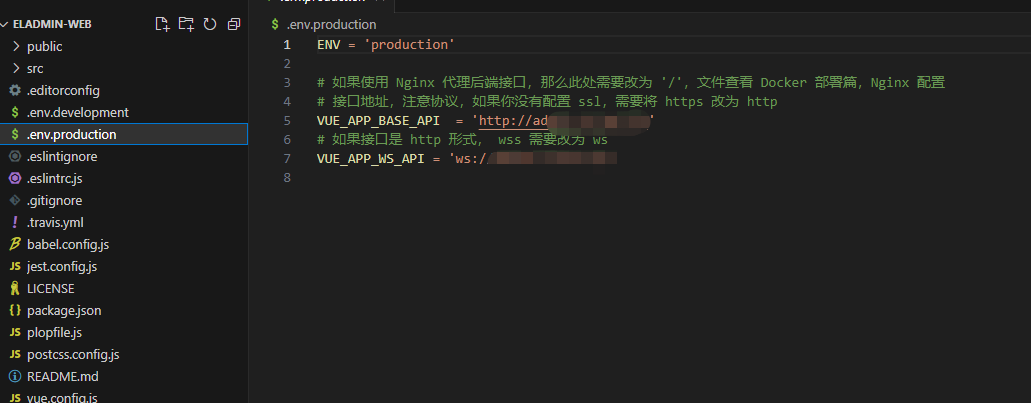
9.重启服务器

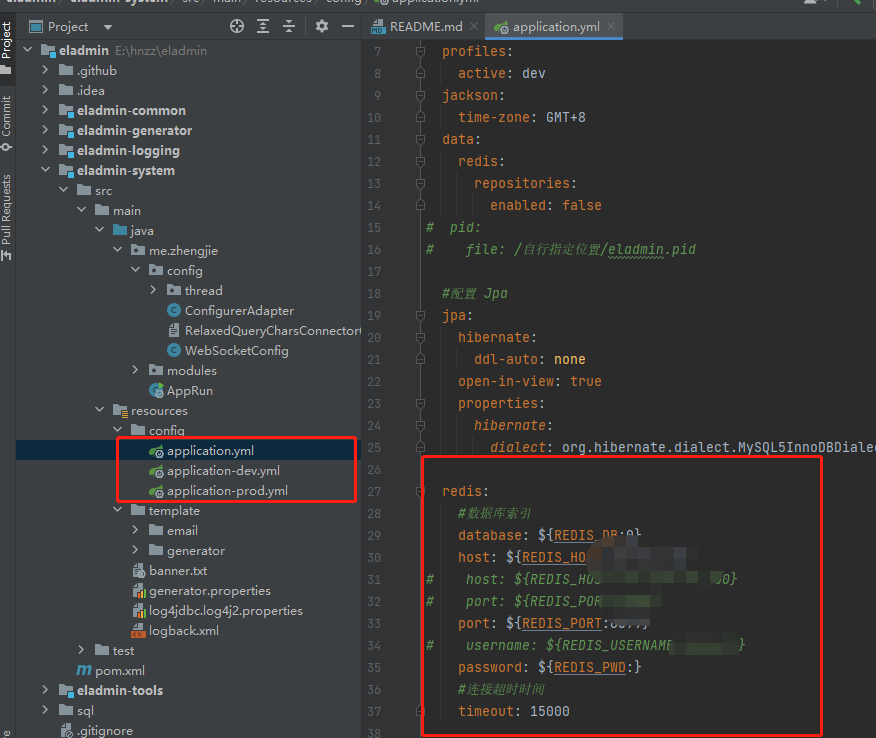
10.将域名换为https进行访问

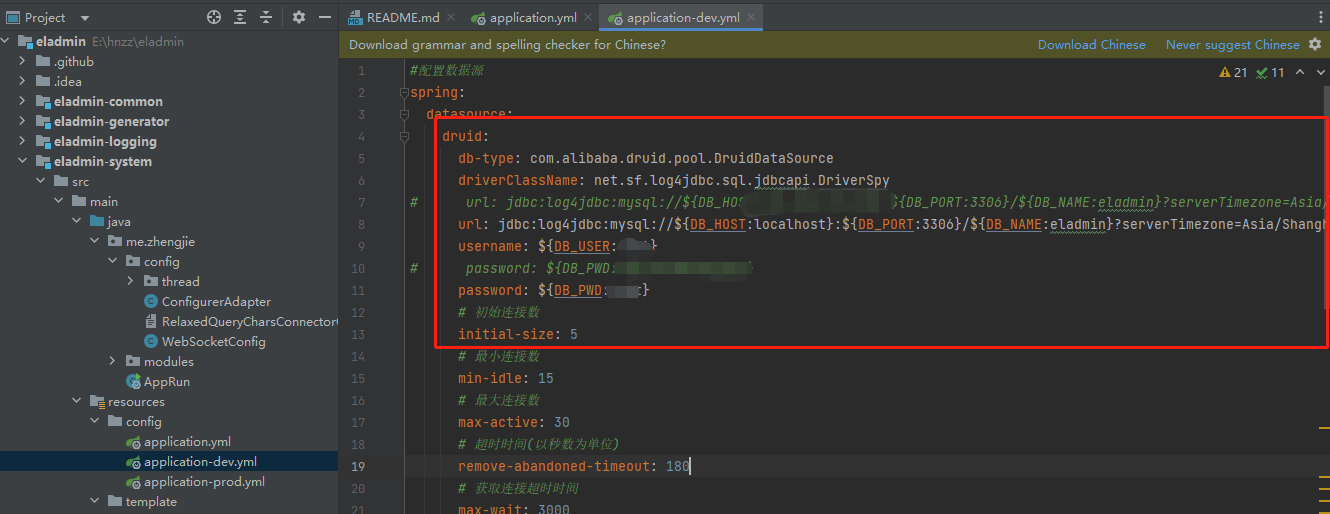
至此所有配置及软件安装全部安装完毕！！！！！！！

克隆项目

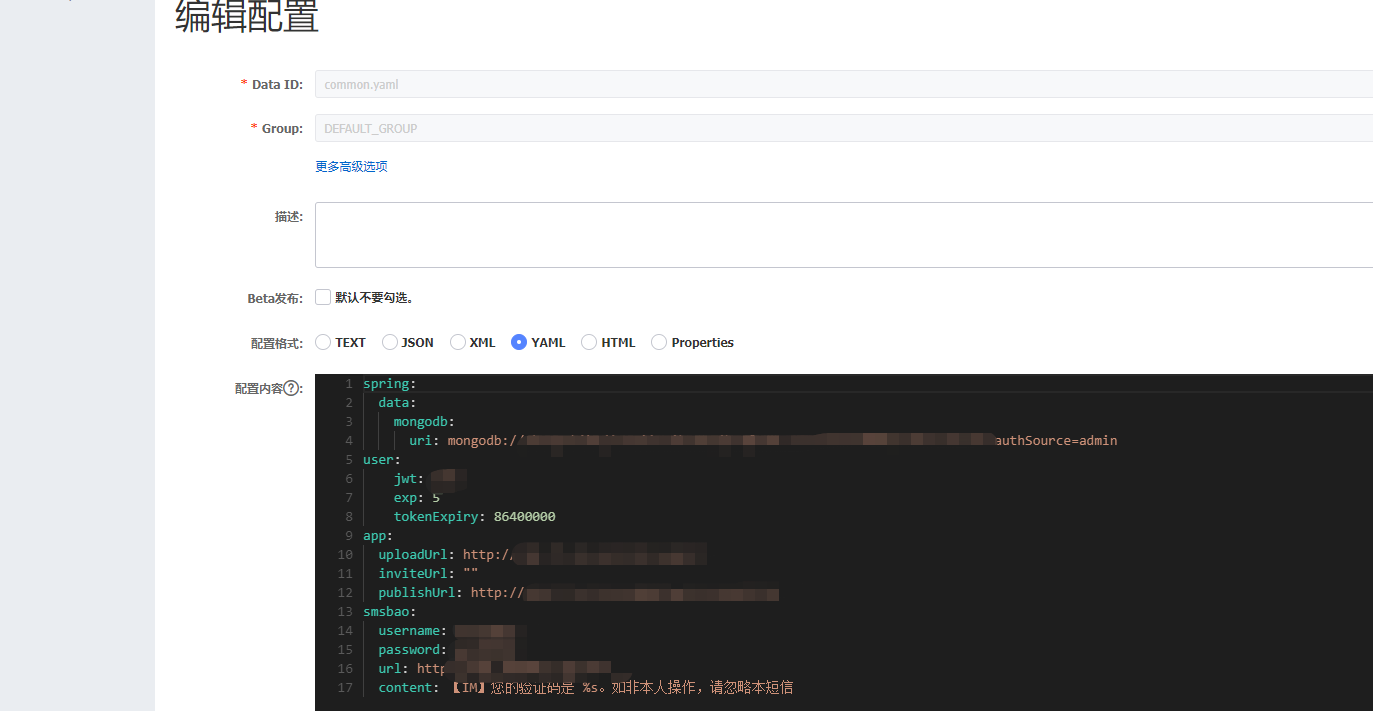
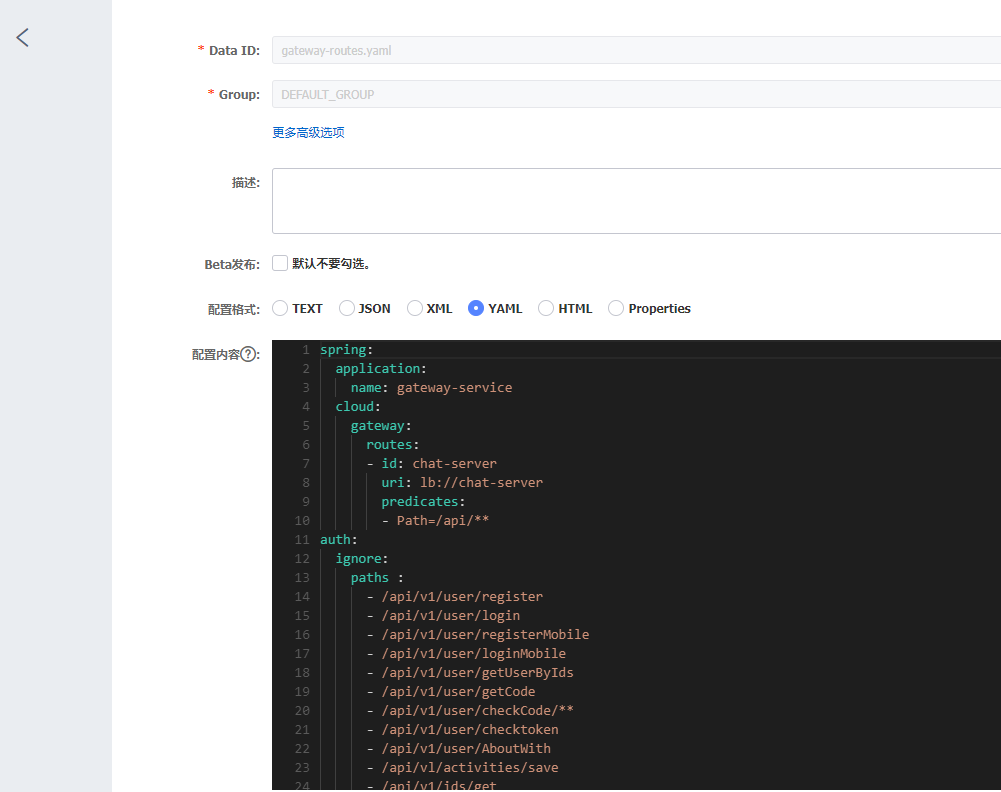
并且配置好后端mysql路径 及移动端nacos地址 mongodb地址

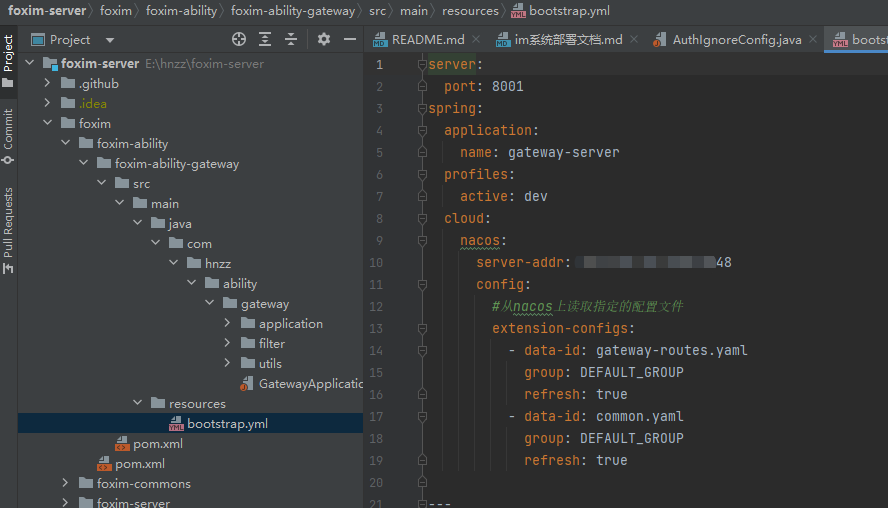
后端ui修改图示： 

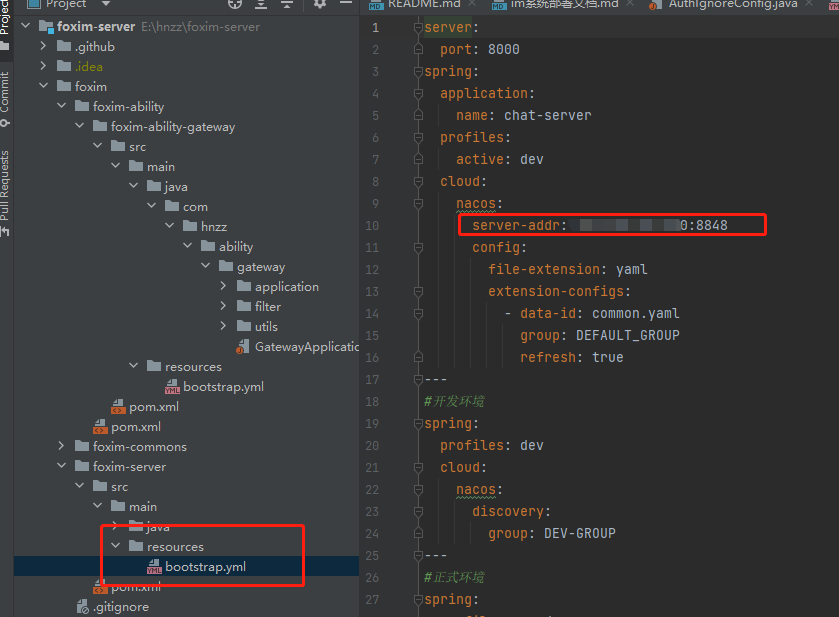
后端admin修改图示： 



img34.image

移动端nacos修改图示： common.yaml  gateway-routes.yaml 

移动端后台源码修改图示： 



img39.image

移动端client修改图示： 

后端ui 移动端client分别进行npm run build打包

java后台源码分别进行maven clean和package

上传dist前端打包文件到nginx目录html下foxim-项目对应文件夹

3.上传jar包到www/jar目录下

rz命令

4.启动命令如下：

nohup java -Xms256m -Xmx256m -Xmn128m -jar /www/jar/foxim-ability-gateway.jar > /www/jar/foxim-ability-gateway.jar\_log.txt 2>&1

nohup java -Xms256m -Xmx256m -Xmn128m -jar /www/jar/foxim-server.jar > /www/jar/foxim-server.jar\_log.txt 2>&1 &

nohup java -Xms256m -Xmx256m -Xmn128m -jar /www/jar/foxim-admin.jar > /www/jar/foxim-admin.jar\_log.txt 2>&1 &