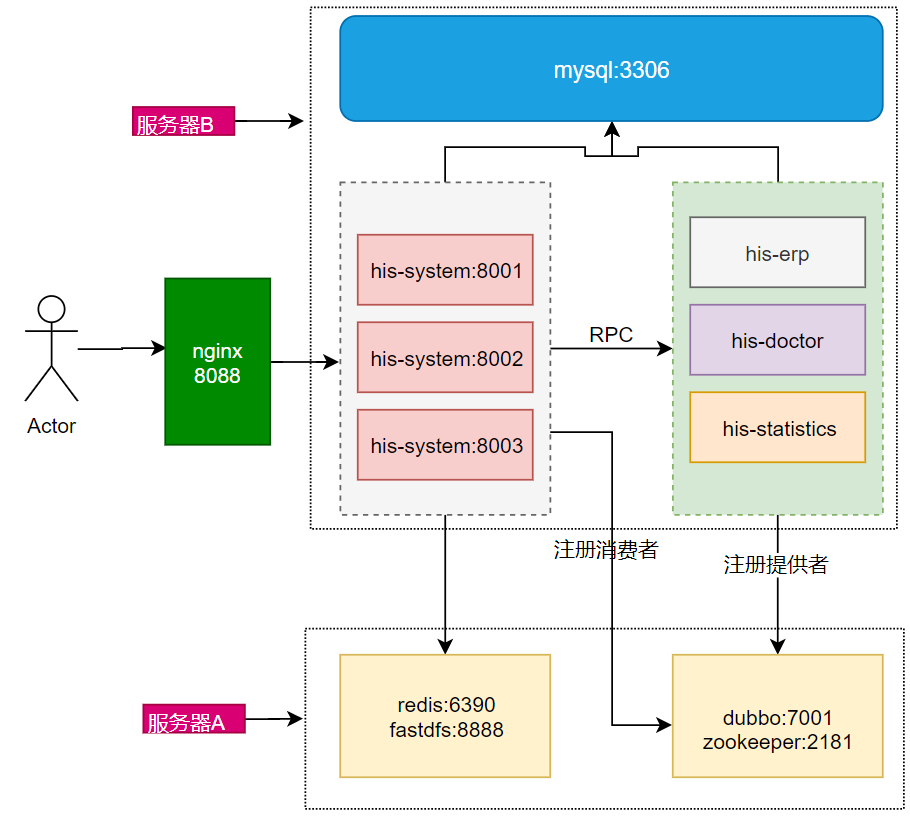
**Open-his阿里云部署开发笔记**

# 部署草图



# 准备工作

## 准备两台新【操作系统linux】的阿里云服务器

以我的为例www.leige.tech为A服务器[1核2G]

[www.leige.plus为B服务器[2核4G]](http://www.leige.plus为B服务器[2核4G])

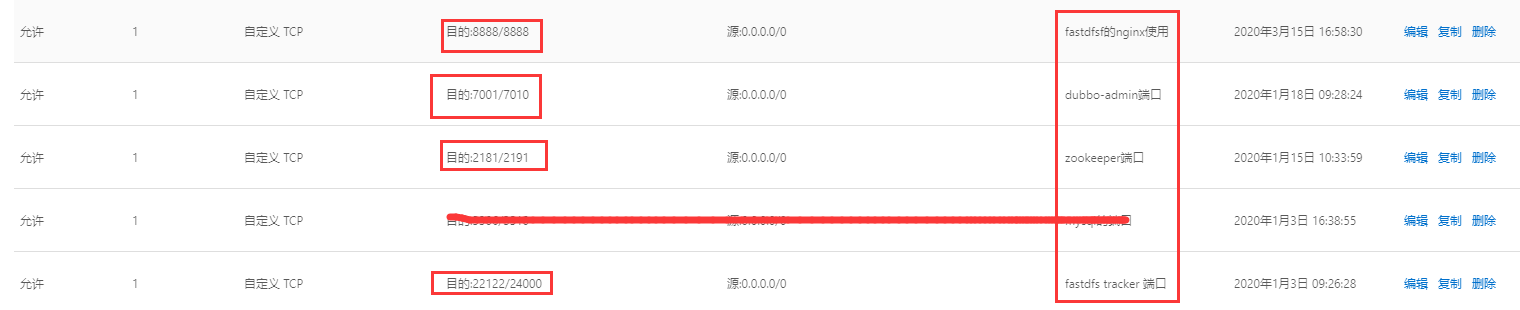
说明：如果你没有域名，就使用IP代替

要确保两个服务器之间可以互相访问

两台服务器，我这里面A服务器和**B[配置最好2H4G]**服务器代替

## 在A服务器上的配置

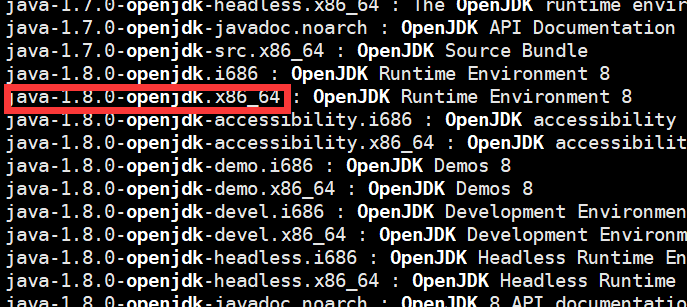
### 在A服务器上开放端口



### 在A服务器上配置JDK

#### 查找

yum search openjdk

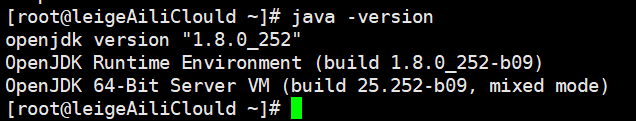


#### 安装

yum -y install java-1.8.0-openjdk

#### 测试

java -version



### 在A服务器上安装docker

yum -y install docker 安装

systemctl start docker 启动

systemctl restart docker 重启

systemctl stop docker 停止

systemctl enable docker 开机自启

**前方有坑---------如果你在docker运行期间操作了防火墙[启动或关闭]必须重启**

docker systemctl restart docker

编辑配置文件

vi /etc/docker/daemon.json

拷贝下面的内容/etc/docker/daemon.json中

{ "registry-mirrors": ["https://32xw0apq.mirror.aliyuncs.com"] }

systemctl daemon-reload

systemctl restart docker

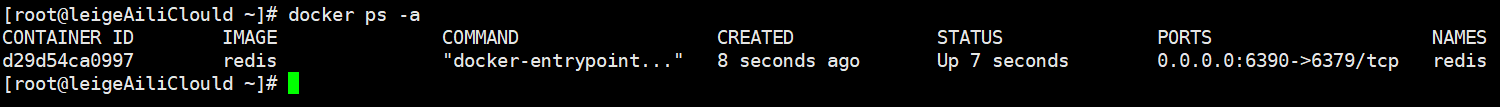
### 在A服务器上docker安装VIM

yum -y install vim

### 在A服务器上docker安装redis

使用docker 运行redis

docker run -d --name redis -p 6390:6379 redis --requirepass "123456"

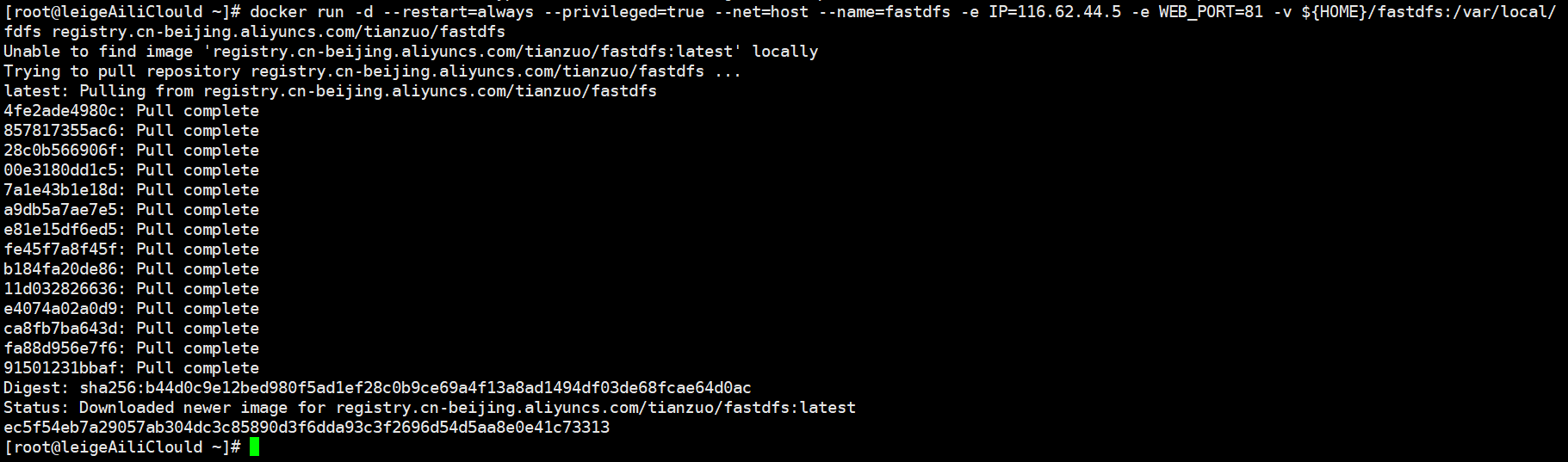


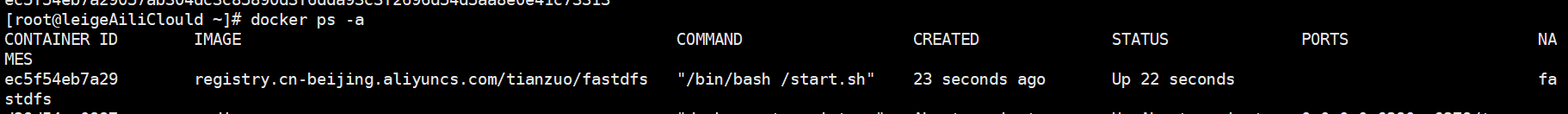
### 在A服务器上docker安装fastdfs

使用docker 运行redis

docker run -d --restart=always --privileged=true --net=host --name=fastdfs -e IP=116.62.44.5 -e WEB\_PORT=8888 -v ${HOME}/fastdfs:/var/local/fdfs registry.cn-beijing.aliyuncs.com/tianzuo/fastdfs

注意：上面的IP换成你虚拟机的IP WEB\_PORT 最好改成**非80**的端口，因为后面部署项目的nginx要使用80端口



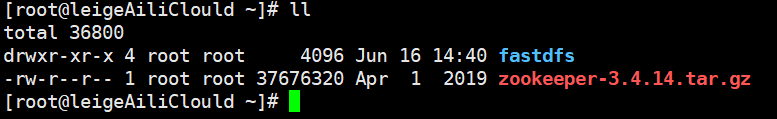


### 在A服务器上安装zookeeper

#### 下载zookeeper

cd /root

wget <https://mirror.bit.edu.cn/apache/zookeeper/zookeeper-3.4.14/zookeeper-3.4.14.tar.gz>



#### 解压

cd /root/software

tar -zxvf zookeeper-3.4.14.tar.gz

#### 重命名并移动

cd /root/software

mv zookeeper-3.4.14 zookeeper

cd /usr/local/

mv zookeeper /usr/local

#### 备份并重命名配置文件

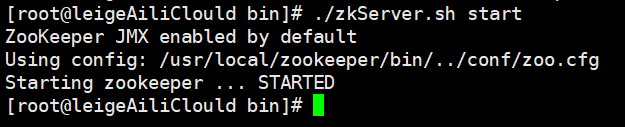
cd /usr/local/zookeeper/conf

cp zoo\_sample.cfg zoo.cfg

#### 启动zookeeper

cd /usr/local/zookeeper/bin

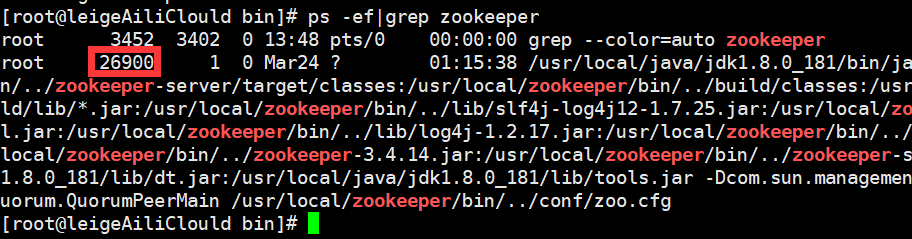
./zkServer.sh start



#### 检查是否启动成功

ps -ef|grep zookeeper

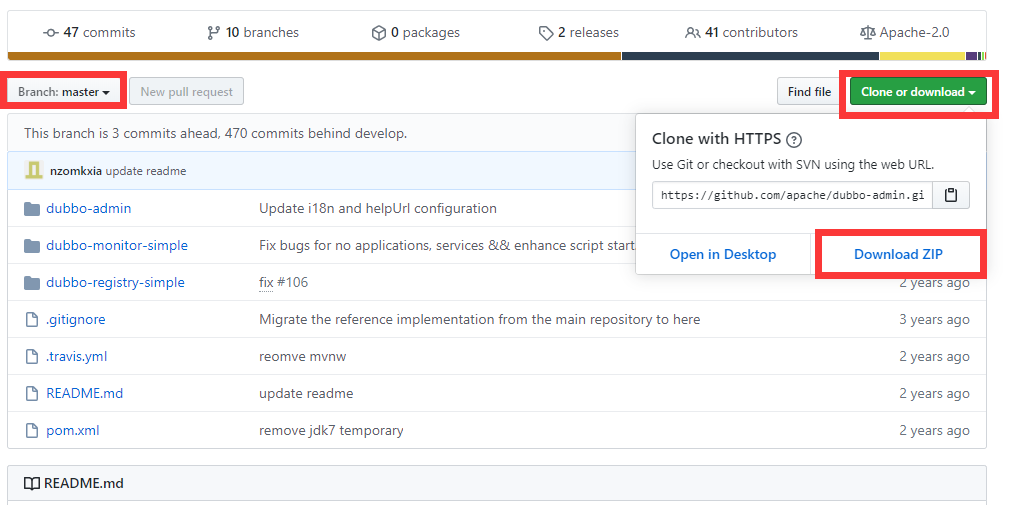
**如果下面有进程ID就是正常的**



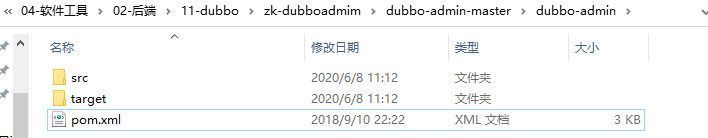
### 在A服务器上启动dubbo-admin

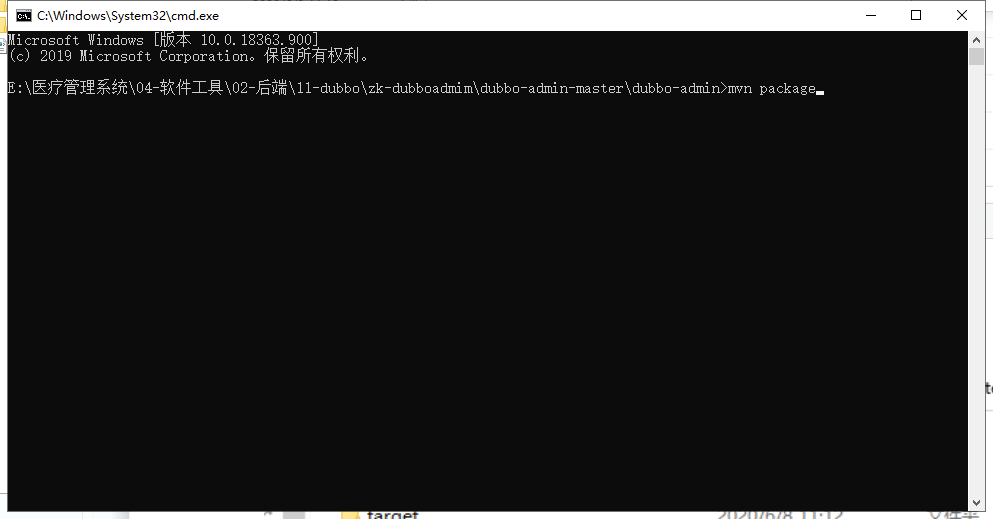
#### 下载dubbo-admin

<https://github.com/apache/dubbo-admin/tree/master>

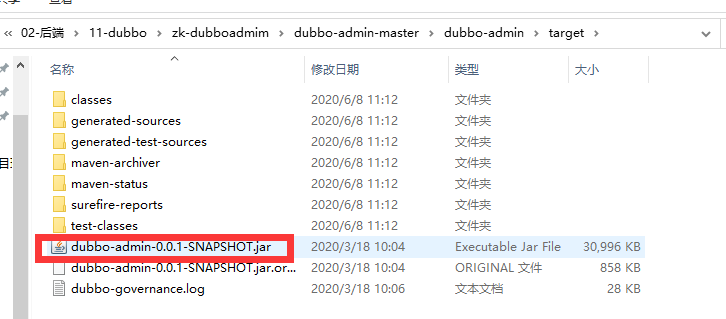


#### 打包dubbo-admin

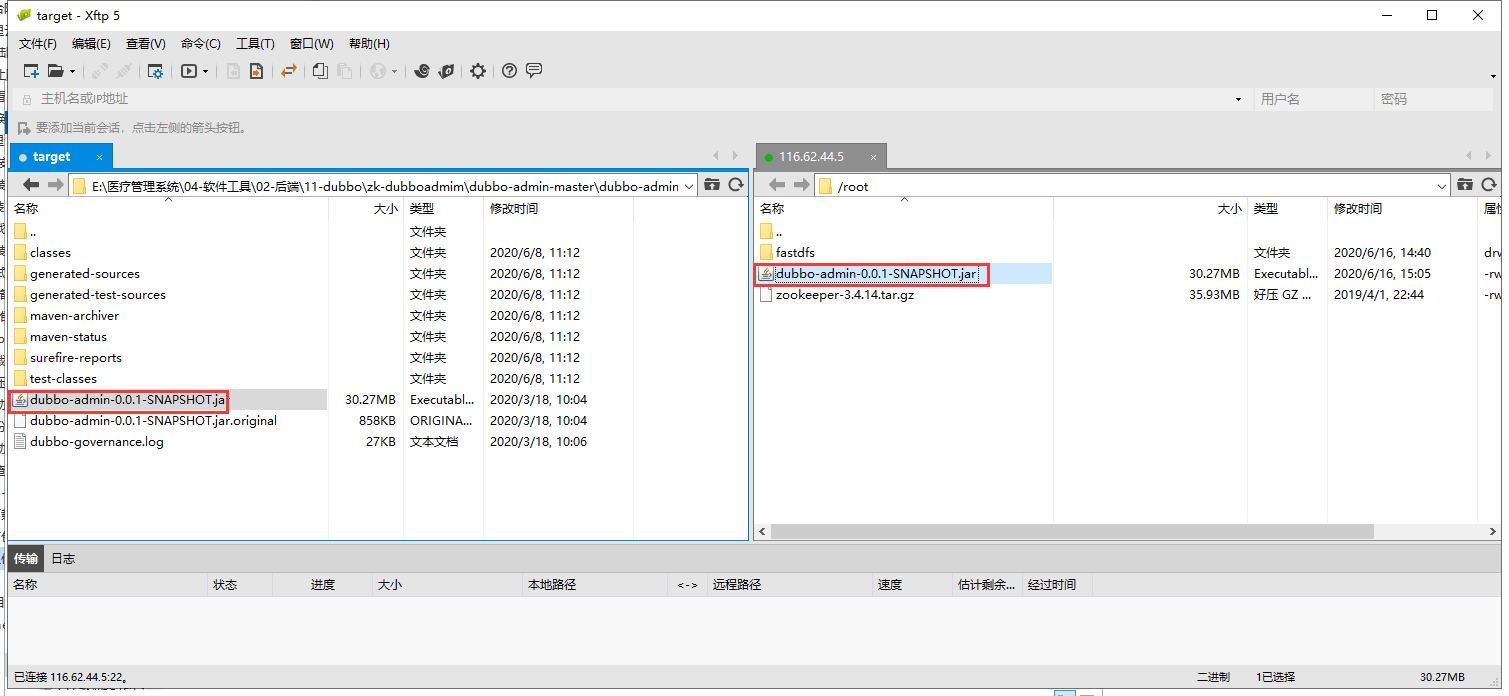




打包成功



#### 上传dubbo-admin的jar包到阿里云

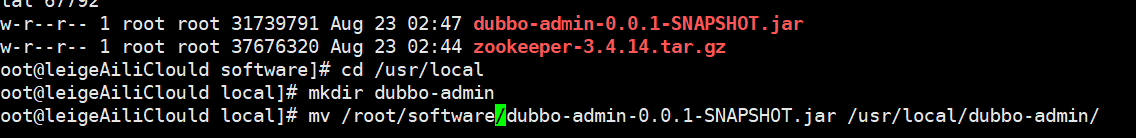


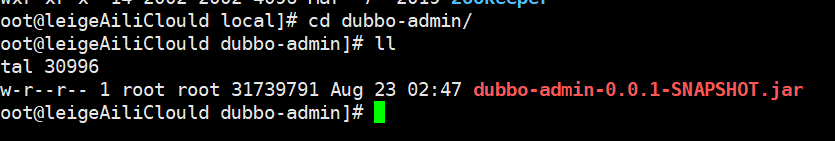
#### 移动jar包到/usr/local/dubbo-admin目录下

cd /usr/local

mkdir dubbo-admin

mv /root/software/dubbo-admin-0.0.1-SNAPSHOT.jar /usr/local/dubbo-admin/

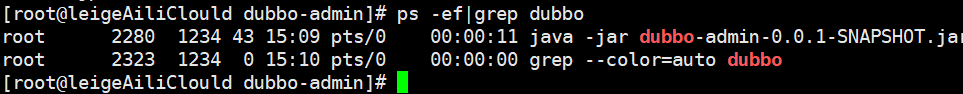




#### 后台启动dubbo-admin

下面这个命令是后台启动jar包并不记录目录

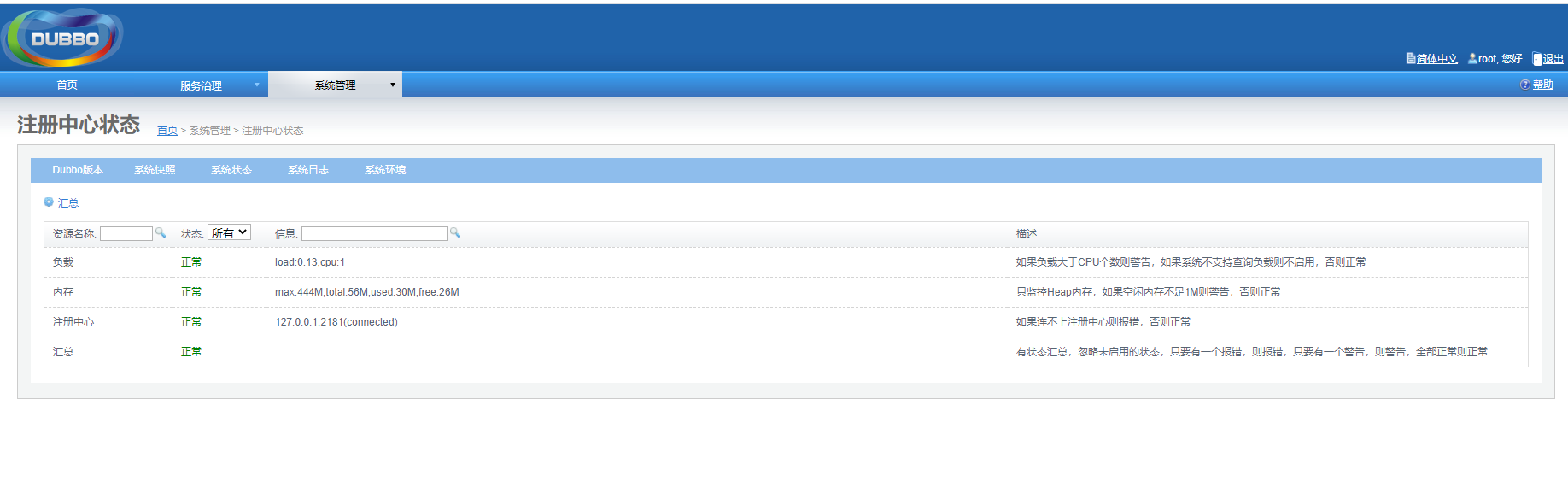
nohup java -jar dubbo-admin-0.0.1-SNAPSHOT.jar >/dev/null &



要关闭直接kill -9 进程ID

#### 访问测试

[www.leige.tech:7001](http://www.leige.tech:7001) 用户/密码-----root/root



## 在B服务器上的配置

### 在B服务器上开放端口

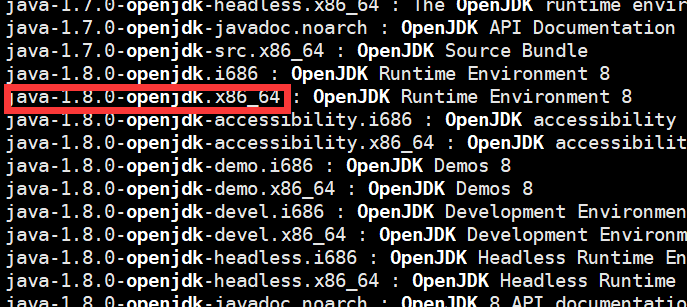


mysql的端口在搭建好之后把规则移除

### 在B服务器上配置JDK

#### 查找

yum search openjdk

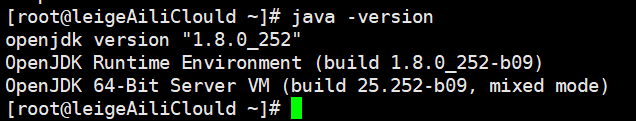


#### 安装

yum -y install java-1.8.0-openjdk

#### 测试

java -version



### 在B服务器上安装docker

yum -y install docker 安装

systemctl start docker 启动

systemctl restart docker 重启

systemctl stop docker 停止

systemctl enable docker 开机自启

**前方有坑---------如果你在docker运行期间操作了防火墙[启动或关闭]必须重启**

docker systemctl restart docker

编辑配置文件

vi /etc/docker/daemon.json

拷贝下面的内容/etc/docker/daemon.json中

{ "registry-mirrors": ["https://32xw0apq.mirror.aliyuncs.com"] }

systemctl daemon-reload

systemctl restart docker

### 在B服务器上docker安装VIM

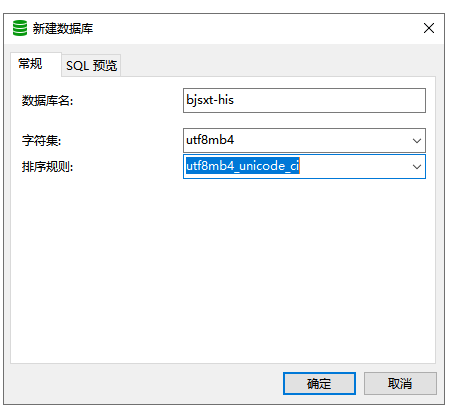
yum -y install vim

### 在B服务器上docker安装mysql5.7

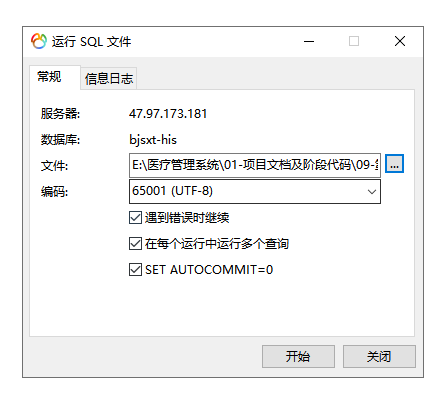
docker run --name his-mysql -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql:5.7

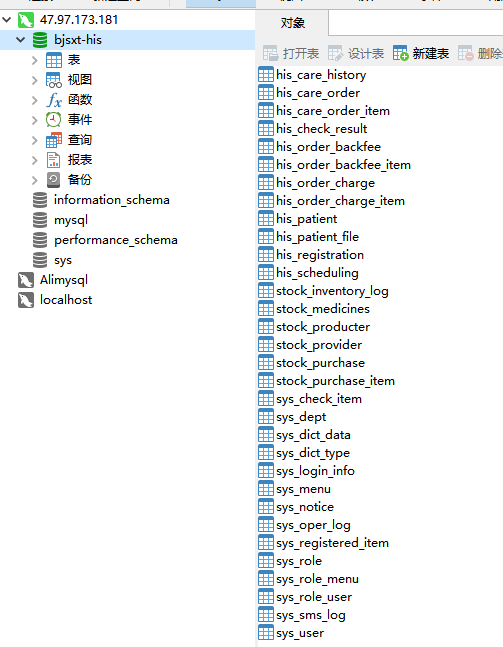
### navicat连接B服务器执行数据库脚本

#### 创建数据库



#### 运行脚本





### 在B服务器上安装nginx

#### 先安装nginx依赖的包

yum install -y gcc-c++

yum install -y pcre pcre-devel

yum install -y zlib zlib-devel

yum install -y openssl openssl-devel

#### 把nginx的源码上传到linux系统，并解压

#把解压nginx-1.14.2.tar.gz包

tar -zxvf nginx-1.14.2.tar.gz

#修改文件夹的名字

mv nginx-1.14.2 nginx

#把nginx拷贝到/usr/local/src里面

cp -r nginx /usr/local/src

#### 安装

cd /usr/local/src/nginx

#在/usr下创建nginx目录

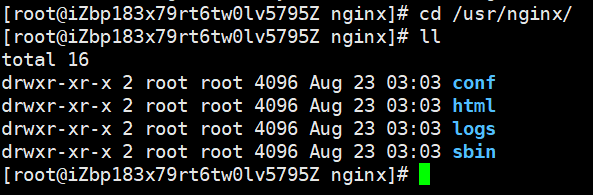
mkdir /usr/nginx

./configure --prefix=/usr/nginx (指定安装目录编译)

make

make install

#### 安装成功后的目录



#### 启动nginx

进入nginx的sbin目录，./nginx就可以启动

关闭nginx可以使用kill命令，但是不推荐使用。推荐使用：./nginx -s stop

刷新配置重启：./nginx -s reload

#### 访问测试

[www.leige.plus](http://www.leige.plus:8088)

47.97.173.181

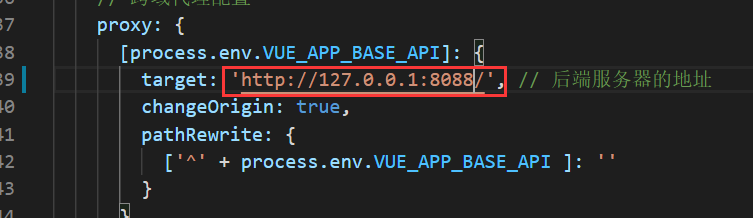


### 在B服务器上安装unzip[后面解压使用]

yum -y install unzip

# 打包前端项目

## 修改代理地址vue.config.js



这里就是nginx的地址

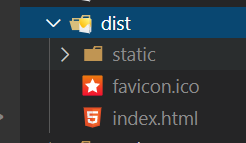
## 打包



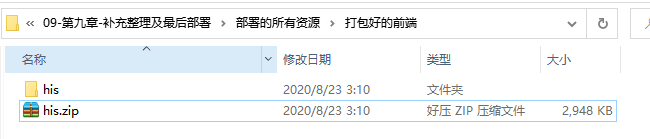
npm run build:prod

## 查看dist目录

这个目录里面就是打包好的最后的静态文件

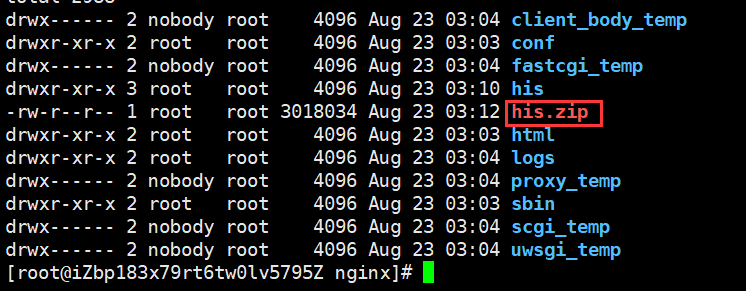


把整个dist目录复制出来改名并压缩



## 上传打包好的代码到nginx的目录并解压

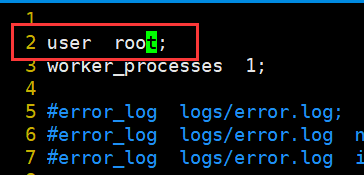
### 上传

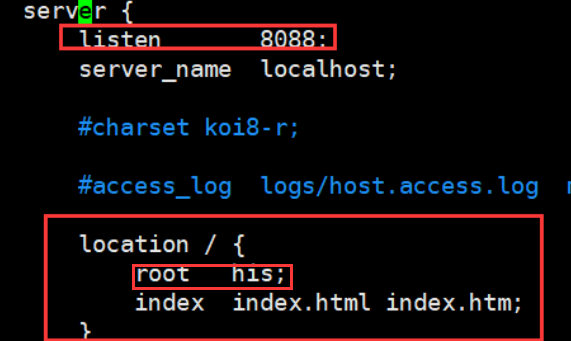


### 解压

unzip his.zip

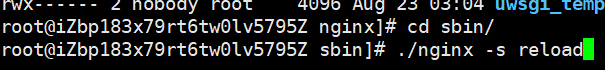
## 修改配置文件



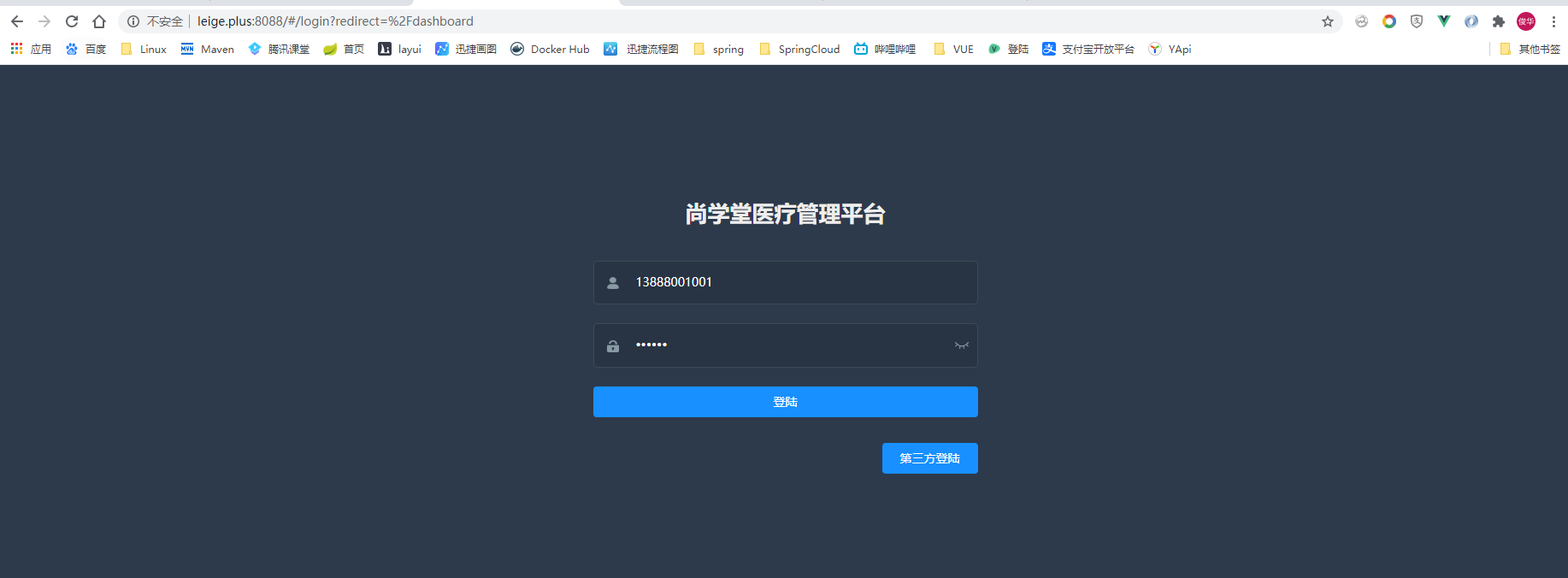


## 刷新配置并访问测试

进入nginx的sbin目录执行./nginx -s reload



www.leige.plus:8088



# 打包后端项目

## 修改打包代码pom.xml

说明，如果不加打包插件，那么第三方的包不会被打到最后的包里面去

### 【system-web】pom.xml

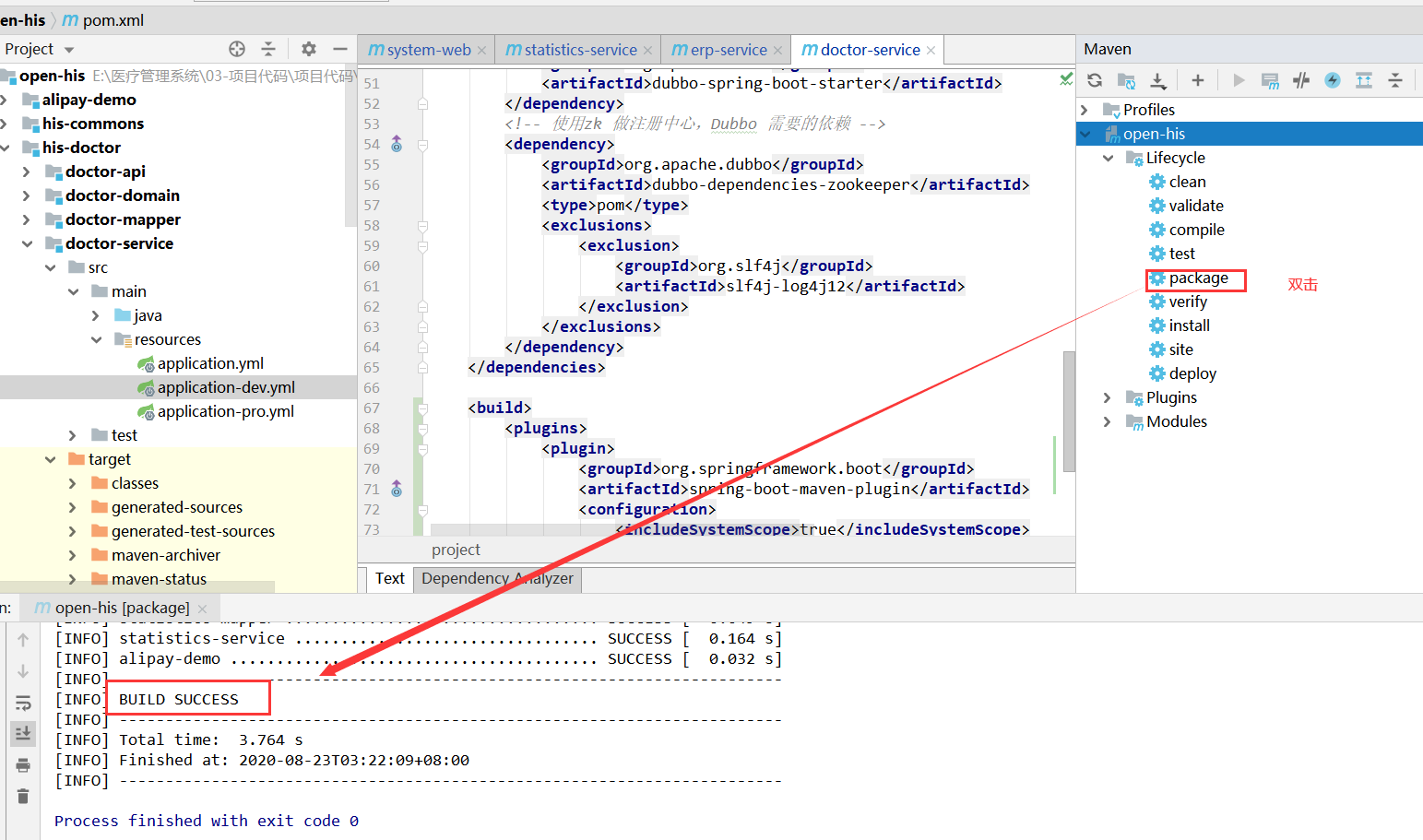
### 【erp-service】pom.xml

### 【doctor-service】pom.xml

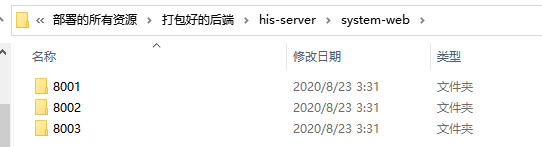
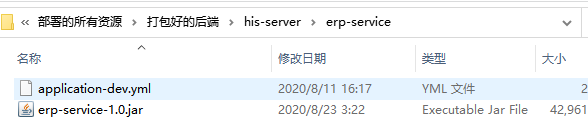
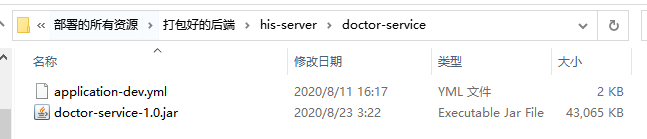
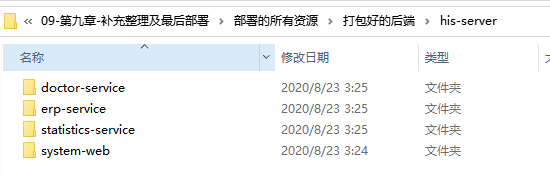
### 【statistics-service】pom.xml

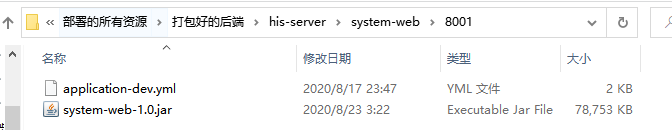
|  |
| --- |
| <**build**>  <**plugins**>  <**plugin**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-maven-plugin</**artifactId**>  <**configuration**>  <**includeSystemScope**>true</**includeSystemScope**>  </**configuration**>  </**plugin**>  </**plugins**> </**build**> |

## 整体打包

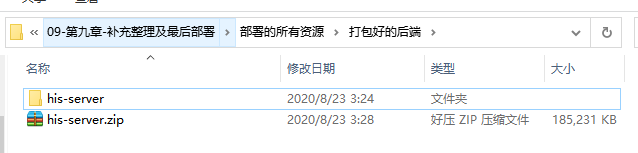


## 复制jar包并整理



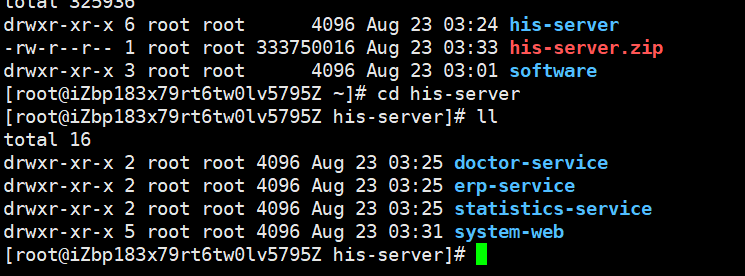


## 把整理好的jar包目录打包上传到linux的/root目录



## 在linux下面解压

unzip his-server.zip



# 启动三个system-web【负载】

## 修改/root/his-server/system-web/8001/application-dev.yaml



## 修改/root/his-server/system-web/8002/application-dev.yaml



## 修改/root/his-server/system-web/8003/application-dev.yaml



## 启动8001下面的jar

cd /root/his-server/system-web/8001

nohup java -jar system-web-1.0.jar >/dev/null &

## 启动8002下面的jar

cd /root/his-server/system-web/8002

nohup java -jar system-web-1.0.jar >/dev/null &

## 启动8003下面的jar

cd /root/his-server/system-web/8003

nohup java -jar system-web-1.0.jar >/dev/null &

# 分别启动其它的子系统【多个也行】

## doctor-service

cd /root/his-server/doctor-service/

nohup java -jar doctor-service-1.0.jar >/dev/null &

## erp-service

cd /root/his-server/erp-service/

nohup java -jar erp-service-1.0.jar >/dev/null &

## statistics-service

cd /root/his-server/statistics-service/

nohup java -jar statistics-service-1.0.jar >/dev/null &

# 查看dubbo-admin



# 配置nginx/conf/nginx.conf

cd /usr/nginx/conf

vim nginx.conf

upstream his-server {

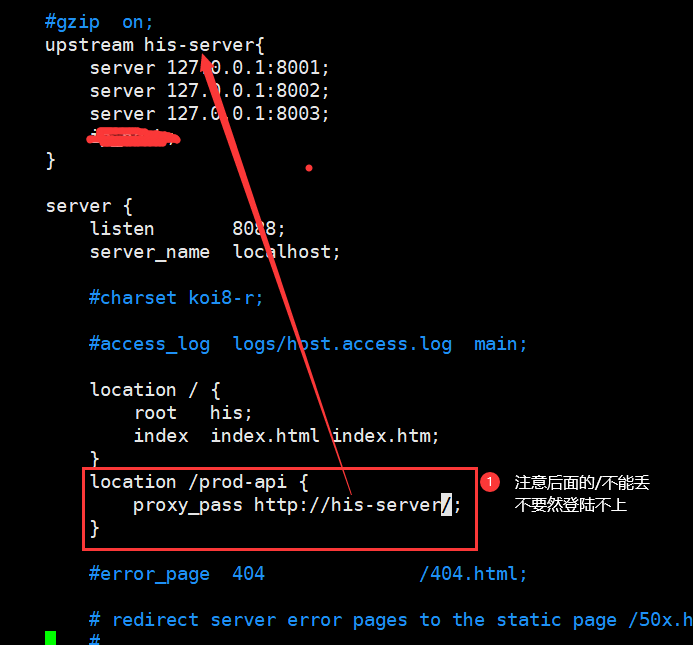
server 127.0.0.1:8001;

server 127.0.0.1:8002;

server 127.0.0.1:8003;

}

proxy\_pass http://his-server; #指向上面配置的server的节点



有坑

proxy\_pass 路径后面的/不能丢

# 刷新配置访问nginx测试

cd /usr/nginx/sbin

./nginx -s reload

# 访问nginx测试

<http://www.leige.plus:8088/>