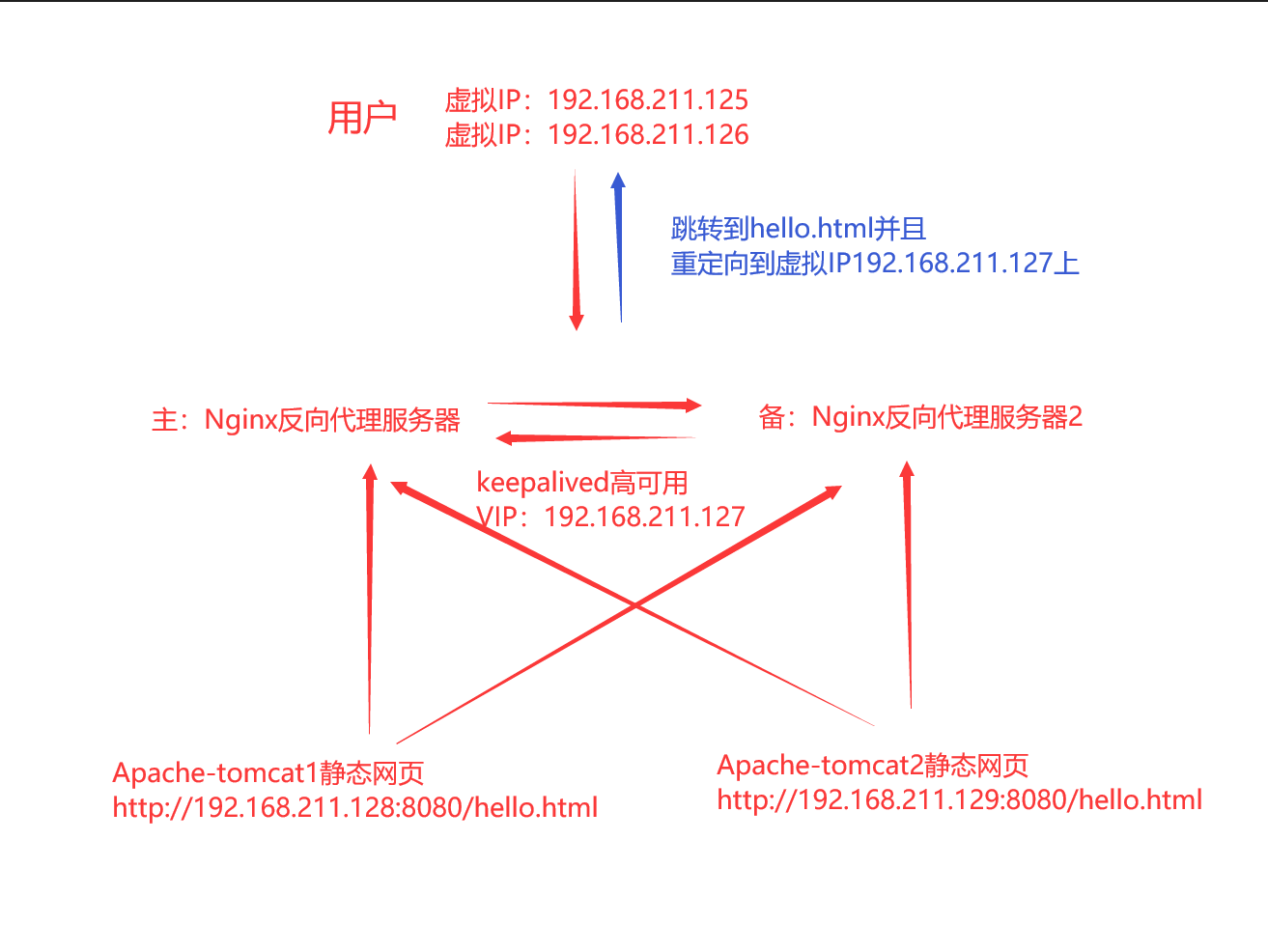
**nginx1.25.1+tomcat10.1.13**

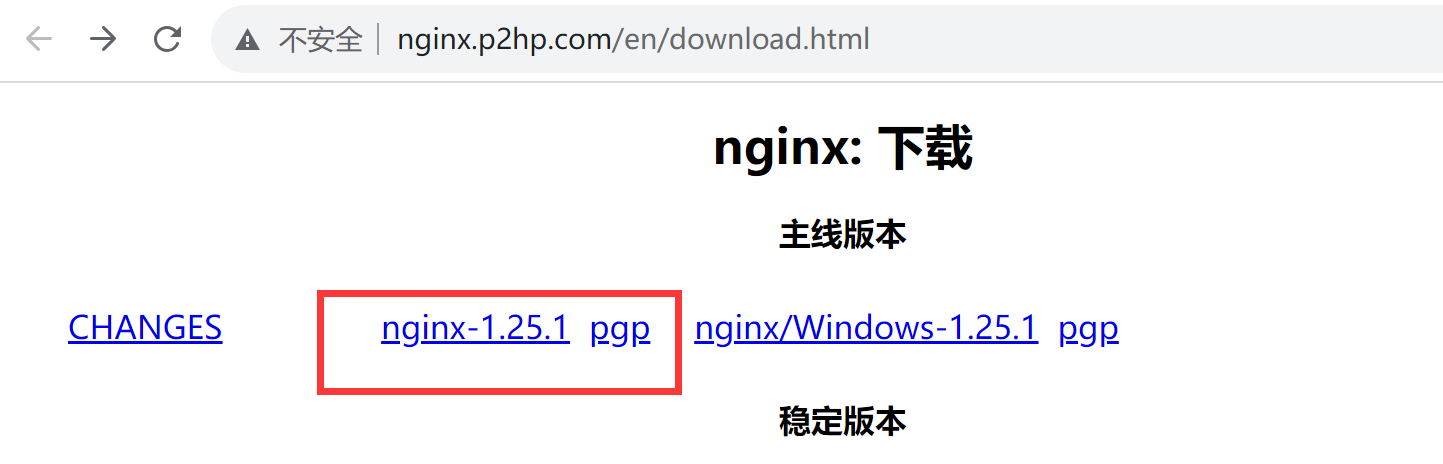
**1、使用tomcat分别在两台虚拟机中部署静态网页，通过虚拟机Ip:8080访问网页时显示该虚拟机的主机名**

**2、使用nginx在两台虚拟机中配置1中tomcat静态网页的反向代理(proxy\_pass)，并使用keepalived实现高可用**

**3、使用rewrite实现多个servername的重定向，统一重定向到ip最后一个数字是127的虚拟ip上**



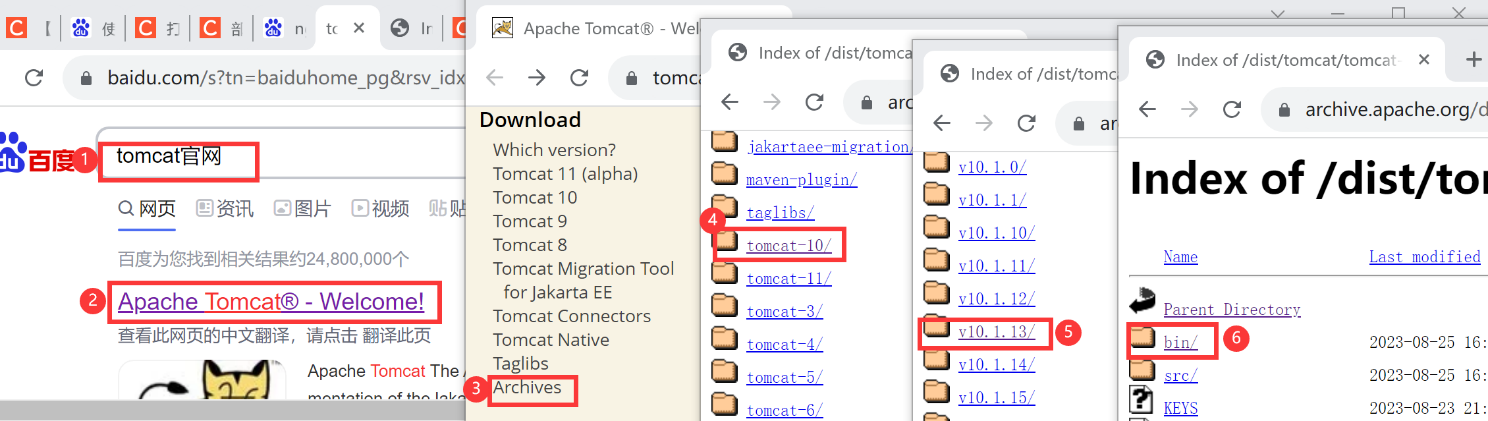
1. **下载nginx1.25.1和tomcat10.1.13安装包以及jdk安装包**
   1. **nginx1.25.1下载网址：**[**http://nginx.p2hp.com/en/download.html**](http://nginx.p2hp.com/en/download.html)

****

**1.2 Tomcat10.1.13官网及下载方法参考**

官网：<https://tomcat.apache.org/>

下载参考：<https://blog.csdn.net/weixin_56800842/article/details/125261227>





**1.3 Jdk安装包下载（jdk版本需要与tomcat版本对应）**

**1.3.1查看要下载的版本**

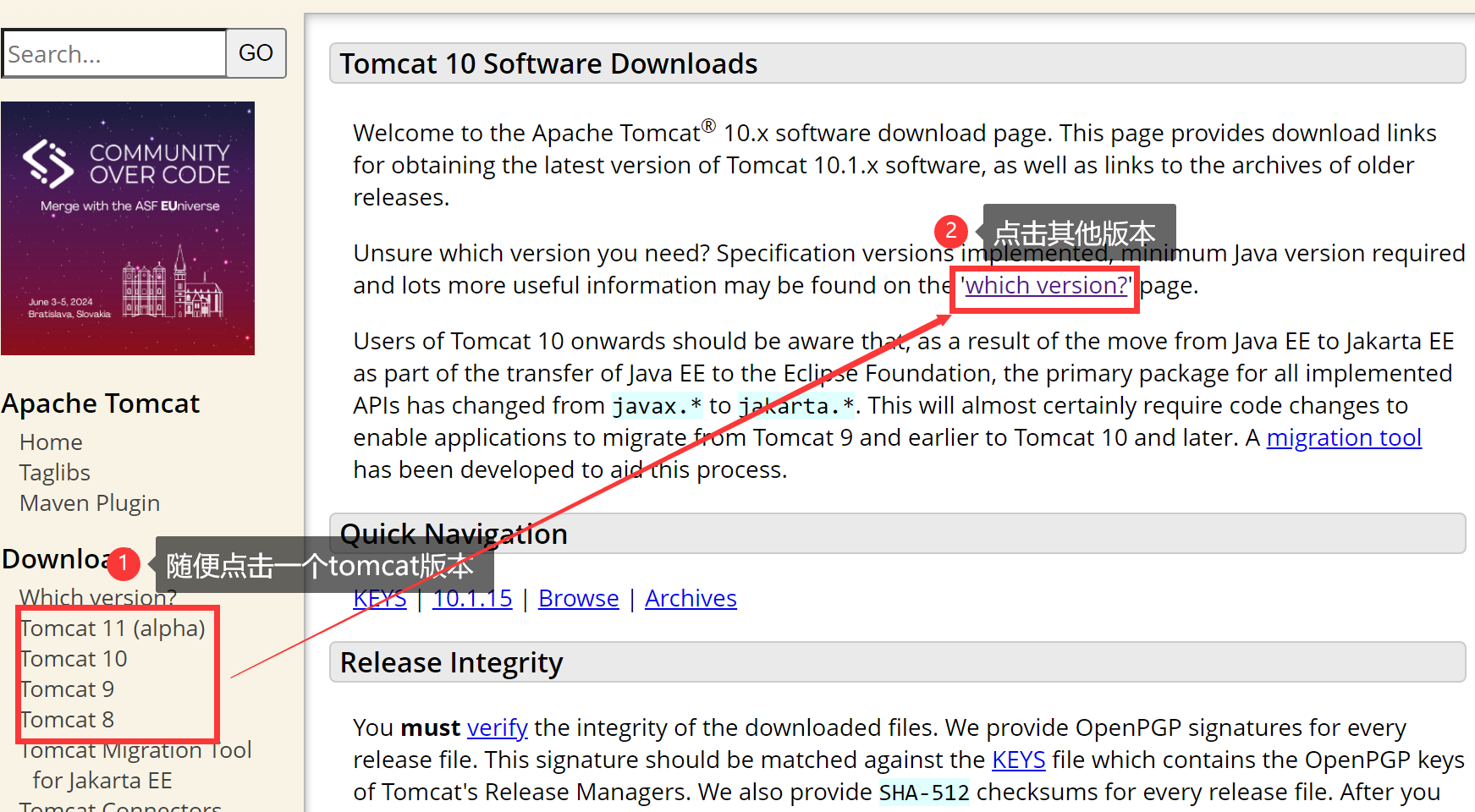
****

**扩展：**

**查看tomcat与jdk对应版本关系方法**

（参考网址：<https://blog.csdn.net/SANXINGS8/article/details/132596314> **）**

1. **登录tomcat网站：**<https://tomcat.apache.org/download-10.cgi>
2. **如图操作**

****

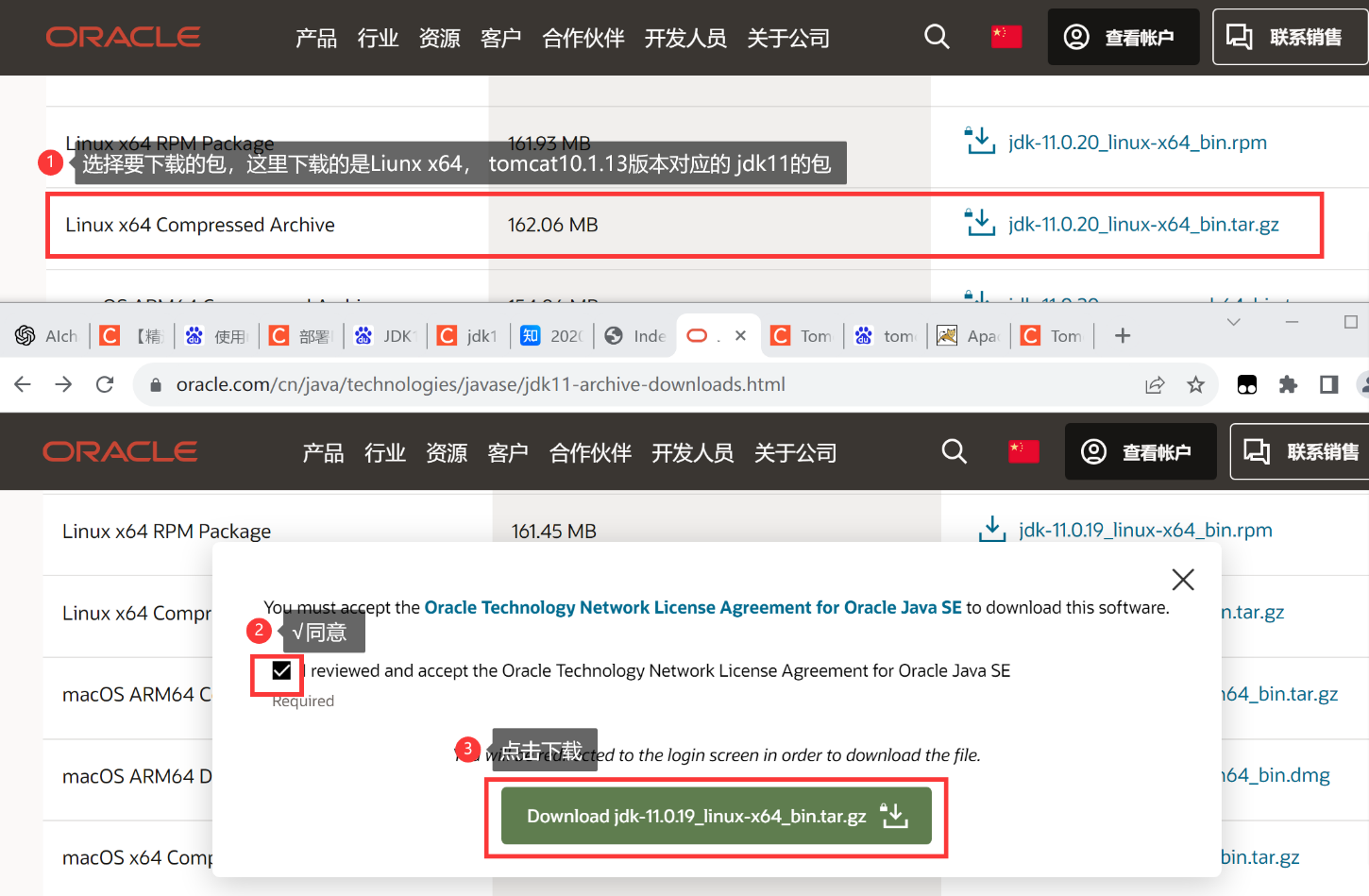
1. **得到结果**

****

**1.3.2下载jdk安装包**

**Jdk下载网址：** <https://www.oracle.com/in/java/technologies/javase/jdk11-archive-downloads.html>

**（下滑找到图中位置）**



**官网下载jdk现在比较麻烦，需要注册登录账号后才能下载，想要下载官网上的注册账号登录即可，且官网速度较慢，可以使用华为的镜像站**：

[**https://repo.huaweicloud.com/java/jdk/**](https://repo.huaweicloud.com/java/jdk/)

**下载想要的版本即可**

镜像站下载图解：

****

**官网注册页面：**

****

**Jdk下载参考网站：**

<https://zhuanlan.zhihu.com/p/184994150?eqid=ada6d54b000061d7000000046481b76c&utm_id=0>

<https://blog.csdn.net/weixin_47626759/article/details/129750023>

1. **两台虚拟机上安装tomcat**

**--安装tomcat之前需要先安装JDK**

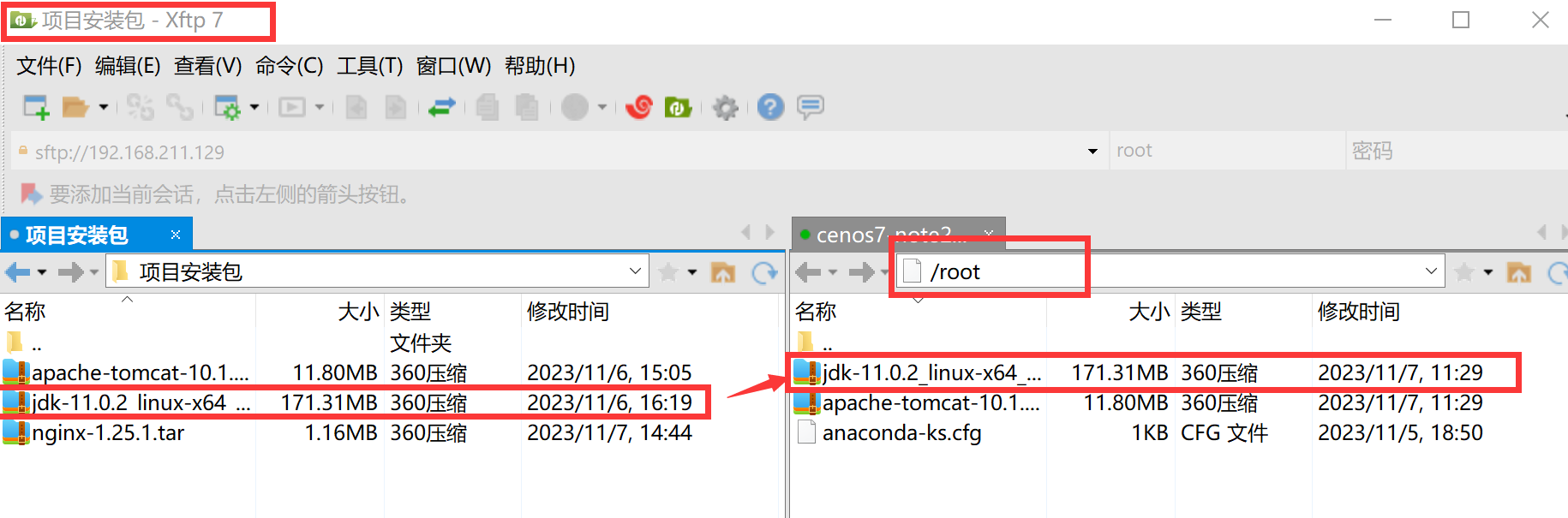
**1.在安装之前先检查该系统是否自带了jdk，检测安装包：**

# rpm -qa | grep java

**2.若有进行删除：**

# rpm -e --nodeps +包名

**3.文档里下载压缩包（版本：jdk-11.0.2\_linux-x64\_bin.tar.gz）到 /root**



**4.压缩包解压，将解压后的包复制到/usr/local/jdk目录下。**

无jdk目录的话建一个：

# mkdir /usr/local/jdk

方法一

进入目录：# cd /usr/local/

解压：# tar -zxvf jdk-11.0.2\_linux-x64\_bin.tar.gz

方法二：

直接在/root目录下输入 ：

# tar -zxvf jdk-8u211-linux-x64.tar.gz -C /usr/local/jdk

**5.接下来进入到/etc/profile配置环境变量:**

# vim /etc/profile

添加：

export JAVA\_HOME=/usr/local/jdk/jdk-11.0.2

export PATH=$JAVA\_HOME/bin:$PATH

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

按ESC键，：wq! 保存退出

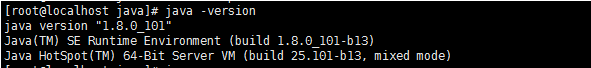


**6.重新加载配置文件**

# source /etc/profile

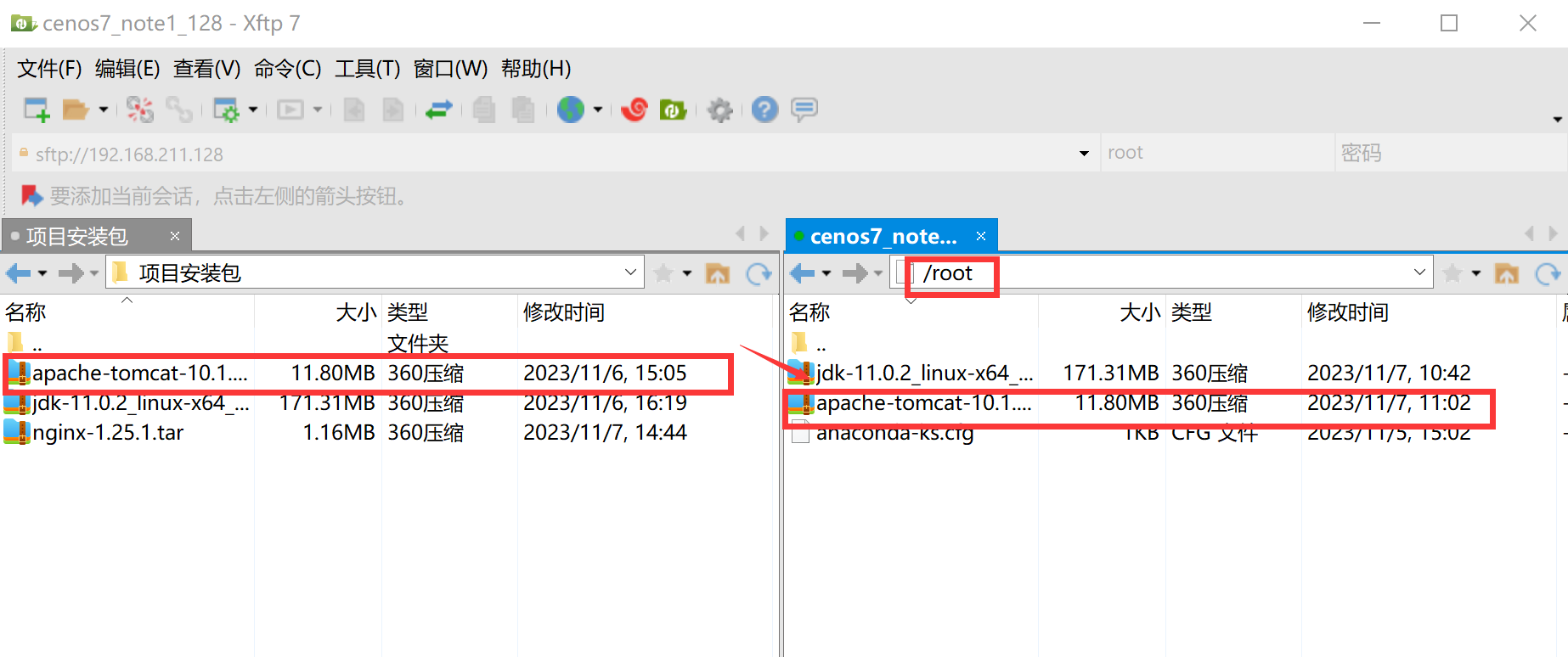
**7.测试一下jdk的安装情况**

# java –version

****

要是达到上图效果，即证明jdk安装配置成功了。

**8.文档里下载tomcat压缩包到 /root**



**9.--创建tomcat目录**

# mkdir /usr/local/tomcat

**10. --将tar包解压到/usr/local/tomcat目录里**

# tar -zxvf apache-tomcat-10.1.13.tar.gz -C /usr/local/tomcat

**11. --启动tomcat服**

# cd /usr/local/tomcat/apache-tomcat-10.1.13.tar.gz/bin

# ./startup.sh

**12.--查看tomcat服务进程**

# ps -aux|grep tomcat

root 95733 39.7 5.7 2979796 106956 pts/1 Sl 16:36 0:03 /usr/local/jdk1.8.0\_211/bin/java -Djava.util.logging.config.file=/usr/local/tomcat/apache-tomcat-9.0.69/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath /usr/local/tomcat/apache-tomcat-9.0.69/bin/bootstrap.jar:/usr/local/tomcat/apache-tomcat-9.0.69/bin/tomcat-juli.jar -Dcatalina.base=/usr/local/tomcat/apache-tomcat-9.0.69 -Dcatalina.home=/usr/local/tomcat/apache-tomcat-9.0.69 -Djava.io.tmpdir=/usr/local/tomcat/apache-tomcat-9.0.69/temp org.apache.catalina.startup.Bootstrap start

(显示以上这段表示启动成功)

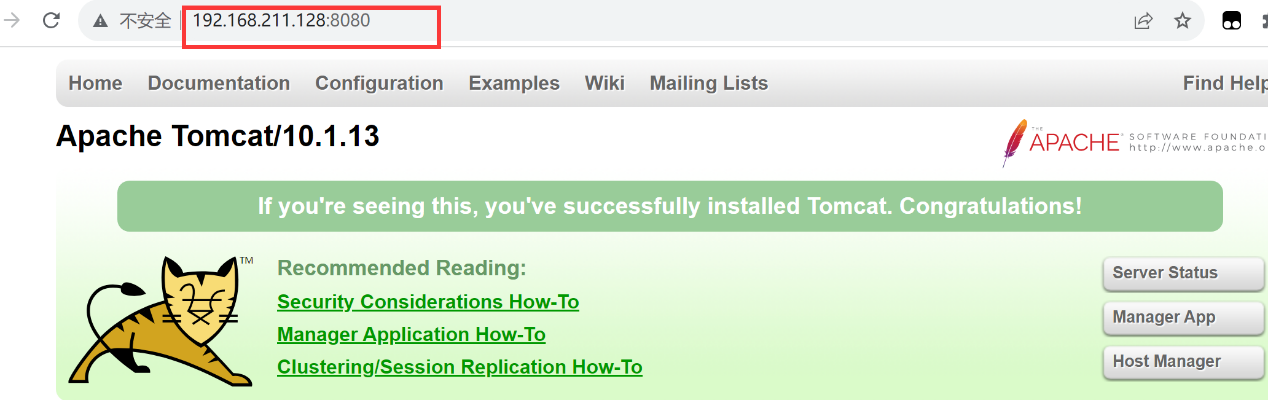
**13.--测试服务器是否可以到达tomcat控制台**

在网页上：http://192.168.211.128:8080/

或

在虚拟机中输入命令：# curl http://192.168.211.128:8080/

如图即为成功

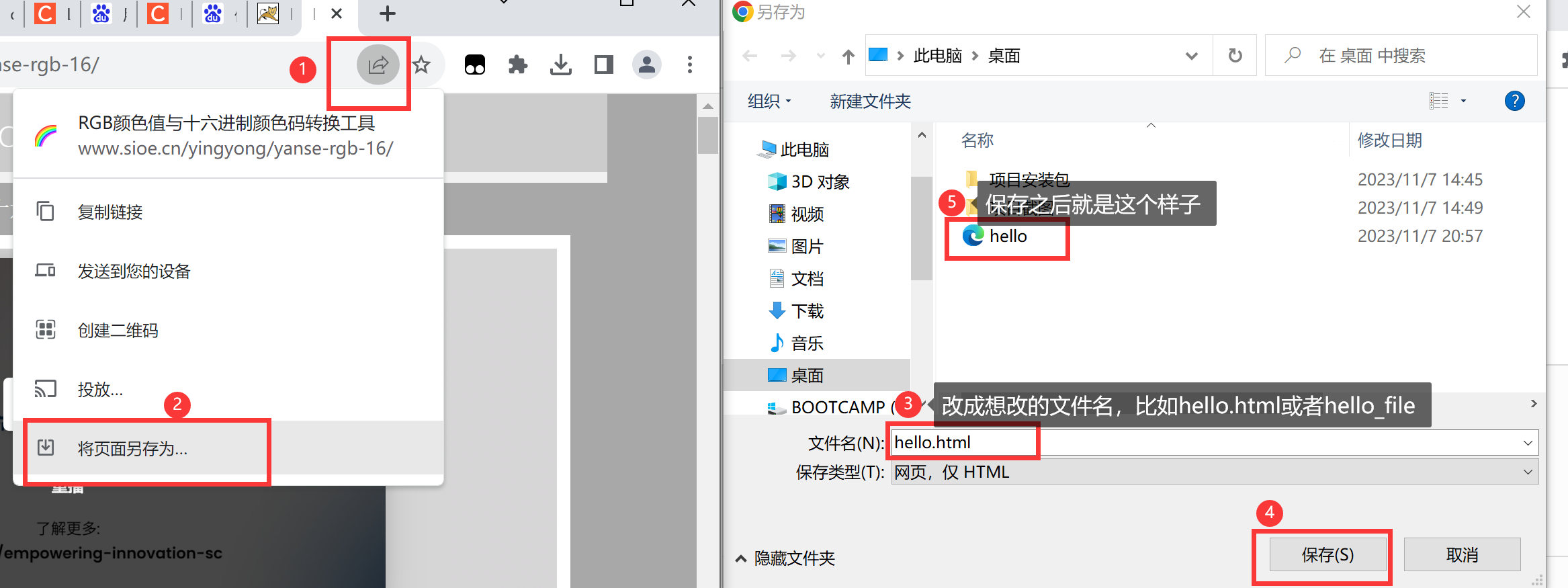


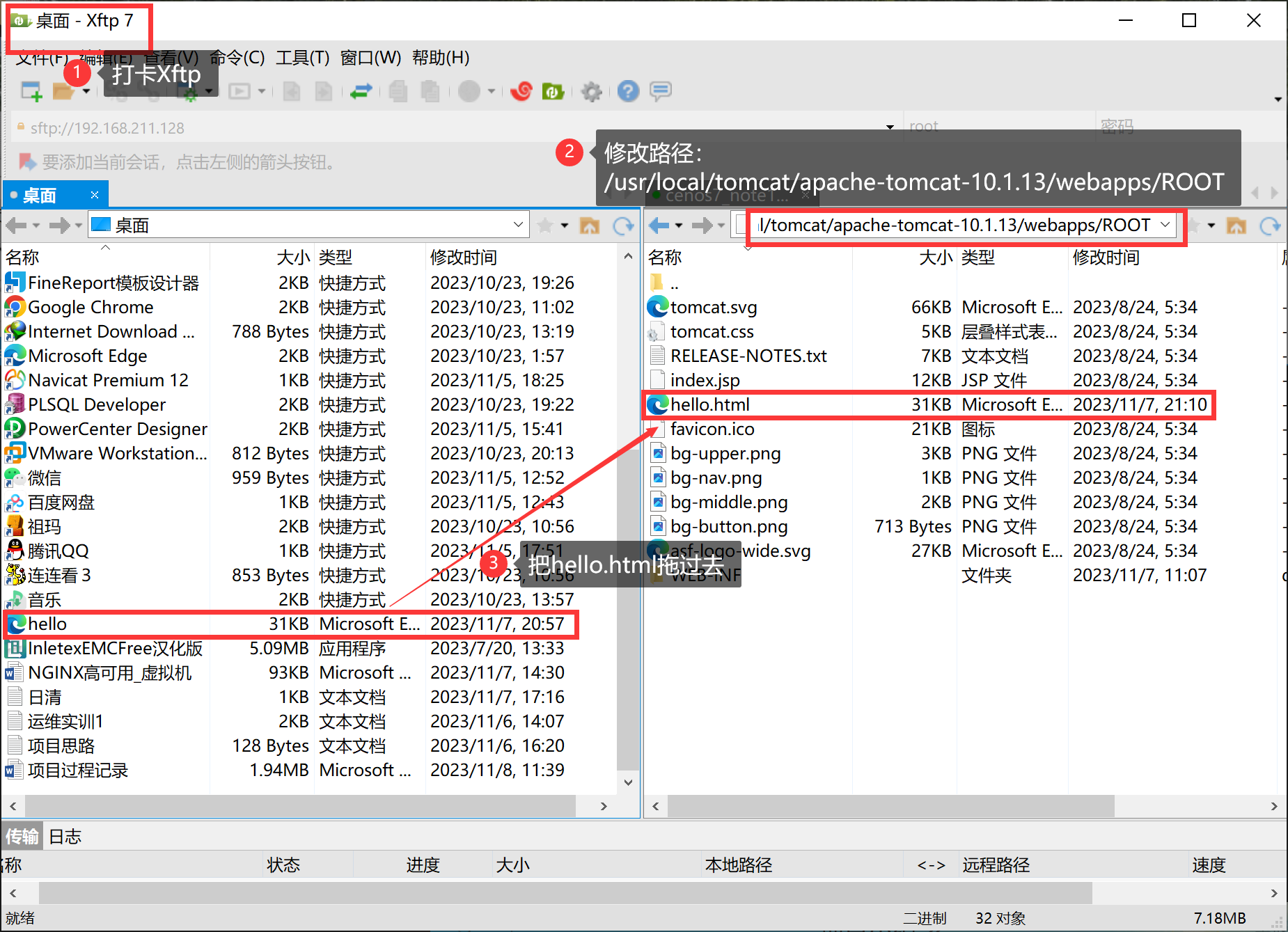
1. **两台虚拟机配置tomcat静态网页**

**3.1下载网站，用Xftp放入此路径：**

# /usr/local/tomcat/apache-tomcat-10.1.13/webapps/ROOT

（什么网站都可以，此处用的是<https://www.sioe.cn/yingyong/yanse-rgb-16/>）





**3.2测试：打开网页输入 IP : 端口 / HTML文件名 (符号为英文)**

**出来的是下载的网站，则为成功（如图）**



**可能出现的问题：**

**若是出来404或乱码，可能是网站下载的问题，可以打开下载好的网站确认一下是否是之前选定下载的网站。**

1. **在两台虚拟机上安装nginx**

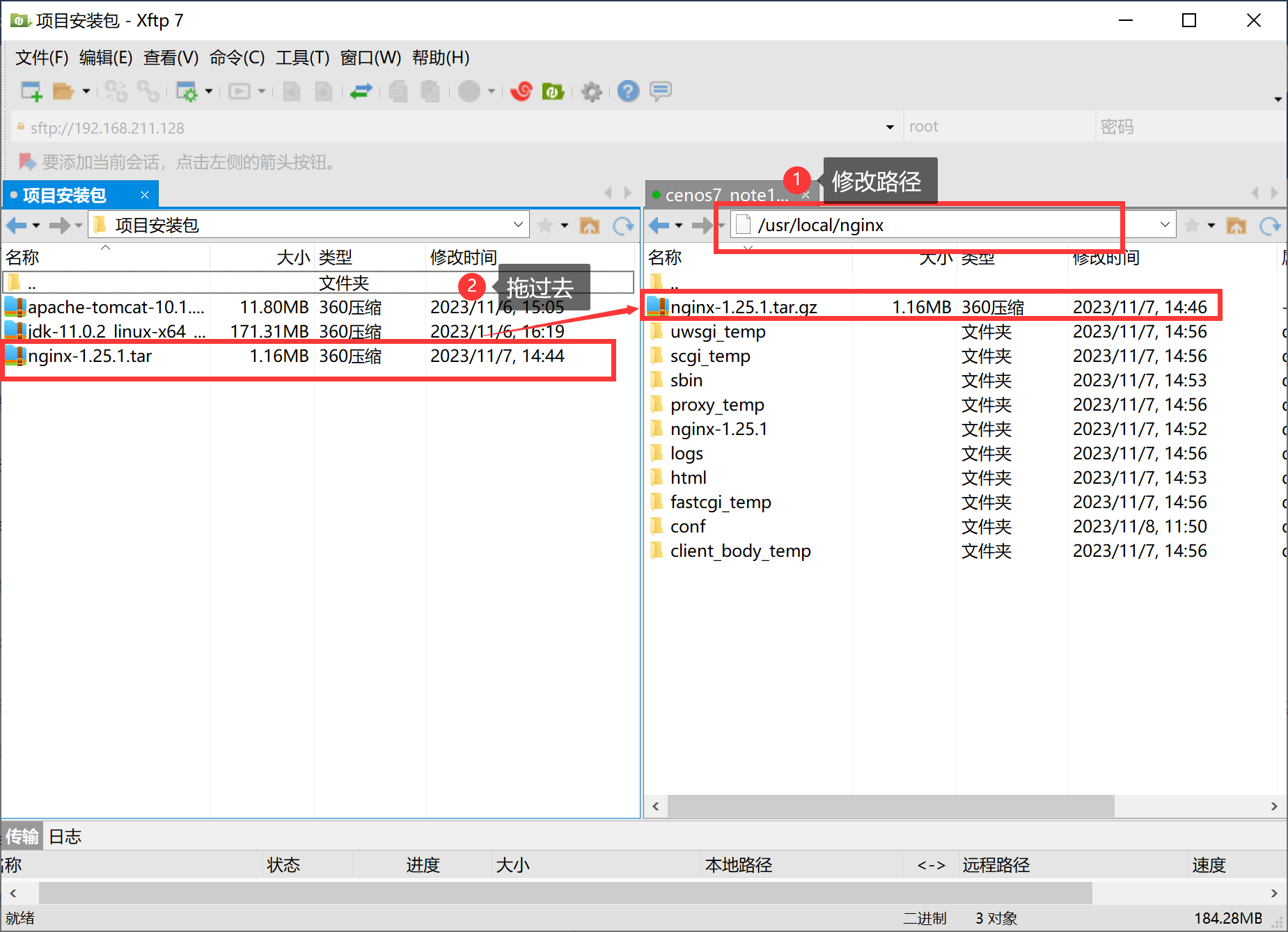
**4.1 安装编译安装所需要的依赖**

# yum install -y gcc-c++ pcre pcre-devel zlib zlib-devel openssl openssl-devel

**4.2创建一个nginx目录**

# mkdir /usr/local/nginx

**4.3下载安装包**



**4.4 解压**

# cd /usr/local/nginx/

# tar -zxvf nginx-1.25.1.tar.gz

**4.5进入解压后的安装包**

# cd nginx-1.25.1

**4.6执行命令**

# ./configure --with-openssl=/path/to/openssl/source）

# make && make install

**4.7启动服务**

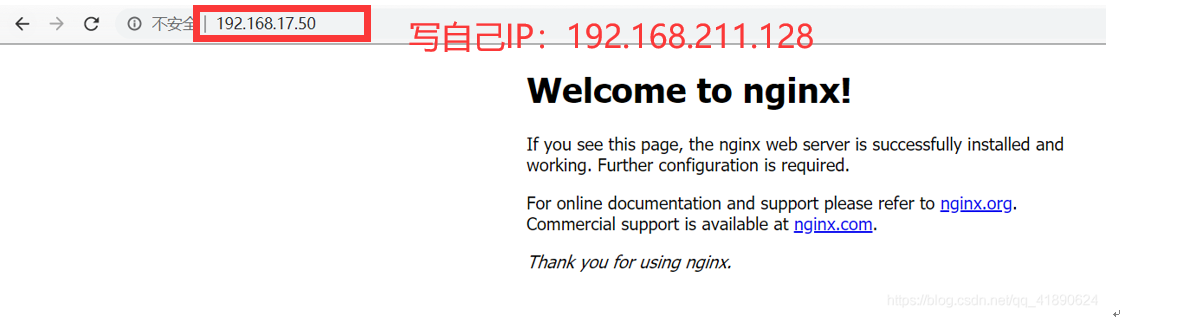
# /usr/local/nginx/sbin/nginx

**查看进程**

# ps –aux | grep nginx



**4.8测试：打开浏览器访问此机器的IP，如果浏览器出现Welcome to nginx!则表示Nginx已经安装并运行成功。**



**4.9 扩展：**

NGINX 的常用命令

重启：# /usr/local/nginx/sbin/nginx 启动命令

重启：# /usr/local/nginx/sbin/nginx –s reload

停止：# /usr/local/nginx/sbin/nginx –s stop

测试配置文件是否正常：# /usr/local/nginx/sbin/nginx –t

强制关闭：# pkill nginx

**5.nginx反向代理tomcat，两台都按以下步骤配置**

**5.1修改配置文件**

# vim /usr/local/nginx/conf/nginx.conf

修改内容：

（1）将localhost改为本机的IP

（2）注释掉图中标示的两行，并且添加以下内容：

proxy\_pass http://192.168.211.128:8080/hello.html;

proxy\_set\_header Host $host;

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

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

（proxy­\_pass +http:// +ip + :端口 +/tomcat配置的静态网页，其他三行直接粘贴）

解释：

proxy\_pass http://192.168.211.128:8080/hello.html;

告诉Nginx将收到的请求转发到指定的后端服务器192.168.211.128服务器的8080端口，并且请求的URI为/hello.html。

proxy\_set\_header Host $host;

设置了向后端服务器发送请求时的Host头部信息，通常用于告诉后端服务器请求的实际主机名。

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

设置向后端服务器发送请求时的X-Real-IP头部信息，用来传递客户端真实的IP地址。

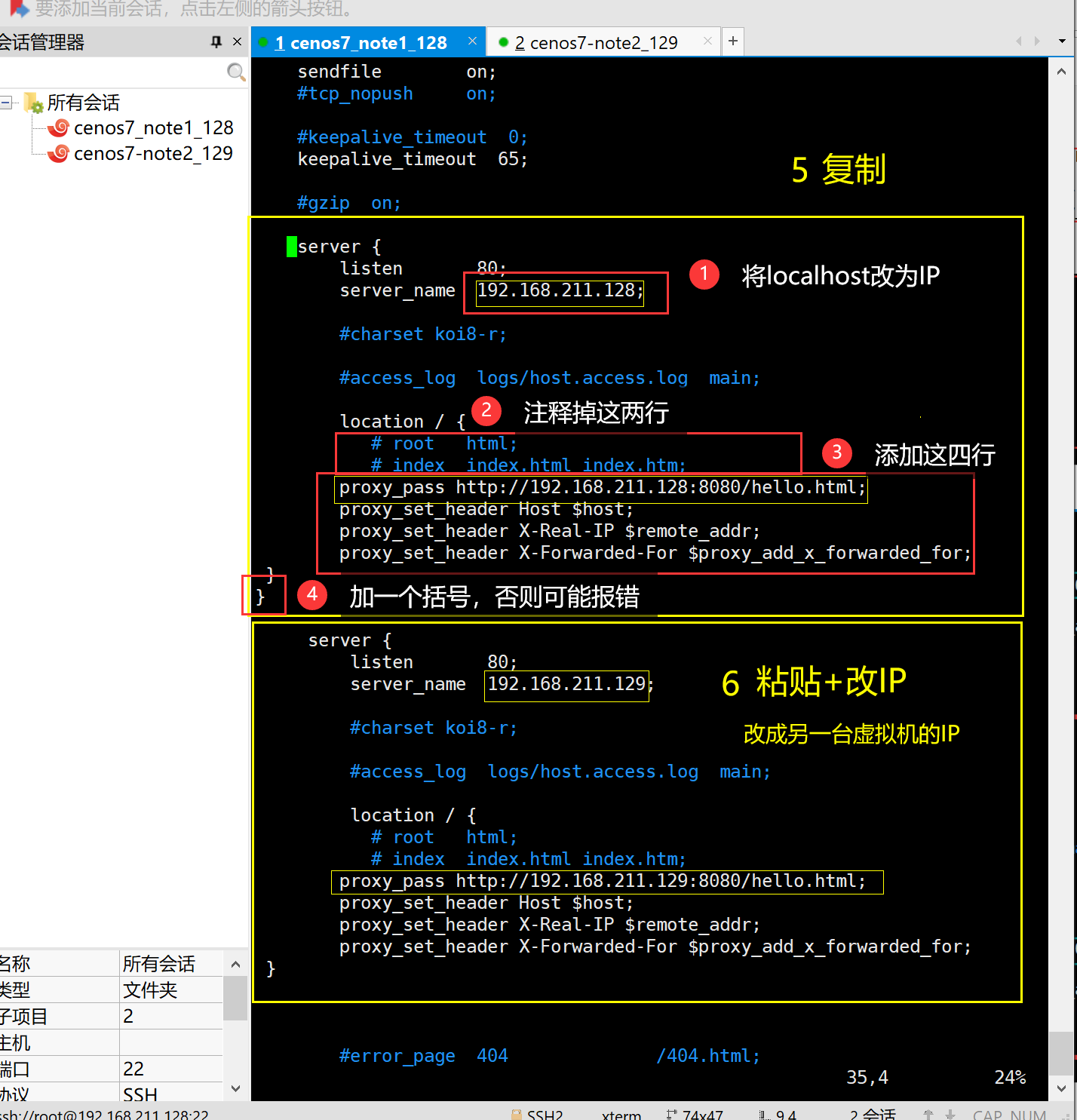
proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

设置了向后端服务器发送请求时的X-Forwarded-For头部信息，用来传递客户端的原始IP地址列表。

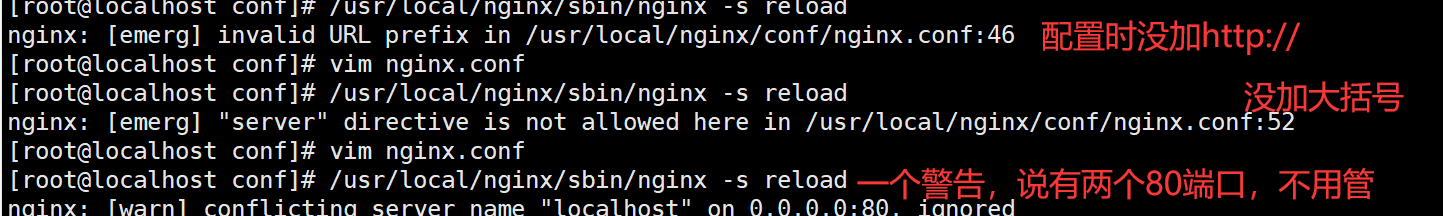
总结：Nginx可以将接收到的请求转发到指定的后端服务器，并且在转发请求时附带了必要的头部信息，确保后端服务器能够正确地处理这些请求。

（3）然后复制整段server，粘贴在此段下面，将localhost和proxy\_pass中的IP改为另一台虚拟机的IP

下图为改完之后的文件



**可能会报的错**

****

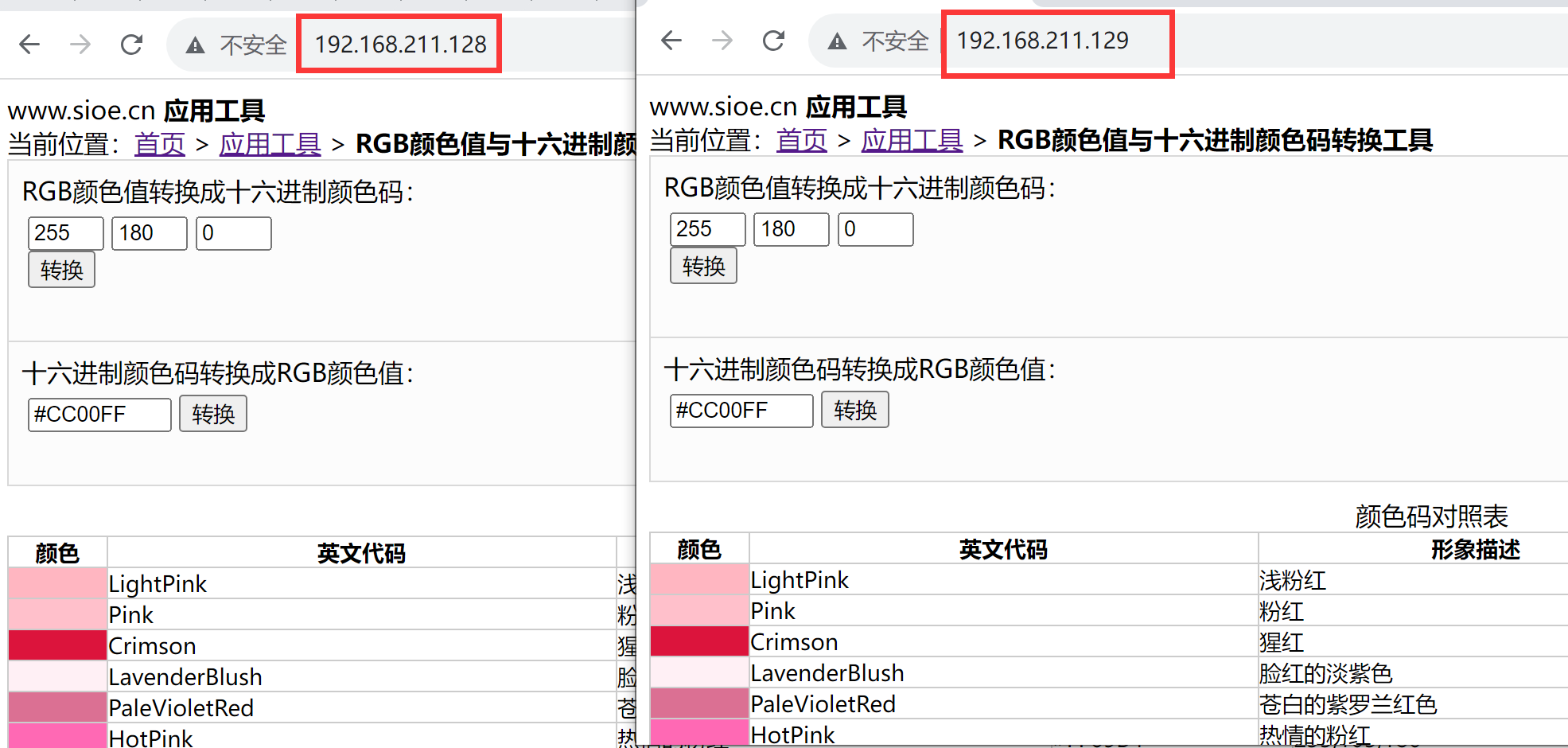
**5.2.重启配置文件**

# /usr/local/nginx/sbin/nginx -s reload

**另一台也按照同样的步骤走一遍，**

**也就是说128的nginx连上两台的tomcat，129的nginx也连上两台的tomcat**

**5.3.打开网页，输入IP，显示hello.html的页面，极为成功，如下图**

****

**6.配置nginx高可用**

**6.1任意目录执行命令**

# yum install keepalived -y

**6.2创建nginx\_check.sh文件，把脚本放到配置文件中所填写的路径中 /usr/local/src/**

**#** vim /usr/local/scr/nginx\_check.sh

nginx\_check.sh脚本内容

#!/bin/bash

A=`ps -C nginx –no-header |wc -l`

if [ $A -eq 0 ];then

/usr/local/nginx/sbin/nginx

sleep 2

if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then

killall keepalived

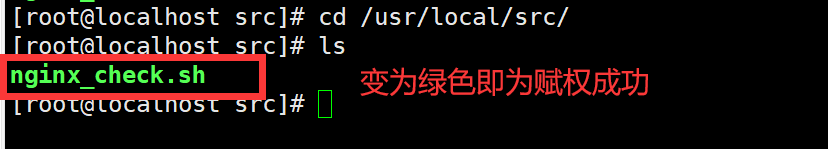
fi

fi

**给脚本赋权：**

# chmod a+x nginx\_check.sh 给所有用户赋权（推荐）

# chmod u+x nginx\_check.sh 给当前用户赋权



**启动一下脚本：**

# sh.nginx\_check.sh

**6.3编辑keepalived.conf文件**

安装keepalived完成后在/etc目录下会有一个keepalived文件夹，文件夹里面有一个keepalived.conf文件

# vim /etc/keepalived/keepalived.conf

替换内容为

**第一台（MASTER）**

global\_defs {

notification\_email {

acassen@firewall.loc

failover@firewall.loc

sysadmin@firewall.loc

}

notification\_email\_from Alexandre.Cassen@firewall.loc

smtp\_server 192.168.211.128 # 虚拟机的IP

smtp\_connect\_timeout 30

router\_id LVS\_DEVEL

}

vrrp\_script chk\_http\_port {

script "/usr/local/src/nginx\_check.sh" # 脚本路径

interval 2 #（检测脚本执行的间隔）

weight 2

}

vrrp\_instance VI\_1 {

state MASTER # 备份服务器上将 MASTER 改为 BACKUP

interface eno16777736 # 网卡 通过ifconfig命令获得

virtual\_router\_id 51 # 主、备机的 virtual\_router\_id 必须相同

priority 100 # 主、备机取不同的优先级，主机值较大，备份机值较小

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress { # VRRP H 虚拟地址

192.168.211.127 #用作高可用

192.168.211.126 #125，126用作重定向

192.168.211.125

}

}

**第二台（BACKUP）**

global\_defs {

notification\_email {

acassen@firewall.loc

failover@firewall.loc

sysadmin@firewall.loc

}

notification\_email\_from Alexandre.Cassen@firewall.loc

smtp\_server 192.168.211.129

smtp\_connect\_timeout 30

router\_id LVS\_DEVEL

}

vrrp\_script chk\_http\_port {

script "/usr/local/src/nginx\_check.sh" # 脚本路径

interval 2 #（检测脚本执行的间隔）

weight 2

}

vrrp\_instance VI\_1 {

state BACKUP # 备份服务器上将 MASTER 改为 BACKUP

interface eno16777736 # 网卡 通过ifconfig命令获得

virtual\_router\_id 51 # 主、备机的 virtual\_router\_id 必须相同

priority 90 # 主、备机取不同的优先级，主机值较大，备份机值较小

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress { # VRRP H 虚拟地址

192.168.211.127 #用作高可用

192.168.211.126 #125，126用作重定向

192.168.211.125

}

}

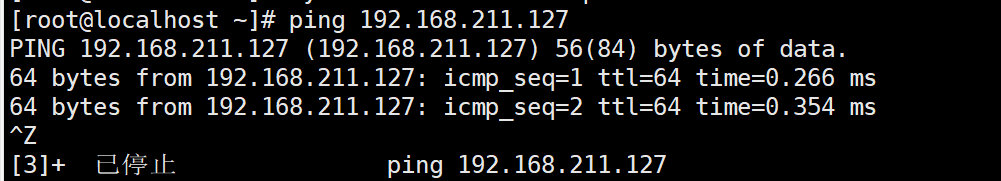
**6.4分别启动两台虚拟机中的Keepalived 和Nginx**

# systemctl start keepalived.service

# cd /usr/local/nginx/sbin

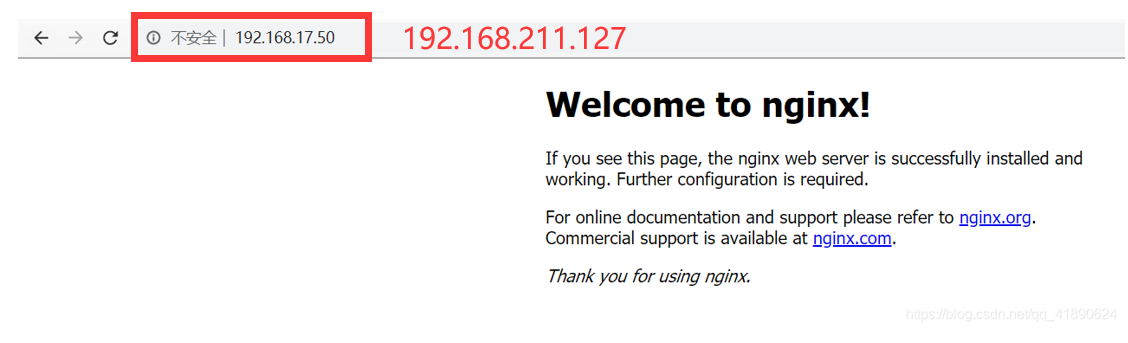
# ./nginx

**配置完之后在本机ping一下虚拟IP**



**6.5 测试**

**（1）访问，只需要访问配置文件中配置的虚拟IP即可**



**（2）停掉其中一台然后再次访问**

停止第一台虚拟机的keepalived和nginx

# systemctl stop keepalived.service

# cd /usr/local/nginx/sbin

# ./nginx -s stop

刷新页面 仍然可以显示welcome to Nginx页面，说明配置成

**7.** **使用rewrite实现多个servername的重定向，统一重定向到ip最后一个数字是127的虚拟ip上**

**7.1修改nginx的配置文件**

# vim /usr/local/nginx/conf/nginx.conf

**在之前的粘贴的server块下添加：**

server{

listen 80; #监听端口

server\_name 192.168.211.125 192.168.211.126;

#server\_name设置为虚拟IP

if ($host = '192.168.211.125' ){

rewrite ^(.\*)$ http://192.168.211.127/$1 permanent;

}

if ($host = '192.168.211.126' ){

rewrite ^(.\*)$ http://192.168.211.127/$1 permanent;

}

}

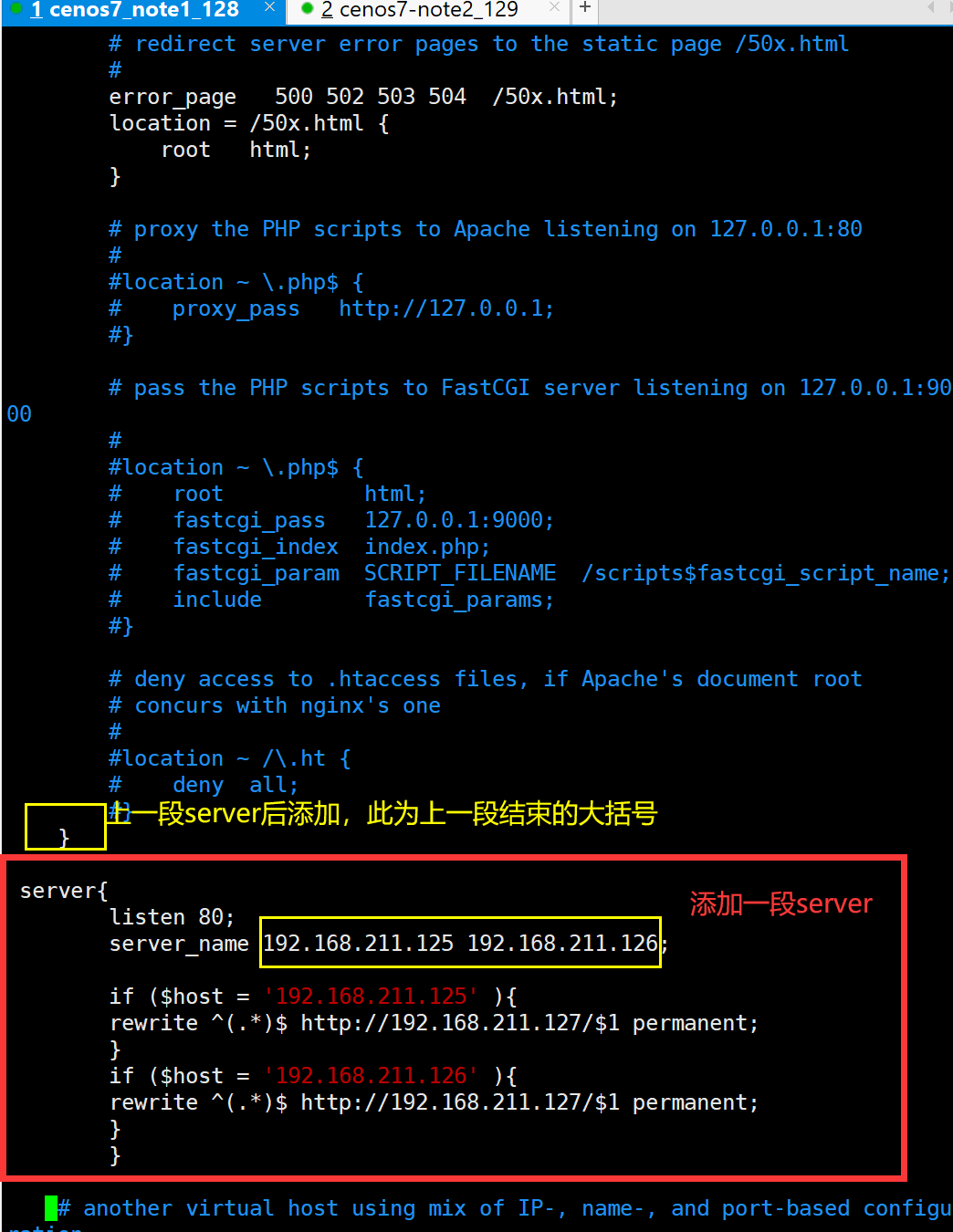
#设置rewrite重定向，用正则表达式，

当输入两个虚拟IP时，重定向到http://192.168.211.127

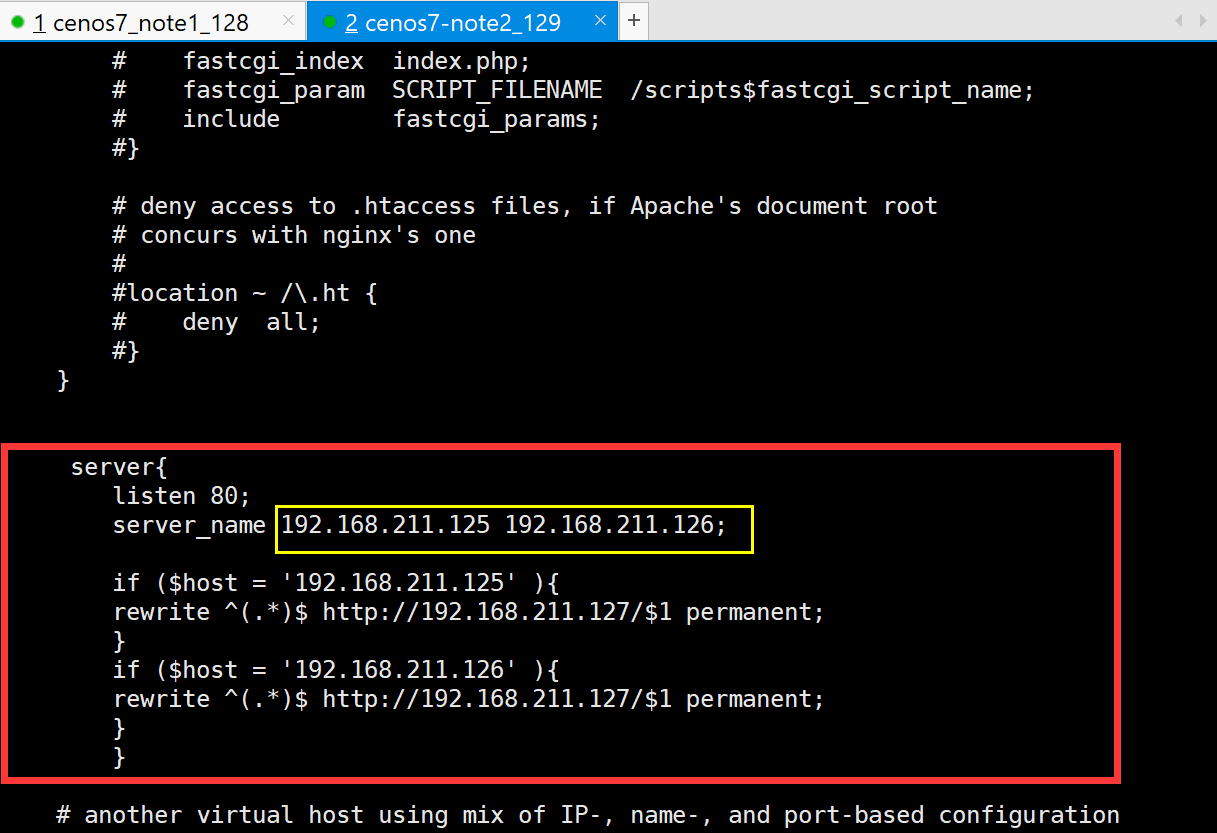
**如下图**

**192.168.211.127用来做高可用**

**所以server\_name写192.168.211.125和192.168.211.126**



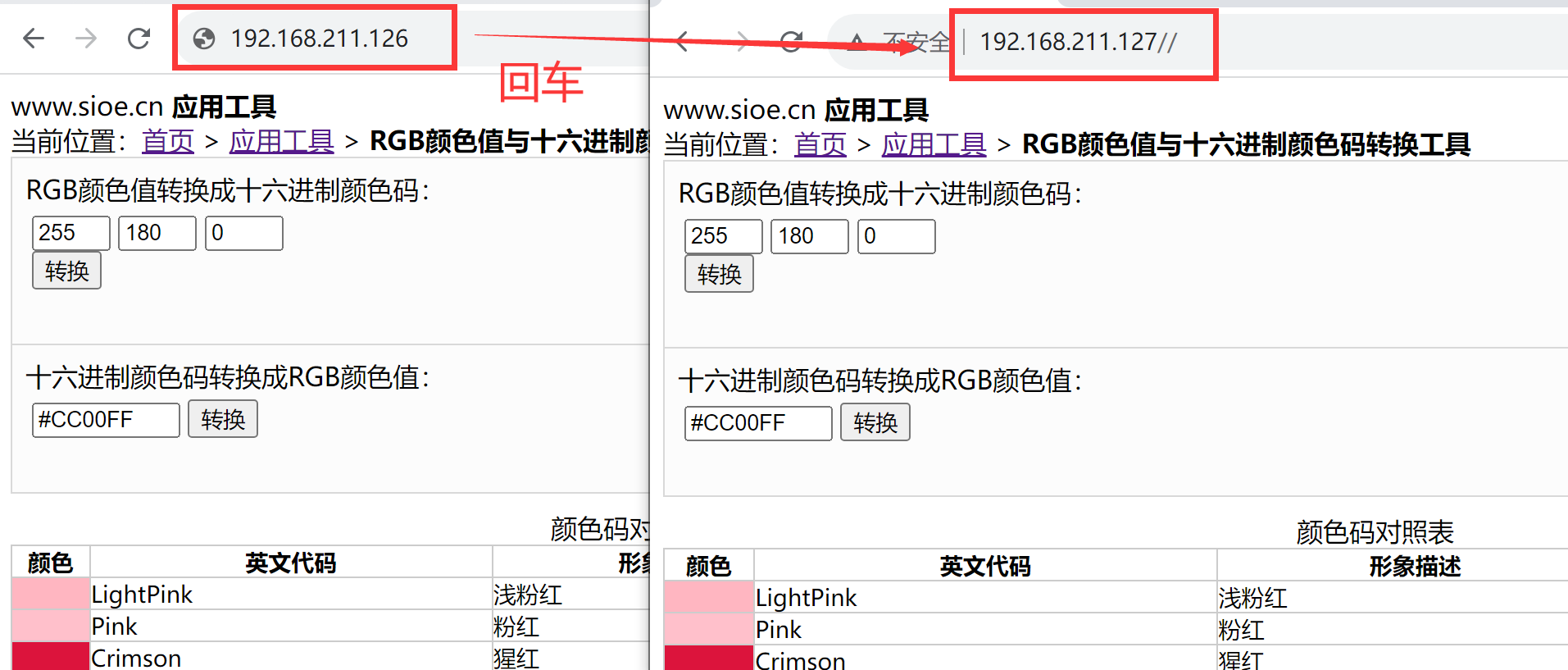
**另一台的配置文件不做任何改变，按照上面同样步骤配置一遍。(如下图)**

****

**7.2 验证一下，网页中输入192.168.211.125 、192.168.211.126，**

**跳转到192.168.211.127//则为成功，如下图**

****

****

**以上，全部配置完成！**

**参考文档：**

**第一步**

**CentOS 7安装Tomcat部署静态网站（5种方法）：**

[**https://blog.csdn.net/qq\_39564555/article/details/101700417**](https://blog.csdn.net/qq_39564555/article/details/101700417)

**第二步**

**\*Keepalived+Nginx+Tomcat搭建高可用的Web服务（一）：主备模式**

[**https://blog.csdn.net/apei830/article/details/78685855?spm=1001.2101.3001.6650.1&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-78685855-blog-131663691.235%5Ev38%5Epc\_relevant\_default\_base&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-78685855-blog-131663691.235%5Ev38%5Epc\_relevant\_default\_base&utm\_relevant\_index=2**](https://blog.csdn.net/apei830/article/details/78685855?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-78685855-blog-131663691.235%5Ev38%5Epc_relevant_default_base&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-78685855-blog-131663691.235%5Ev38%5Epc_relevant_default_base&utm_relevant_index=2)

**\*tomcat 配置nginx 反向代理:**

[**https://blog.51cto.com/cyfang/5454948**](https://blog.51cto.com/cyfang/5454948)

部署Nginx服务器，实现对Tomcat服务器的反向代理

[**https://blog.csdn.net/HelloWorld\_4396/article/details/112860389**](https://blog.csdn.net/HelloWorld_4396/article/details/112860389)

**第三步：**

**详解Nginx中的Rewrite的重定向配置与实践：**

[**https://www.yii666.com/blog/223906.html**](https://www.yii666.com/blog/223906.html)