下载地址 http://nginx.org

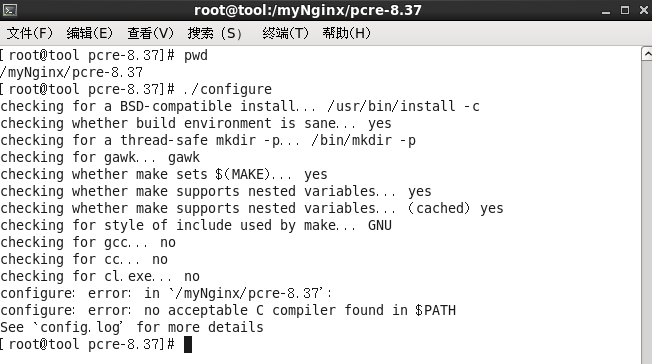
# 安装nginx

## 安装pcre依赖

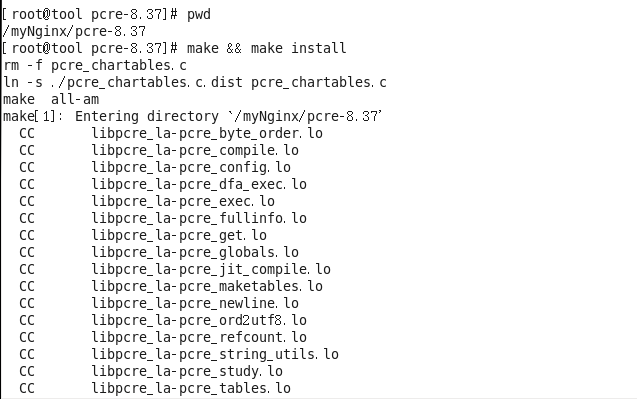
### 1.1.1、解压pcre

tar -zxvf pcre-8.37

### 1.1.2、进入解压目录 执行 ./configure



### 1.1.3、把文件编译并且安装 make && make install



pcre-config --version 查看版本



## 安装zlib依赖等其他依赖

使用命令：

yum -y install make zlib zlib-devel gcc-c++ libtool openssl openssl-devel

## 安装nginx

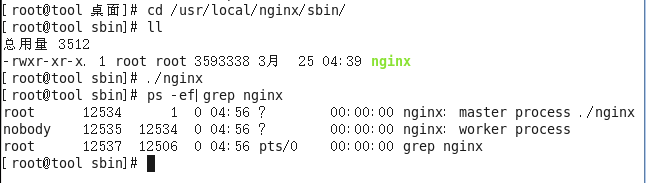
拷贝解压

./configuer ,

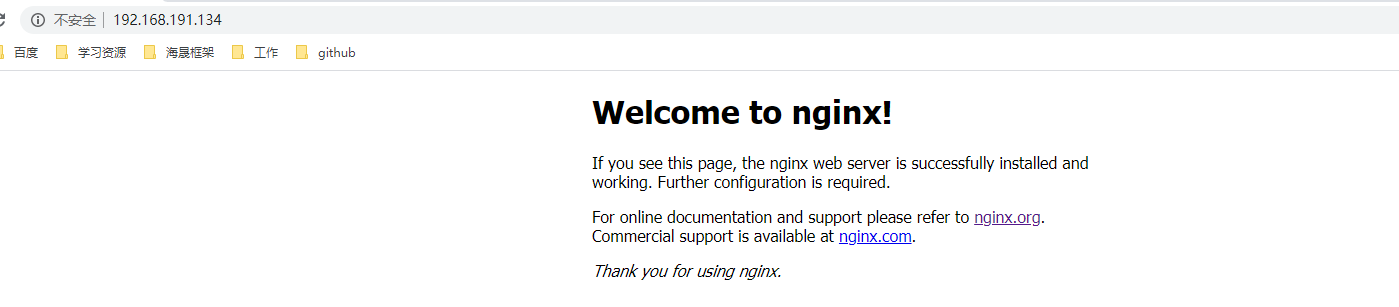
make && make install 三步骤

装好nginx之后，会在/usr/local文件中创建一个nginx文件夹，该文件夹的sbin目录中有启动脚本。

## 启动nginx



访问的默认端口是80端口

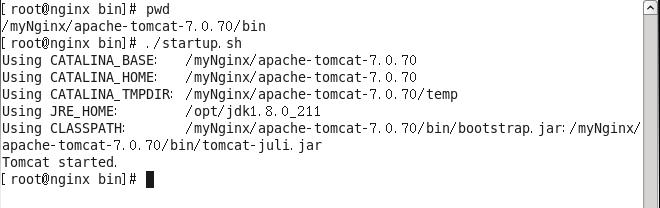


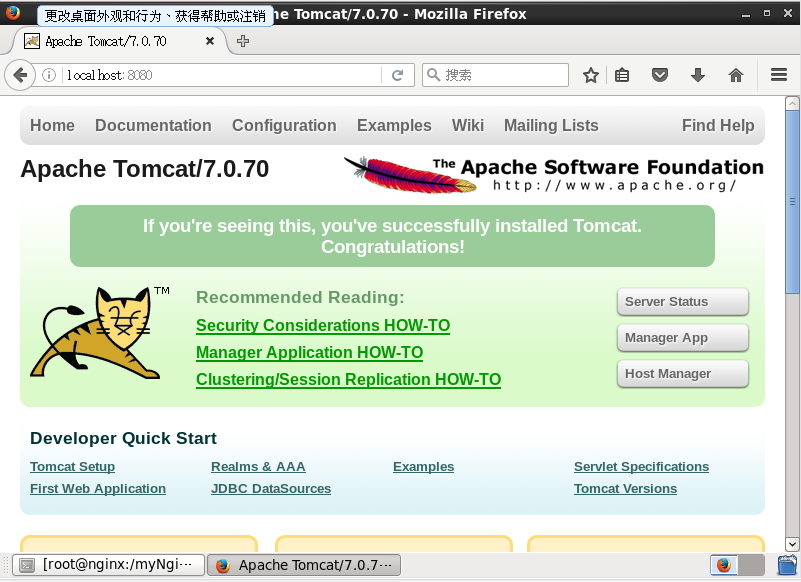
# nginx反向代理1

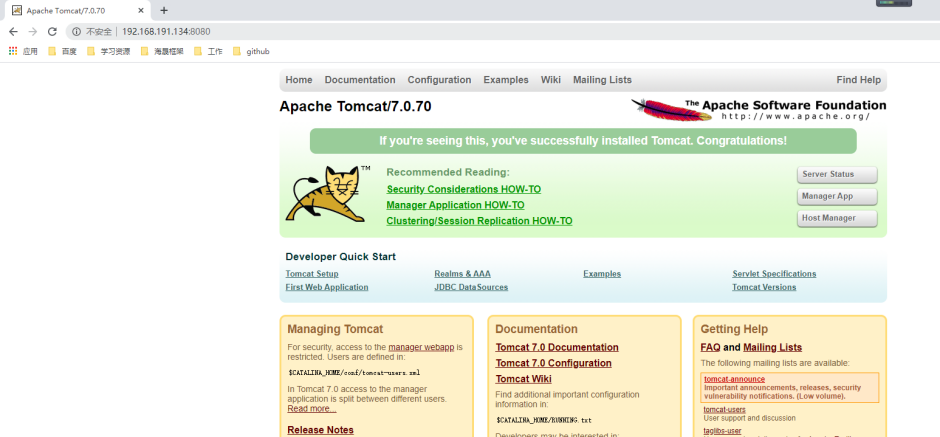
实现的效果：在浏览器中输入www.123xiaosha.com，跳出的窗口是tomcat默认的页面。

## 2.1、安装tomcat

拷贝、解压，需要一个JDK

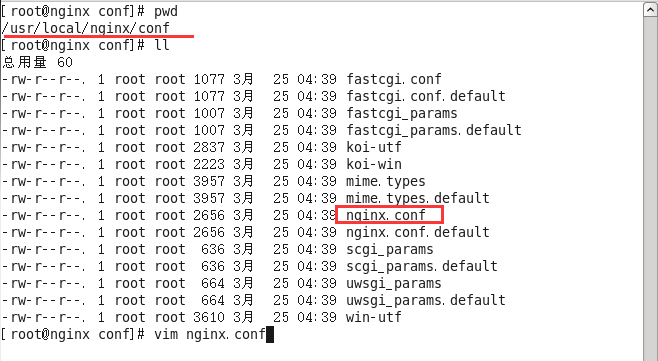


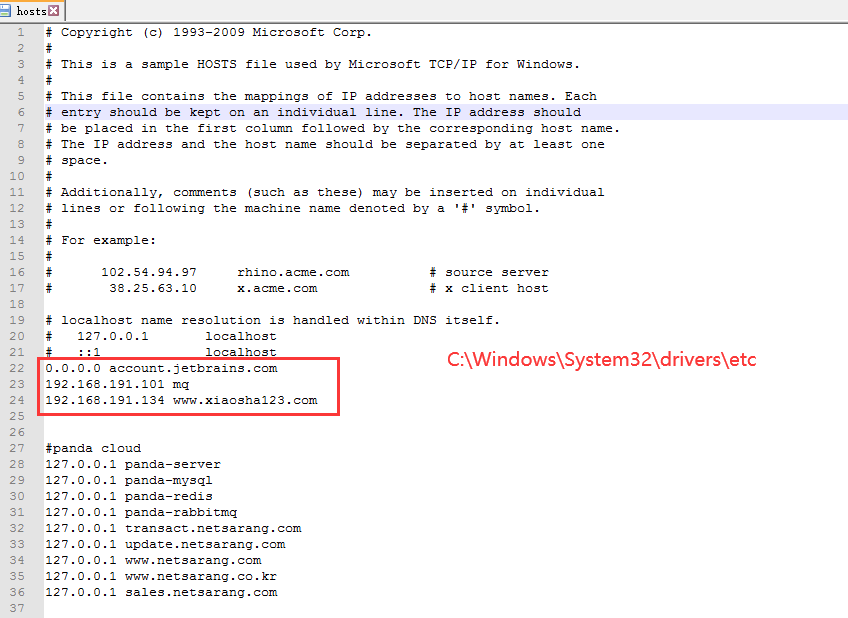




## 2.2、反向代理

操作文件之前先备份。



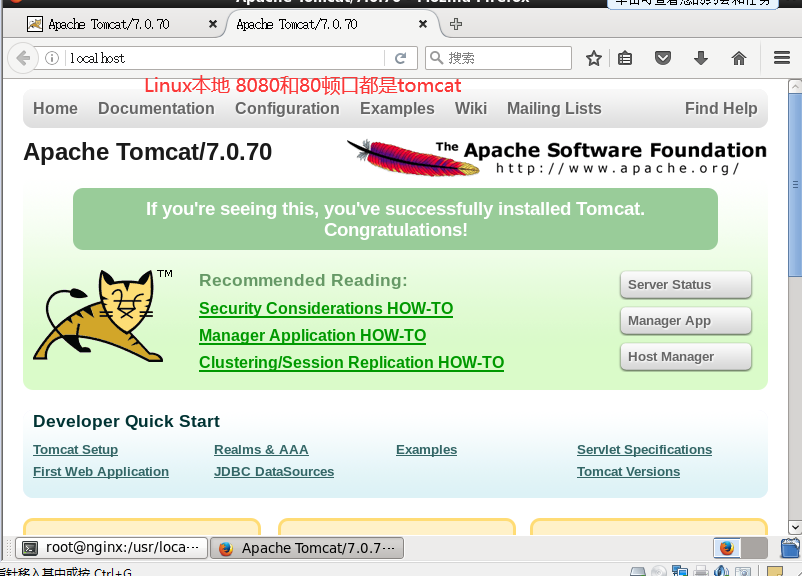




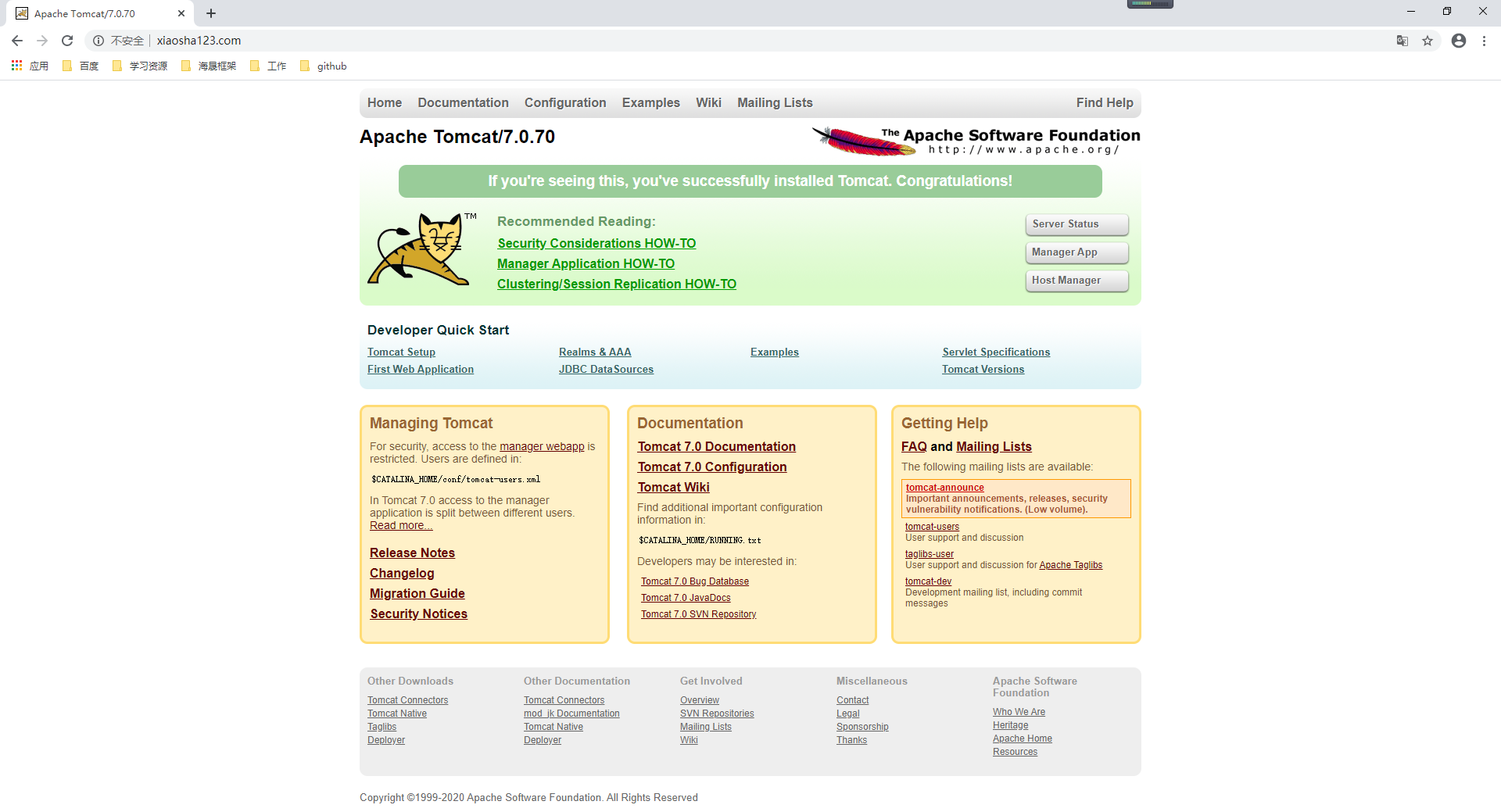
重启nginx

到/usr/local/nginx/sbin 目录下

查看结果



windows访问： [www.xiaosha123.com/](http://www.xiaosha123.com/)

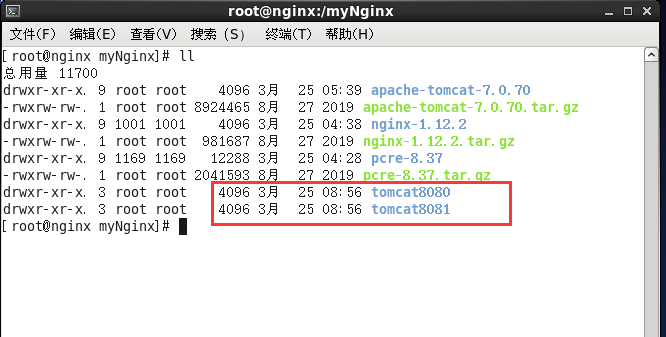


# nginx反向代理2

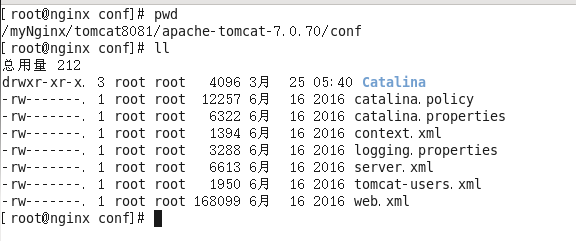
访问http://127.0.0.1:9001/edu/ 跳转到 127.0.0.1:8080端口

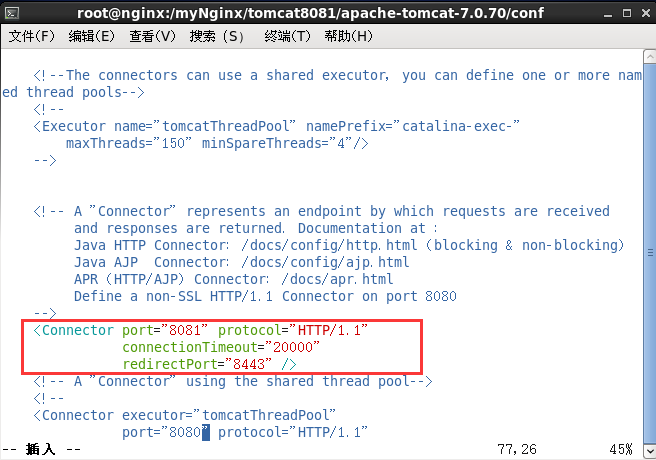
访问http://127.0.0.1:9001/vod/ 跳转到 127.0.0.1:8081端口

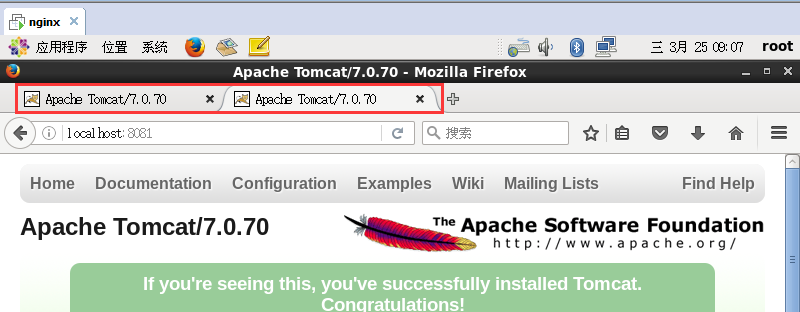
## 3.1、在两个目录中准备两个tomcat



**编辑server.xml 修改端口**



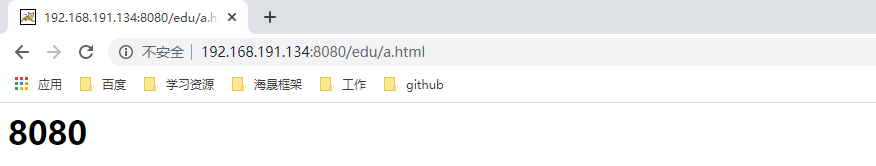




## 3.2、准备测试文件

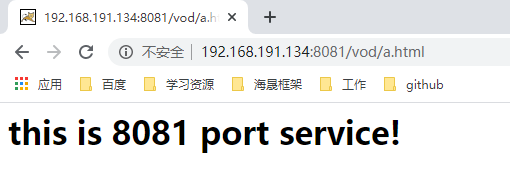
8080下放一个访问路径是/edu/a.html的测试页面。





8081下是一个访问路径是/vod/a.html的测试页面

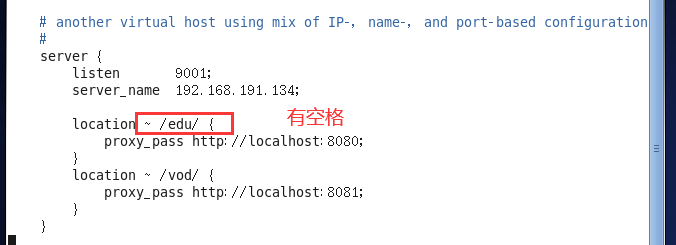


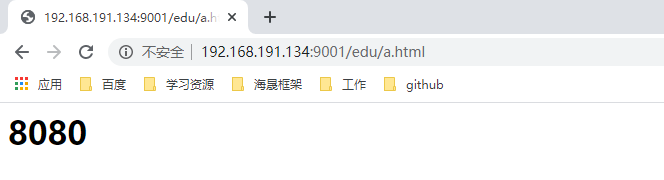


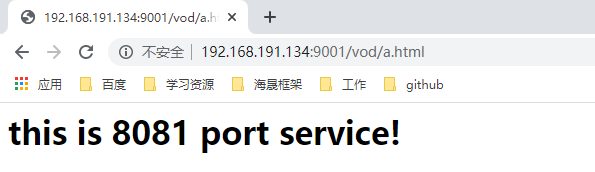
## 3.3、配置反向代理的效果

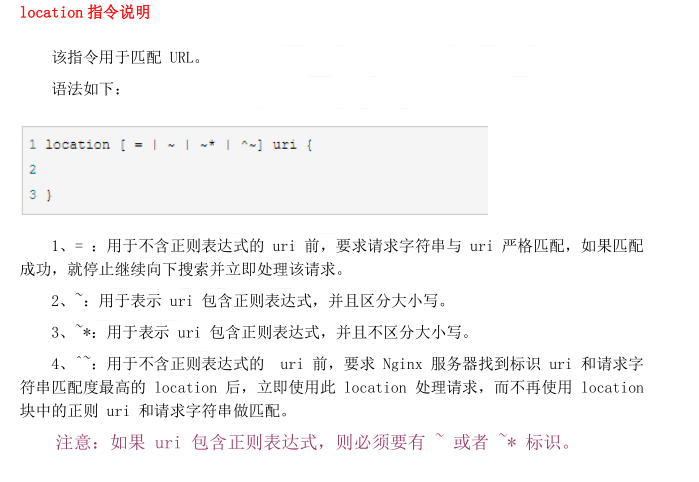
**加一个**

有空格和没空格是一样的。









# 负载均衡

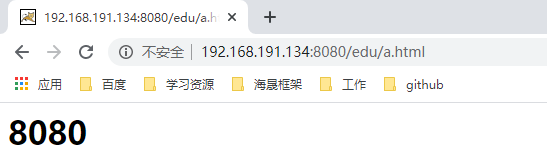
实现的效果：

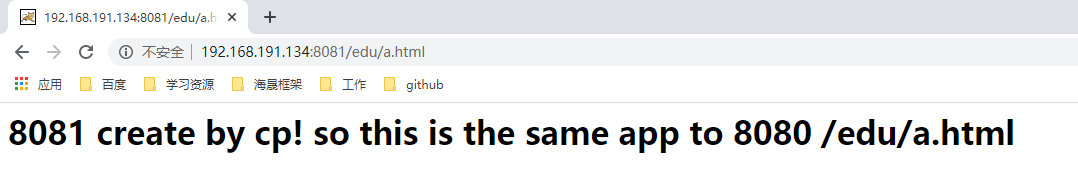
浏览器输入http://192.168.191.134/edu/a.html ,多次请求平均分摊到8080和8081端口上去。

## 4.1、准备工作

同样需要两台服务器。（多台也可以）这两台tomcat的webapps文件夹都拥有一个叫做 edu的文件夹，里面都有一个a.html的文件。

准备工作完成之后：



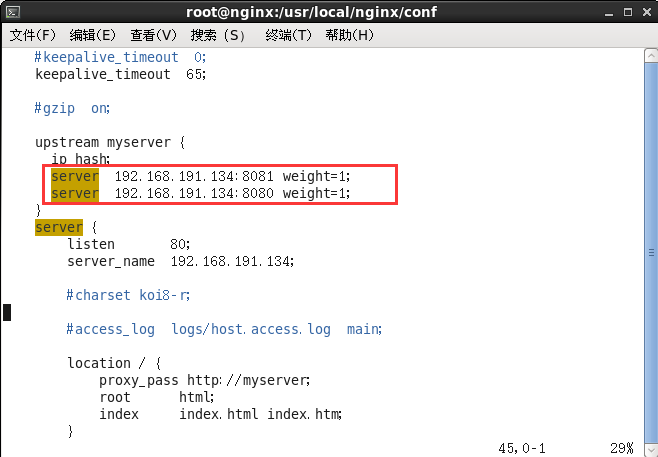


## 4.2、配置nginx

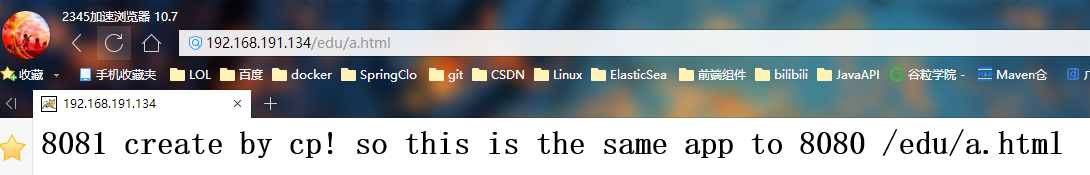


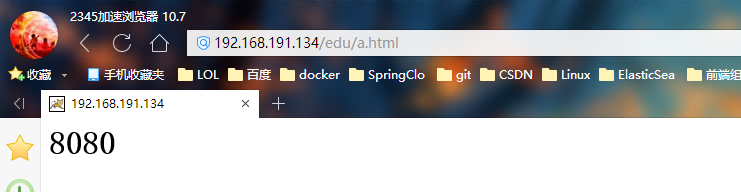


试了好几遍，谁写在前面就一直分配给谁。



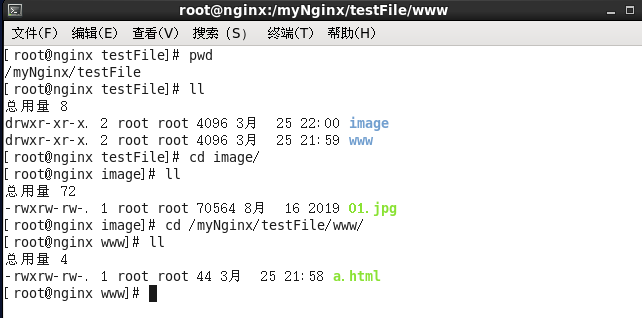
谷歌不行，2345浏览器可以。





# 动静分离

## 5.1、资源准备

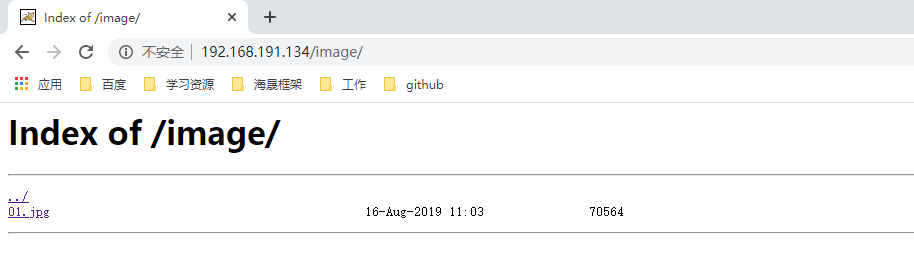


## 5.2、动静分离配置

这个图片列举界面好像似曾相识的感觉。

因为配置了，就能把文件夹列表显示出来。

autoindex on;

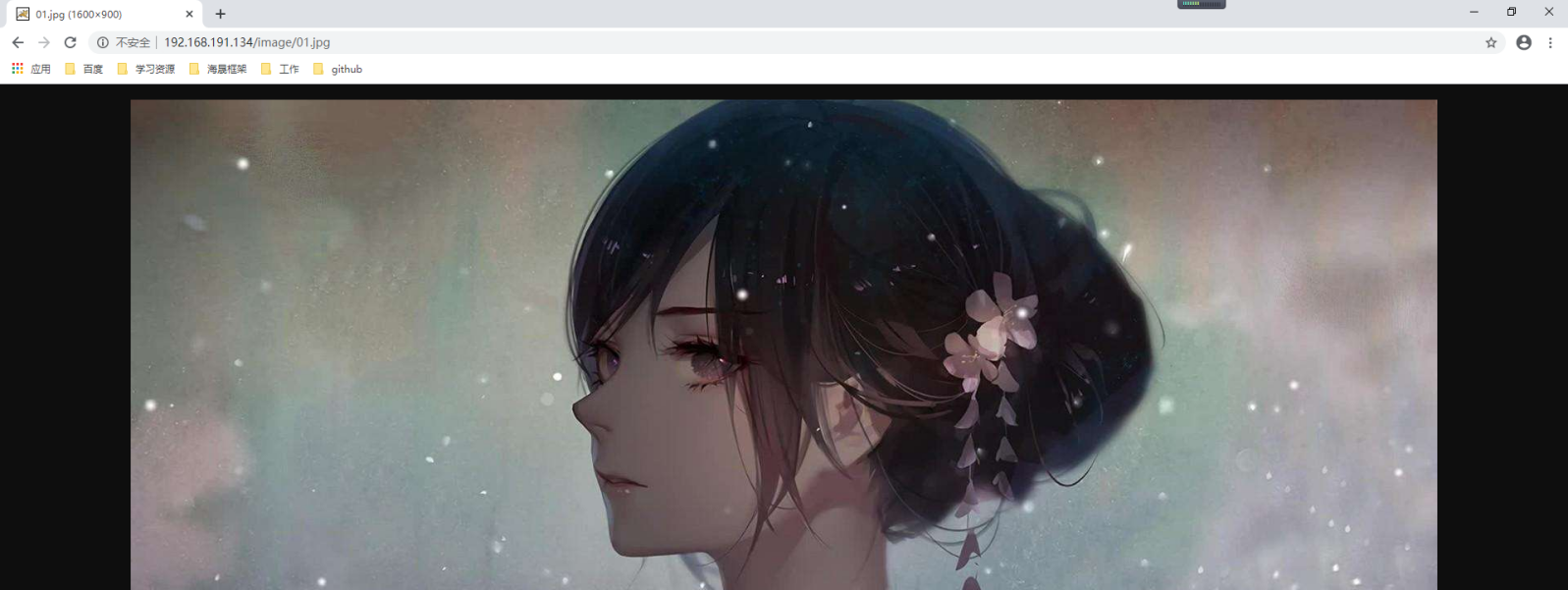


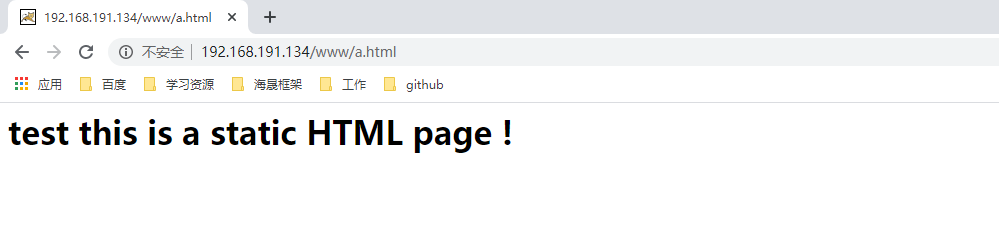


配置



测试

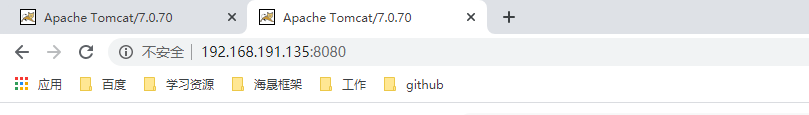


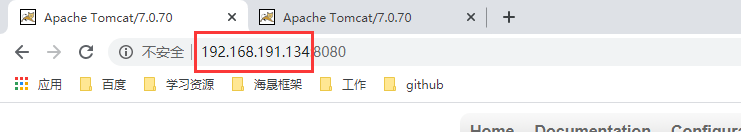


# HA

## 6.1、准备工作

克隆一个虚拟机不就完事了么。





给两个服务器都安装上 keepalived

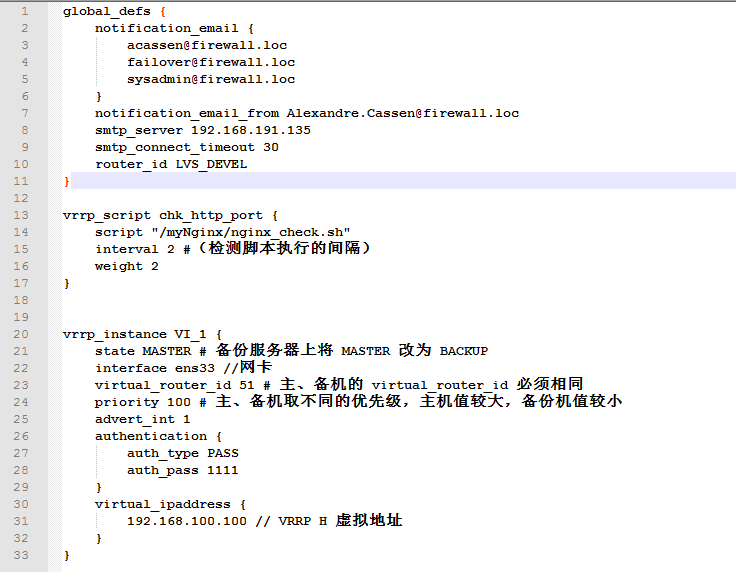
**yum install keepalived -y**

**安装到了 etc keepalived的配置文件。**

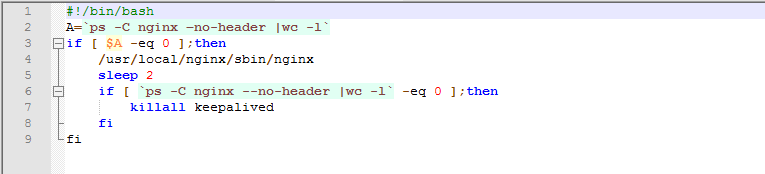


## 6.2、高可用配置

主要就是修改keepalived的配置文件。



检测脚本



**测试：**

**先启动nginx**

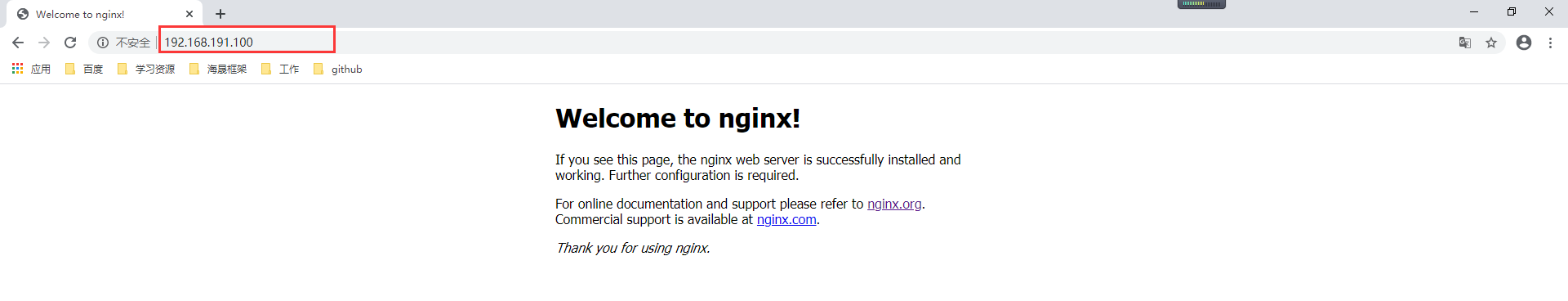
**再启动keepalived**

**命令： syatemctl start keepalived.service 这个命令是centos7之后的**

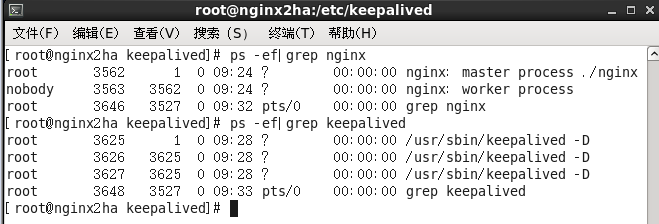
 /etc/init.d/keepalived start

## 6.3.1、正常启动运行情况

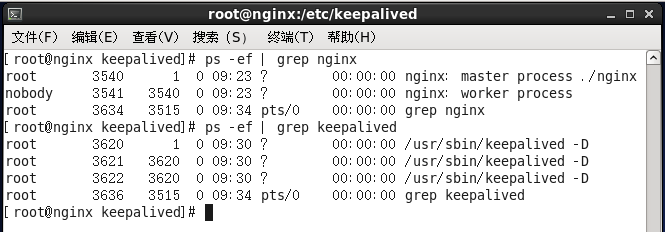
两台机器都启动之后：



**从机上nginx和keepalived运行情况**



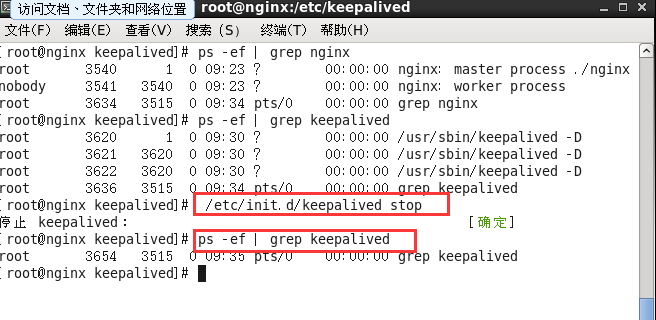
主机运行情况



## 6.4、把主机进程关闭，测试高可用

两台机器情况下，把主机关闭，丛机仍然能够保证nginx可用。

/etc/init.d/keepalived stop



页面

