

**#创建的是不可以登陆的普通用户**

**#可以做网站也可以做反向代理**

**#在浏览器的开发者工具中可以查看开发的软件，F12**

**#根据公司开发语言来选择软件**

**#安装非默认模块（./configure是安装默认模块）**

**#满足大并发，5w; apche只能满足2-3w并发**

**#静态网站，PHP、Python开发**

**#JAVA开发**

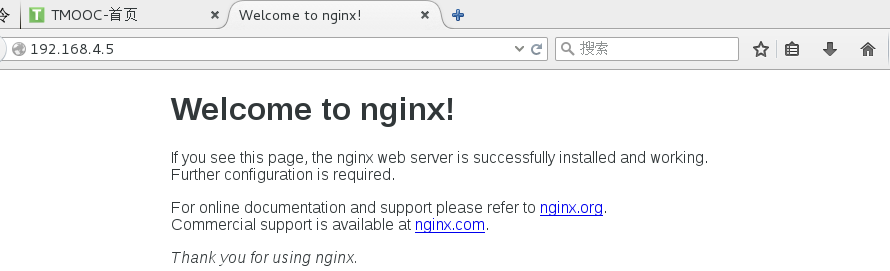
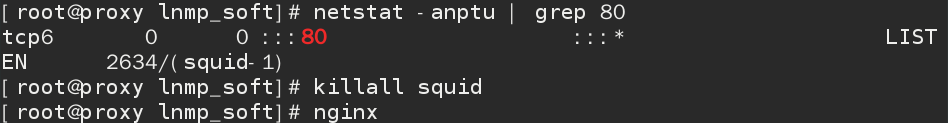
**搭建Nginx服务**

1. 安装nginx软件包

脚本安装；依赖包[yum -y install gcc pcre-devel（支持正则） openssl-devel（支持加密）] useradd -s /sbin/nologin nginx

2.启用nginx

**#查看80端口使用者**



**#浏览器可以查到页面**

**#查看nginx状态**

**#systemctl stop squid**

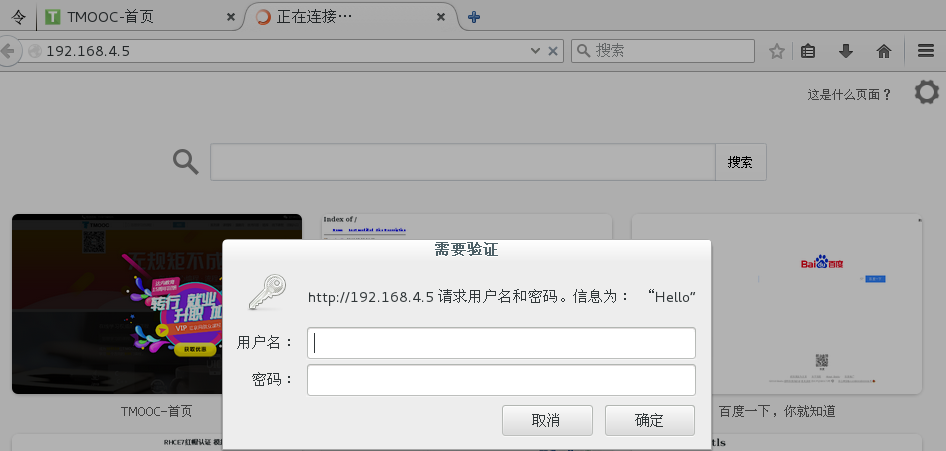
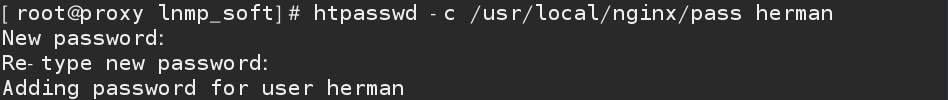
**用户认证**



**nginx主配置文件，全局配置[用户、进程、并发、日志]**

**nginx中，每一个server是一个虚拟主机**





**重启服务，重新加载配置文件stop关闭**

**创建新密码文件（密文）**

**追加用户，不要-c**



可能错误：配置文件中密码文件名和创建的密码文件名不同或路径不同；密码文件的权限

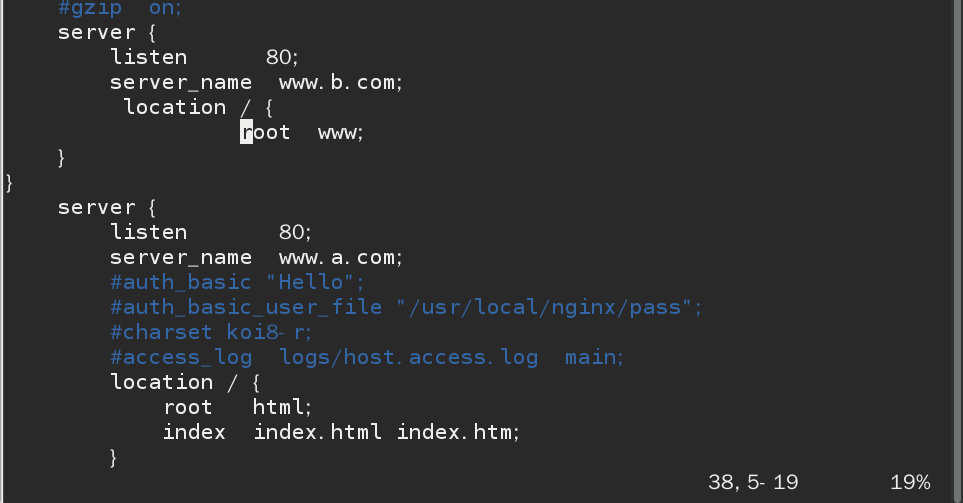
**排错查看日志** tailf /usr/local/nginx/logs/error.log

tailf /usr/local/nginx/logs/access.log

虚拟主机：基于域名、基于端口、基于域名

**基于域名的虚拟主机**





**#实验环境没有DNS，直接修改主机名和ip配置文件**

**#网页内容**

**#创建网站根目录**

**#启动、重启Nginx**

**#指定网站根路径**

**\*相对路径，在/usr/local/nginx下创建**

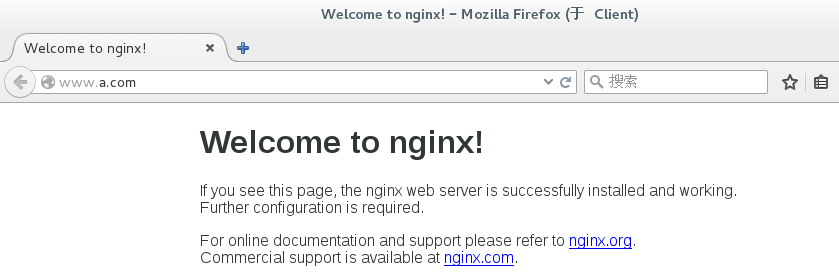
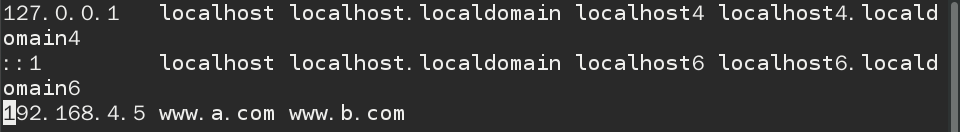
**#指定网站根路径**

**#域名**

**#端口**

**#域名**

**#端口**



**#检验**

**#用该文件做域名解析**

常见加密算法：

对称加密(单机)：AES，DES（压缩软件加密）

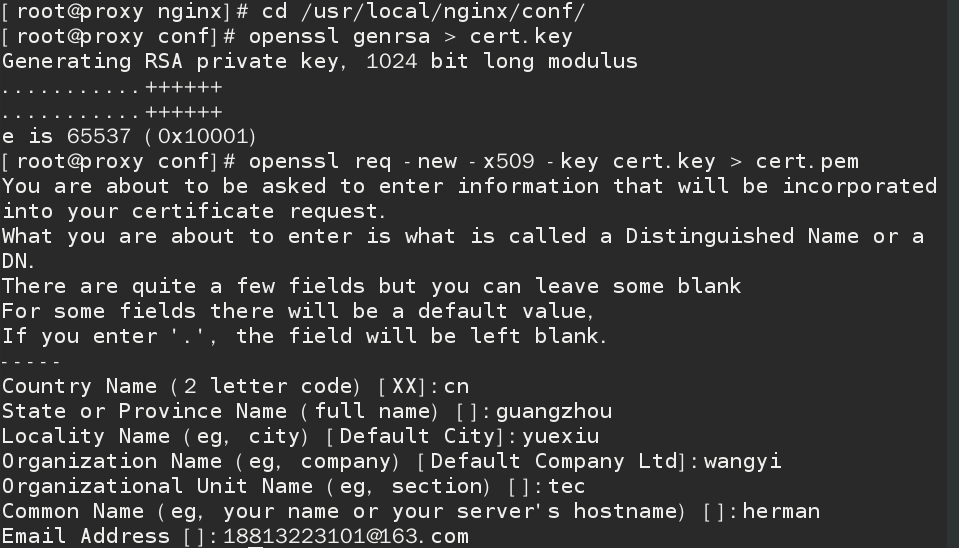
非对称加密（网络）：RSA，DSA

信息摘要：md5,sha128,sha512,sha256（确保数据安全）

**#例，生成MD5码**



**SSL虚拟主机（基于加密网站的虚拟主机）**



**#加密算法**

**#调用密钥**

**#邮箱**

**#名称**

**#部门**

**#公司**

**#城市**

**#省份**

**#国家**

**#生成证书，（公钥）**

**#生成私钥**





**#修改客户端本地解析文件**

**#证书**

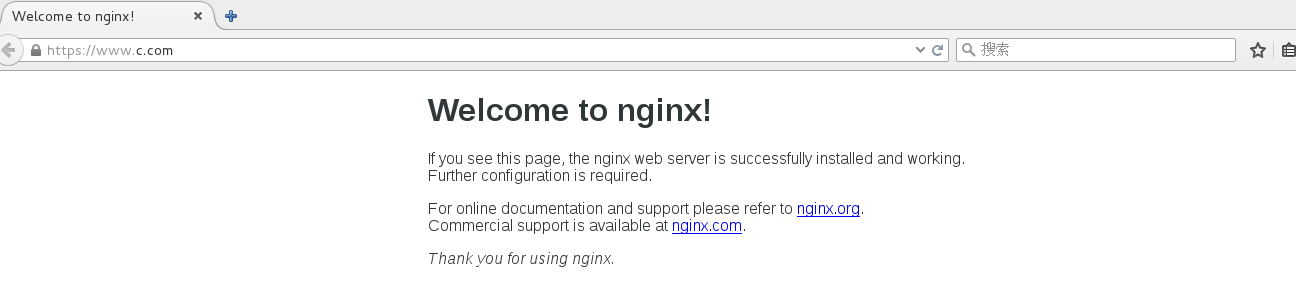
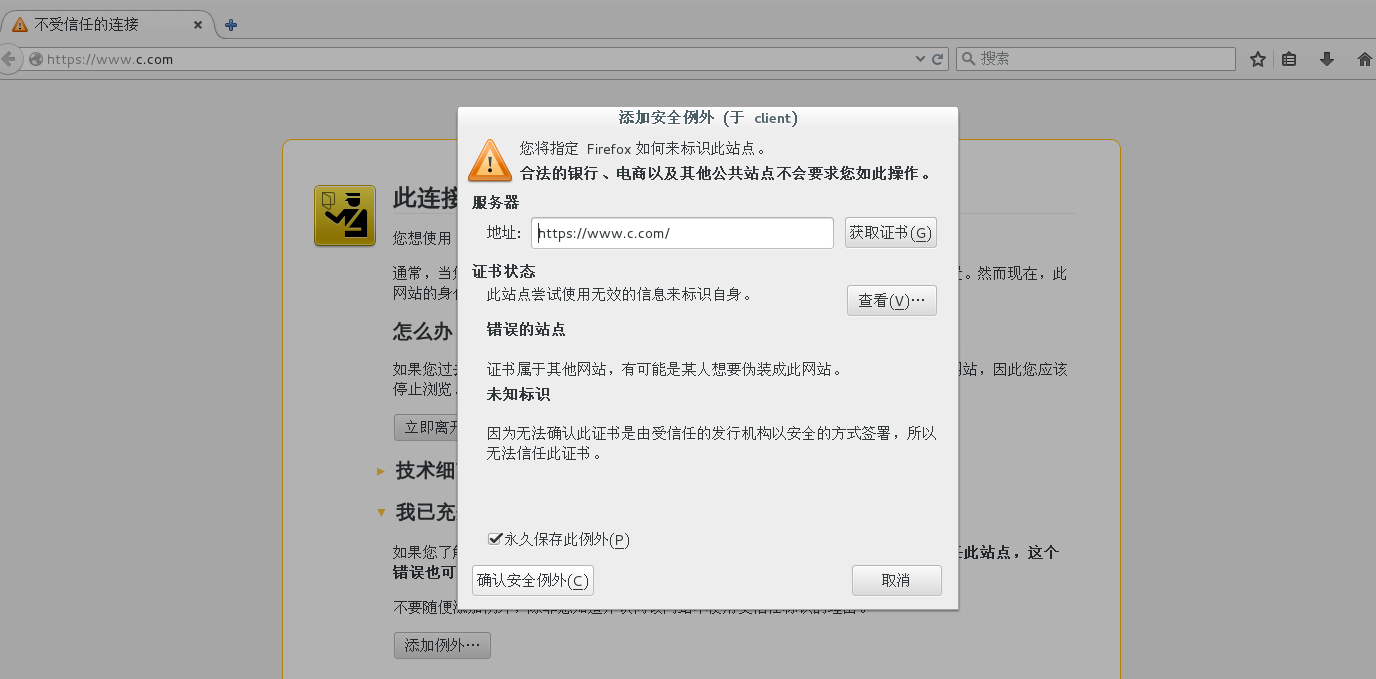
**#私钥**





**#客户端验证**





**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*梳理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**nginx安装 【依赖， ./configure模块】**

**nginx升级 【make upgrade】**

**nginx 【默认welcome测试网站】**

**nginx 【用户认证】**

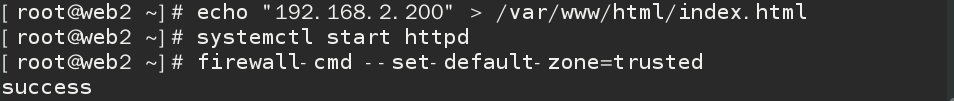
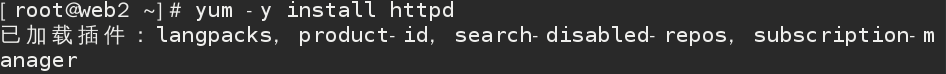
**nginx 【server{}虚拟主机】**

**nginx 【https 加密网站】**

**nginx 【调度器】 负载均衡，web高可用**

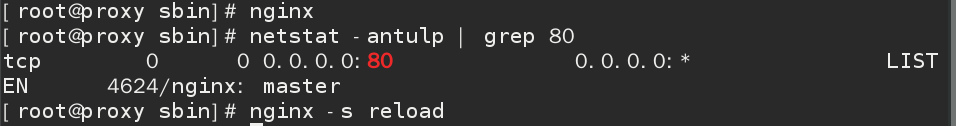
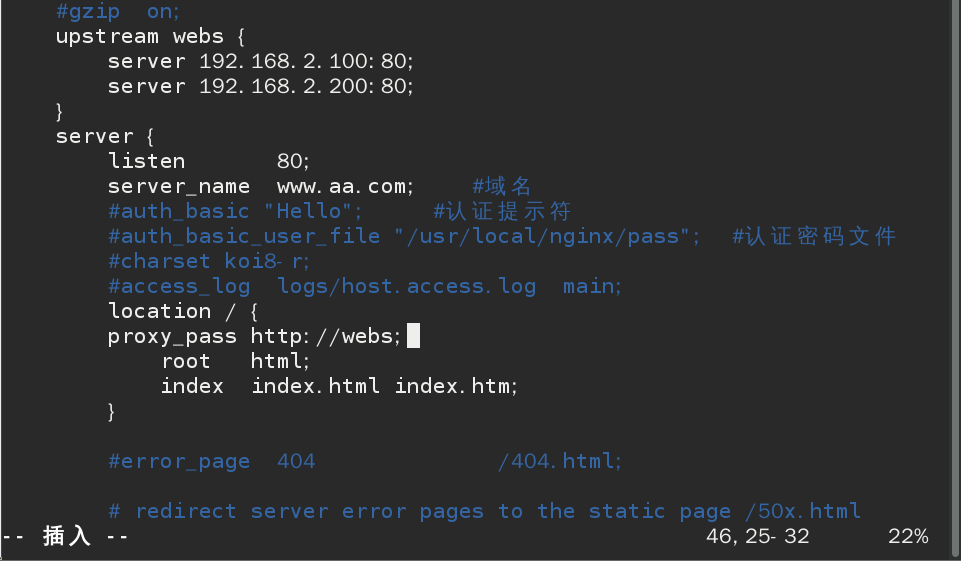
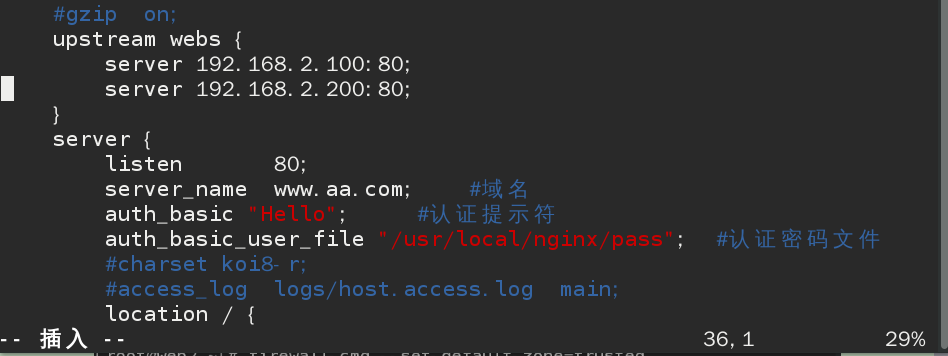
**\*\*\*\*\*\*\*反向代理：【调度器】 负载均衡，web高可用\*\*\*\*\***

**1.搭建后端环境**



1. **做调度器，配置Nginx服务器，添加服务器池，实现反向代理功能**





**#调用、转发集群，如果有人域名或者IP访问，直接调用上面定义的集群，后面的语句不会再执行(root html上面)**

**#手动添加，集群名字webs,定义集群，有几台主机就写几个server**

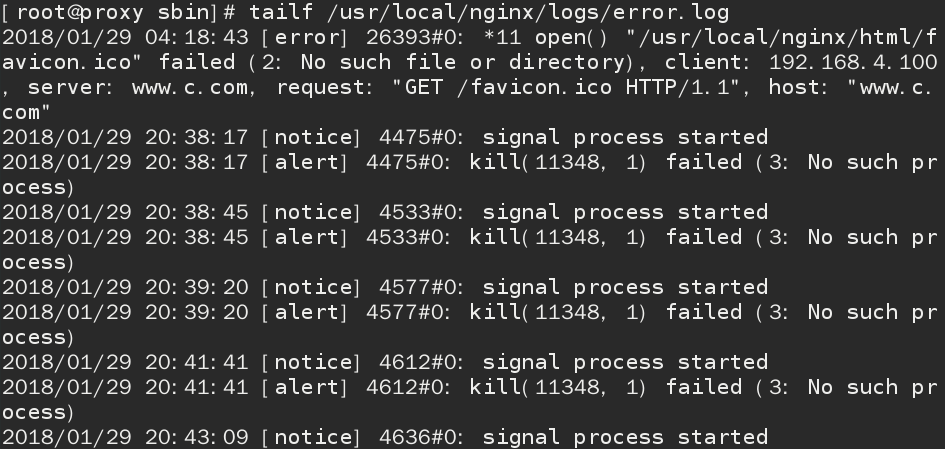
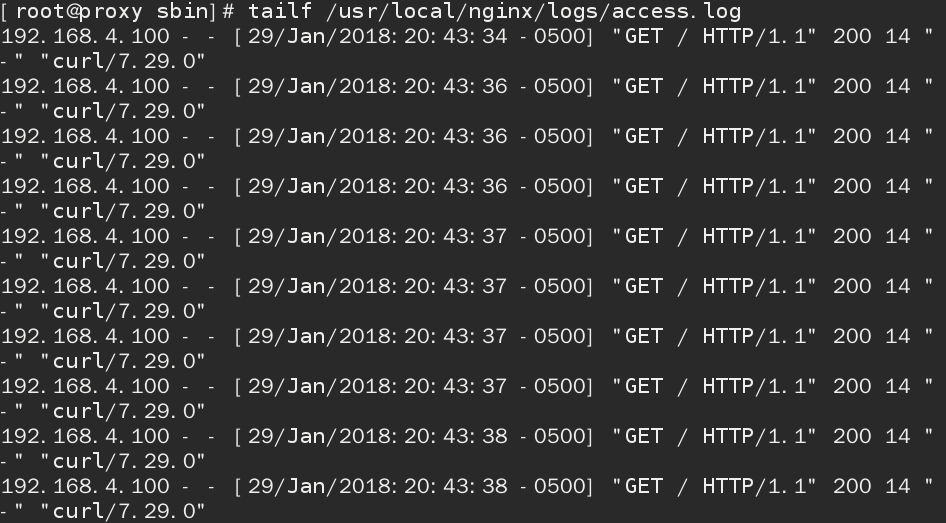
1. **客户端验证**



动态查看日志



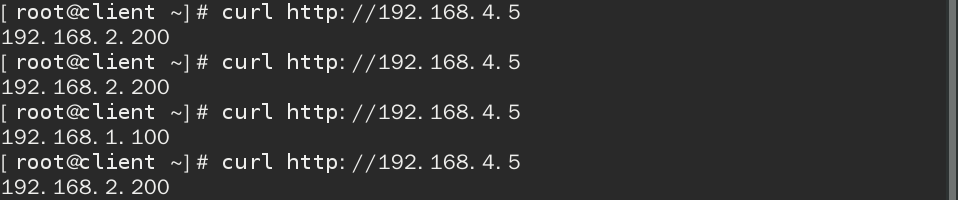
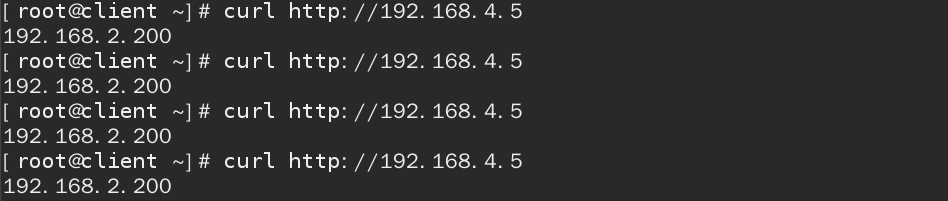
**#访问日志**

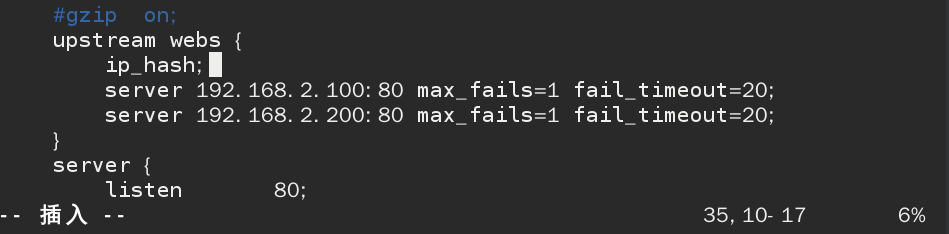


**#错误日志**

检测高可用







**#权重分配，根据机器性能配置来分配权重**

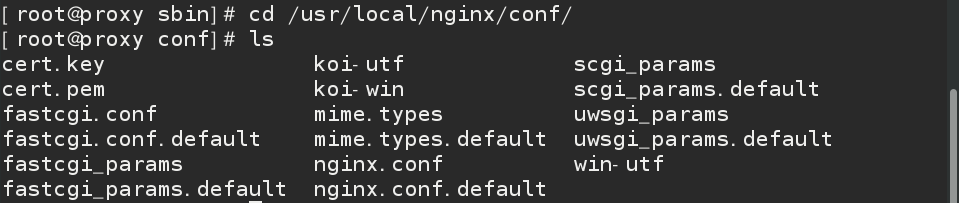
**#失败后20秒内不再Ping该主机**

**#最大失败次数，Ping有一次失败认为机器死机**

**#算法（默认是轮询）设置相同客户端访问相同WEB服务器,对整个集群生效**

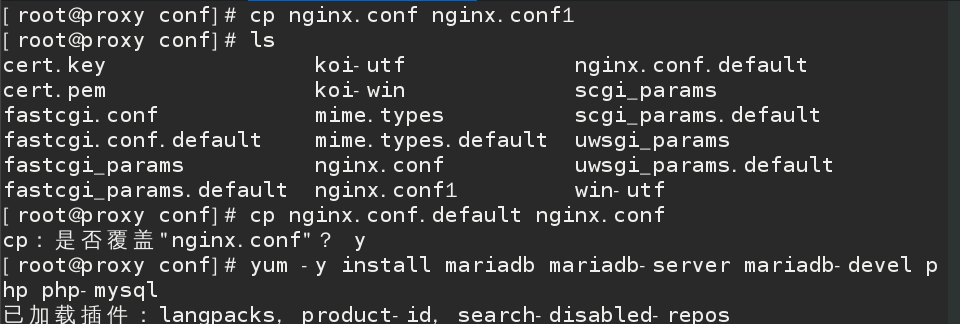
**#人为诊断主机不可用，可以标记down，让它关机，就不会再ping这台主机了**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



**#原始模板**

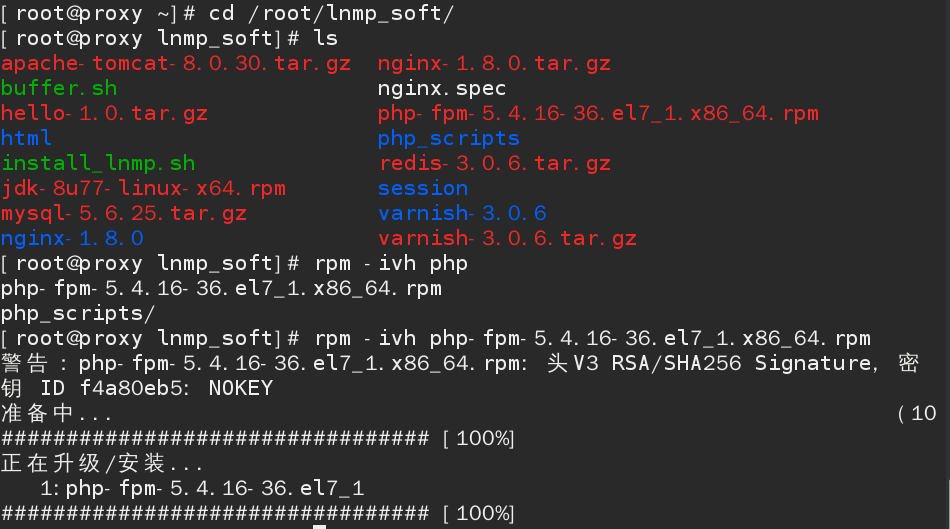
**\*Nginx[动态+静态网站] LNMP[Linux+Nginx+Mariadb+PHP\Python\perl]\***



**#安装软件**

**#用模板文件覆盖原配置文件**

**#将自己前面改的配置文件备份**



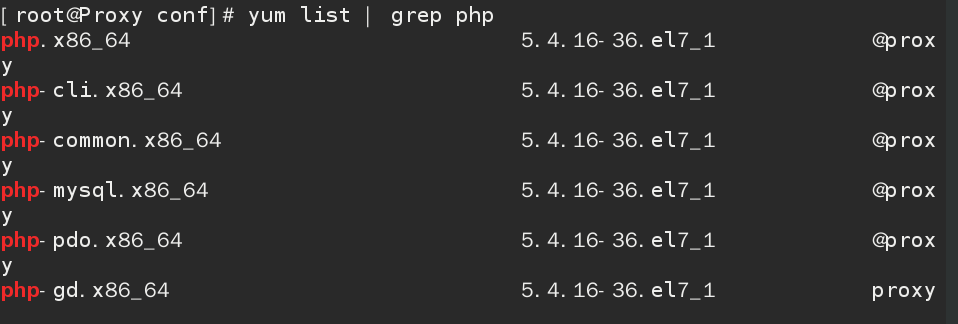
Nginx【源码安装】【web服务器，接收用户请求】

php【解释器】php-mysql【可连接数据库】+php-fpm【服务,允许并发,rpm安装的软件日志在/var/log下】\*\*模块化安装

Mariadb【数据库客户端软件,mysql】+mariadb-server【数据库服务】

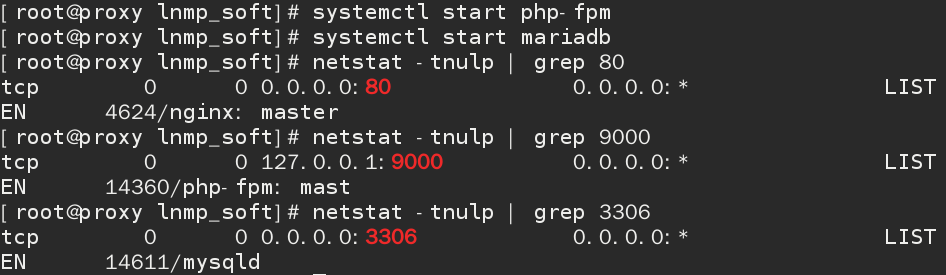
mariadb-devel(运行mysql 的**依赖包**)

\*\*\*安装nginx报错，require pcre libariy--->依赖包pcre-dev



**#显示PHP相关的模块**

**修改配置文件，将这些软件串连在一起，工作**



1. **nginx+PHP(实现动静分离)**

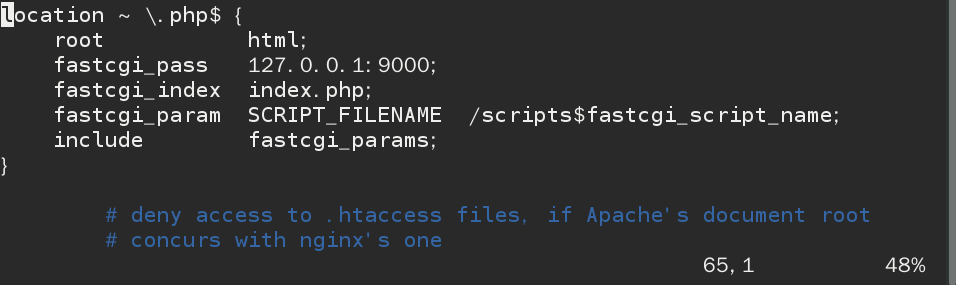
**动态：代码需要再服务器上执行，将执行的结果给用户**

**静态：用户访问的任何界面，服务器直接将文件给用户**

**用户发送请求到Nginx ：80**

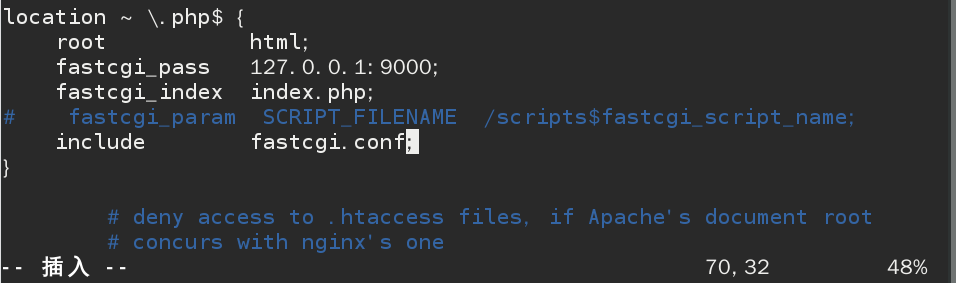
**nginx要分析用户的请求，是静态还是动态，如果是静态，则nginx找root网站根[如果可以找到，就成功返回；否则报404]；如果是动态，则nginx转发给9000端口(php-fpm)。**

**配置文件中，location的作用是匹配用户的地址栏，支持正则（模糊匹配，location~\* ~表示模糊匹配，\*表示不区分大小写）。如果没有匹配，则跟根匹配，也就是第一个location**

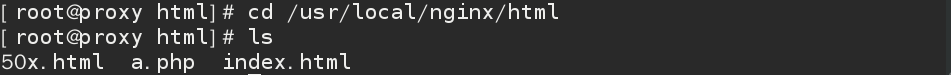
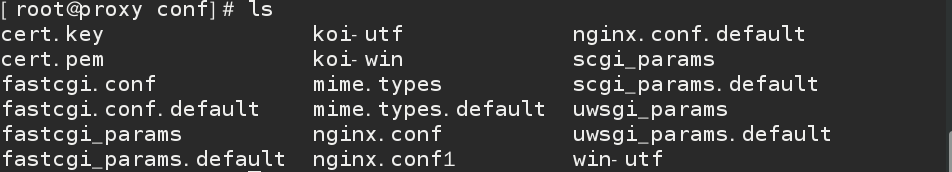


**#原来是注释，将注释打开**





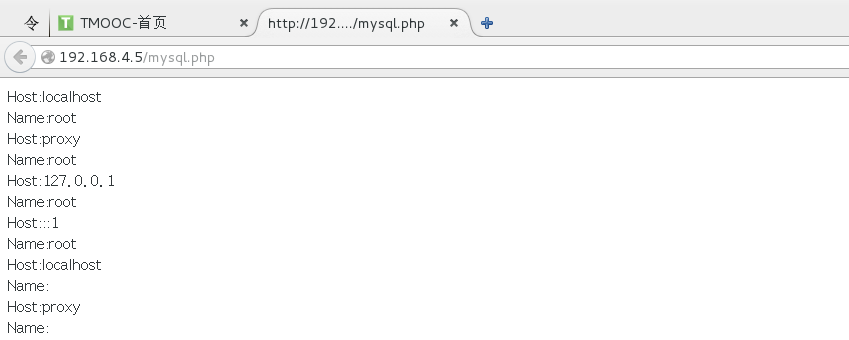
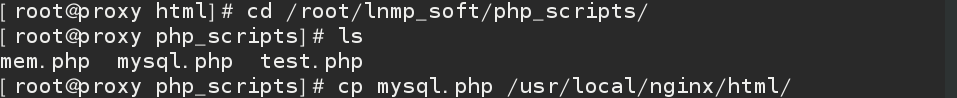
**#加载其他文件**



**#将脚本放到对应目录下**

**#检验**

1. **连通mariadb**



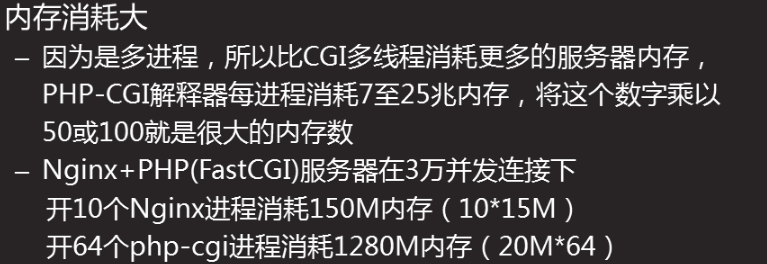
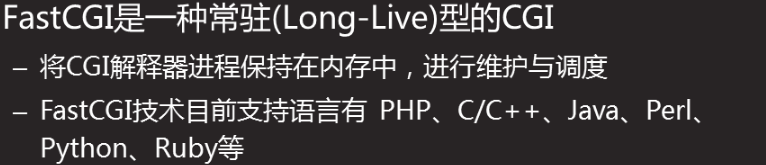
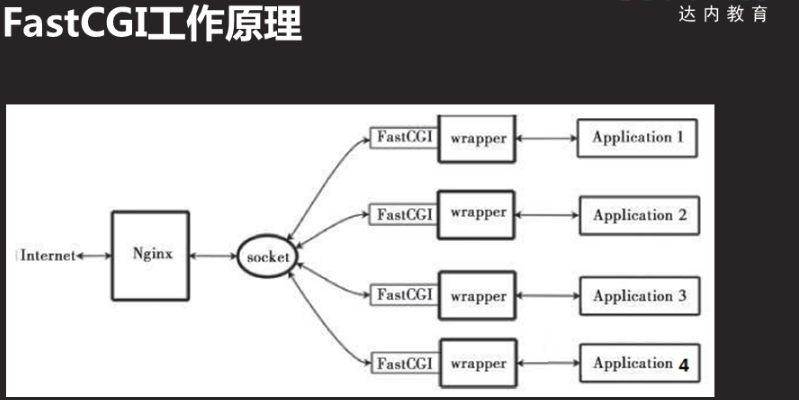
**#nginx+php通过脚本代码连通数据库**

**#将页面拷贝到网页目录下**

**#用户可以访问此页面**

**看到的不是源代码而是数据，说明连通了数据库**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FastGCI\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

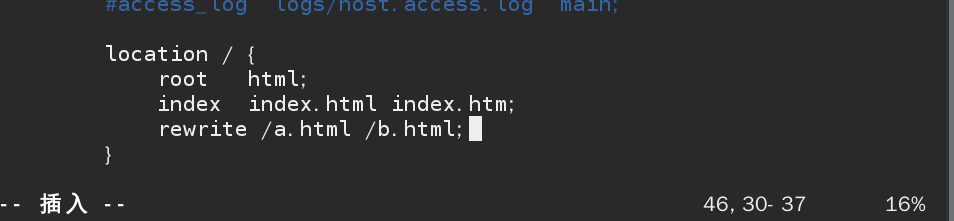


**地址重写**：地址栏被重新改写，支持正则

**rewrite 旧页面 新页面 [选项]；选项【last不再读其他rewrite；break:不再读其他语句，结束访问；redirect：临时（蜘蛛）；permamet：永久（蜘蛛）；】**

**实验1：访问a.html，跳转到b.html**





**#添加选项**

**#检验**

**#添加地址重写**

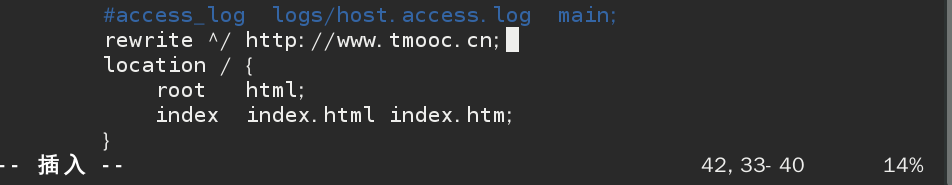
\*\*\*\*\*



**#访问a.html，地址栏会自动变成b.html**

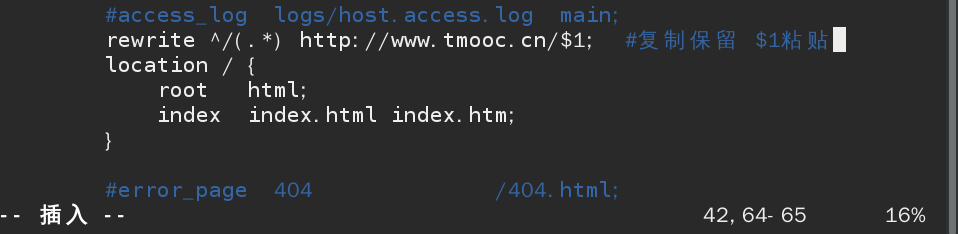
**实验2：访问192.168.4.5，跳转到www.tmooc.cn**





**实验3：访问192.168.4.5/任意页面 跳转到www.tmooc.cn/相同页面（正则保留）**





**“集群软件【有些是带健康检查(nginx)，有些不带】，如果不带健康检查，则需要写脚本来实现**

**脚本：**

**测主机：ping**

**测端口：nmap命令测试端口是否打开**

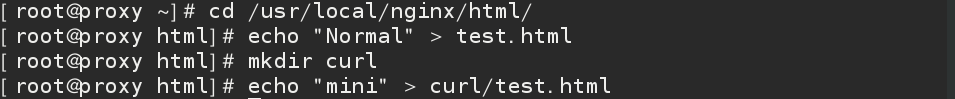
**测页面：curl http://192.168.4.5/index.html(例)测试页面是否被删除**

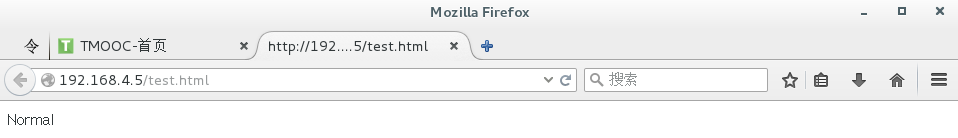
**测hash值：md5sum命令测试页面内容是否被更改”**

**实验4：不同浏览器访问相同页面，结果不同**

1. **：准备两套页面**







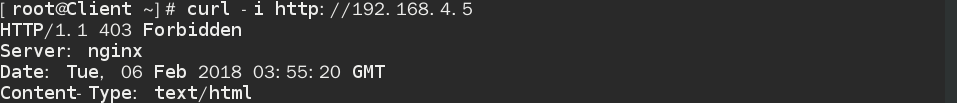
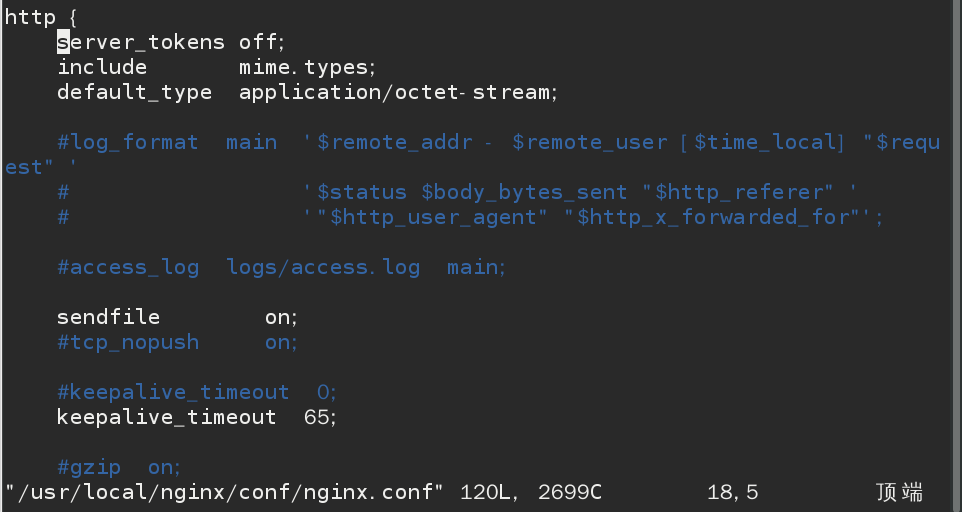
**#伪装成其他浏览器**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Nginx性能、安全、并发\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

功能：用户认证；虚拟主机（基于域名、基于IP、基于端口）；加密网站（https）；调度器；地址重写

**\*\*\*不显示nginx软件版本号**

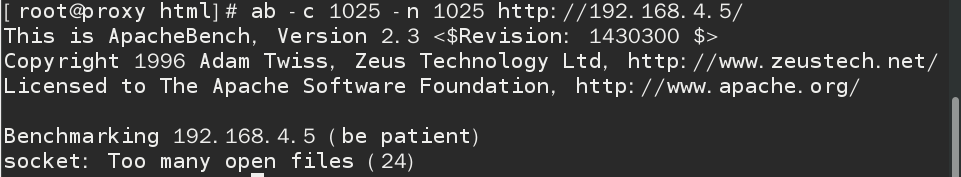
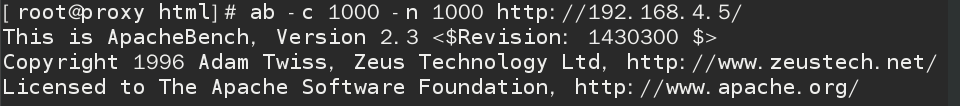




**#添加命令。不显示Nginx软件版本号**

**\*\*\*并发**

**#安装之后才会有ab软件，不会补齐/**



**#测试1025人访问**

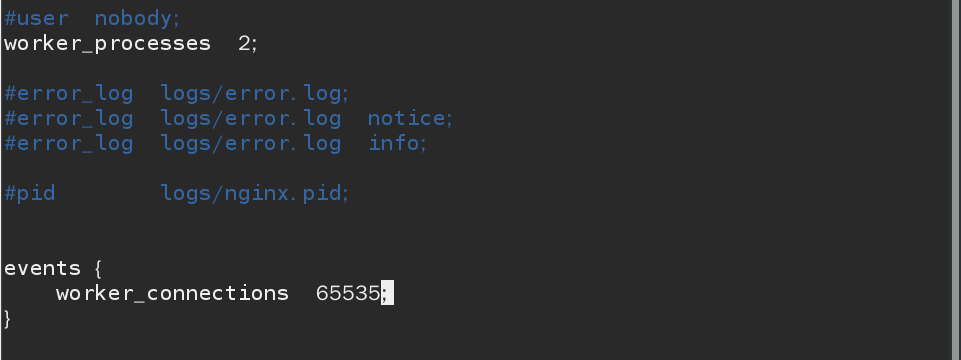
**1025次，失败，需要修改配置文件调优**

**#测试1000人访问**

**1000次，可以成功**



**#查看cpu核心数量**

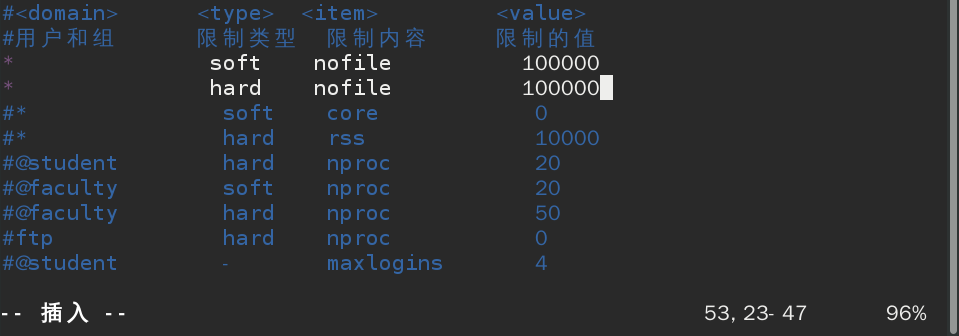
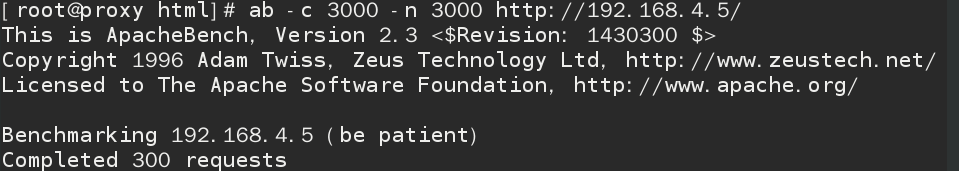
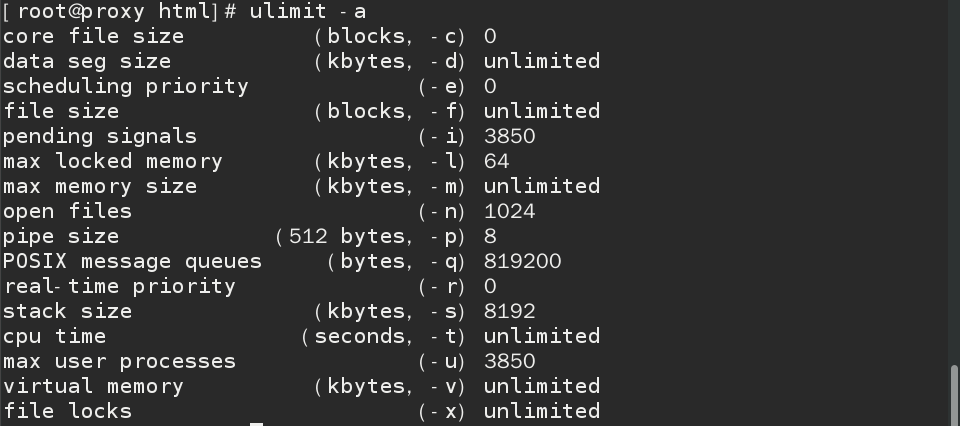


**#每个worker最大并发连接数，大于65535就可以**

**#进程数，要与cpu核心数量一致**

（Linux内核对能并发打开的文件、进程、登陆者数量、cpu数量都有限制）

**#查看**



**#打开文件的最大数量**

**#修改配置文件，重启后永久生效**

**#硬限制hard，其他人不可以修改**

**#软限制soft，其他人可以修改**

**#临时生效**

**#测试并发，成功**

**#打开文件的数量**

**\*\*\*如何解决客户端访问头部信息过长的问题**





**#客户端\_头部\_缓存\_大小**

**#大请求包头部信息的缓存个数与容量**

**#先根据client\_header\_buffer分配，如果不够，再根据large值分配（默认：4 4k）**

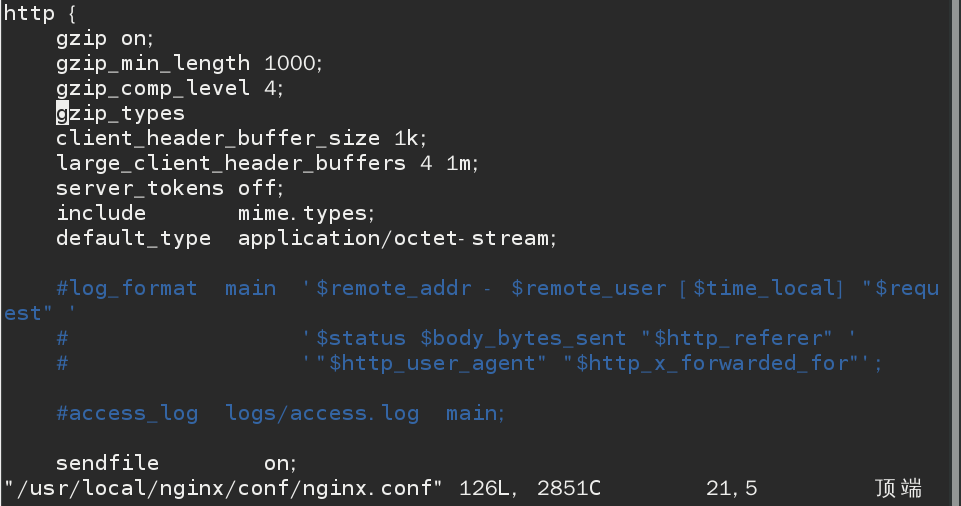
**#默认请求包头信息的缓存**

**\*\*\*gzip压缩页面**

**mp3、jpg、mp4、flv等多媒体文件不支持压缩（越压缩越大），本身就是压缩格式**

**word、xls、html、txt、css、pdf等文档文件支持压缩**

**如何压缩\对什么格式的文件压缩，参考文件(/usr/local/nginx/conf/mime.types)**

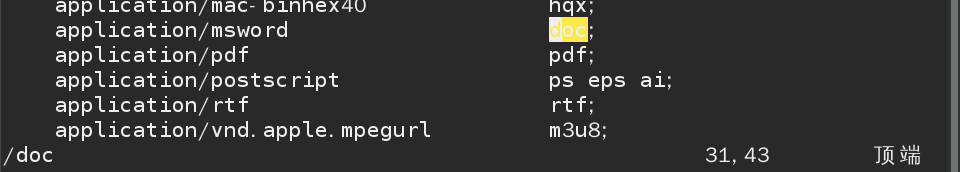


**#压缩类型**

**#压缩级别（1-9，越大压缩效果越好，cpu占用越大）**

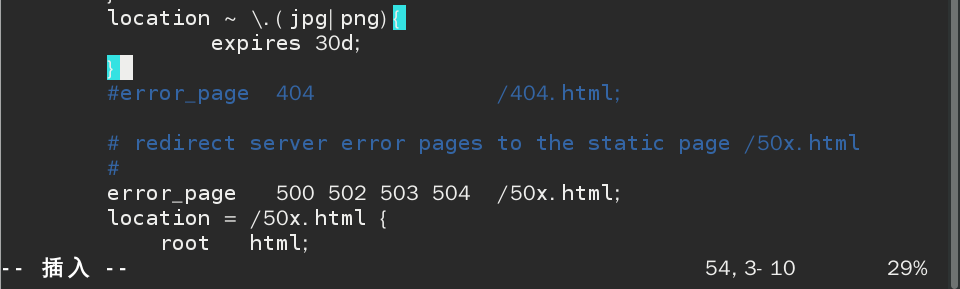
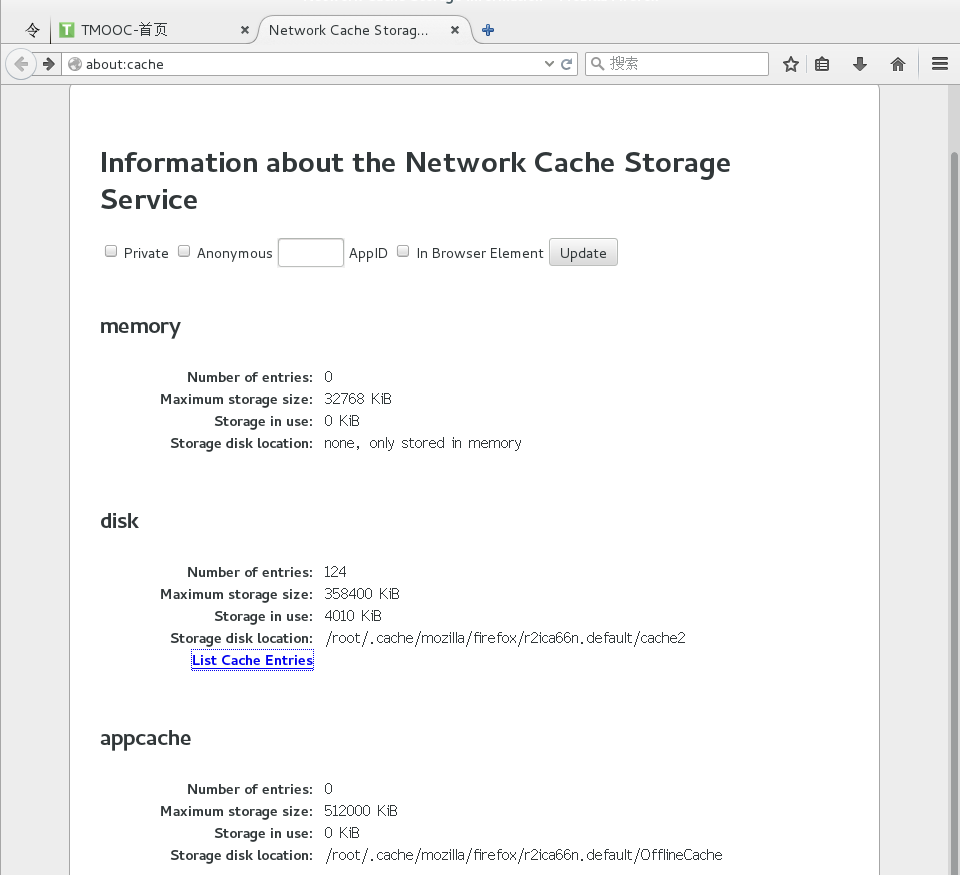
**#最小达到1000字节才压缩,小文件不压缩**

**#打开压缩功能**



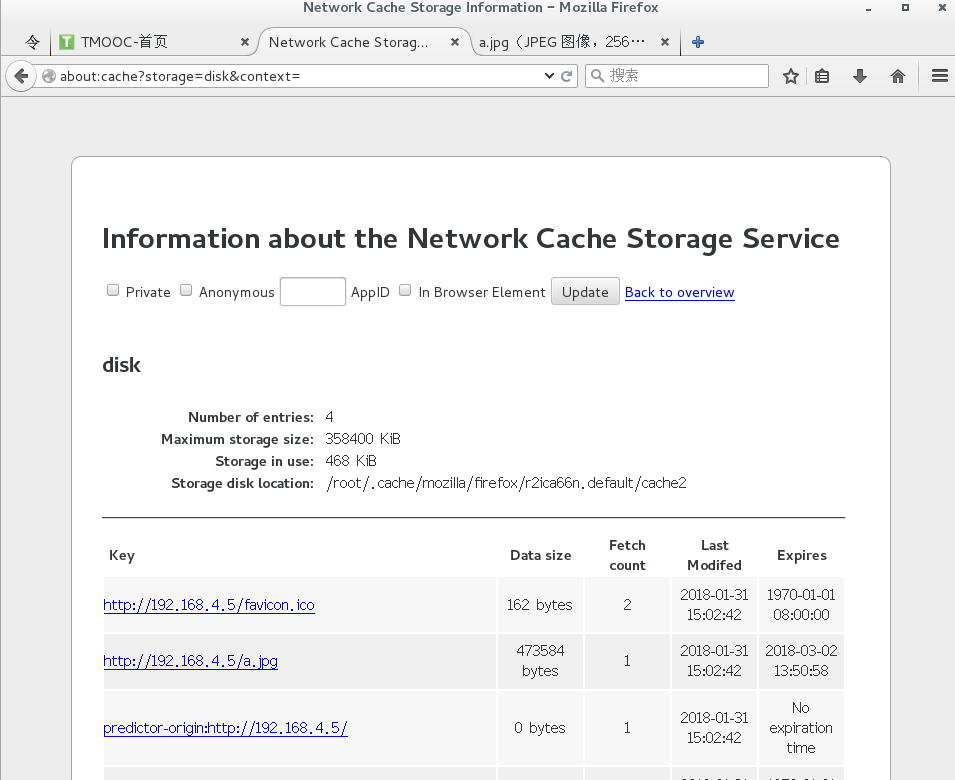
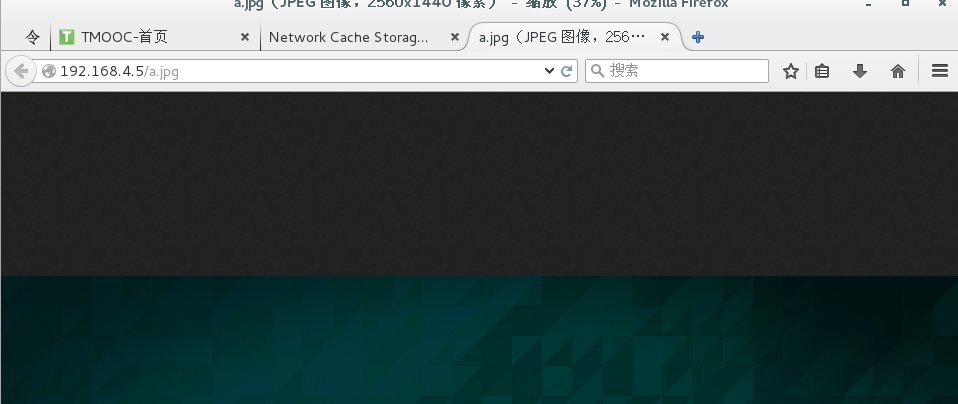
**#搜索**

**\*\*\*缓存（让用户的浏览器缓存）仅对静态数据缓存**

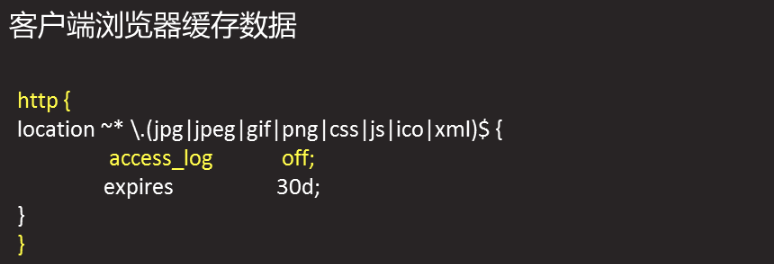
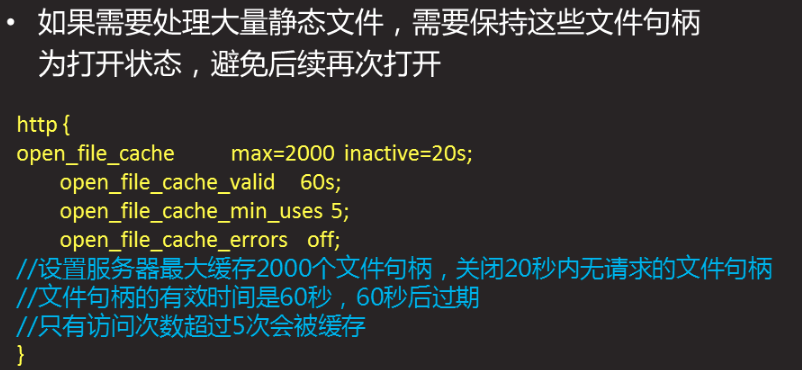
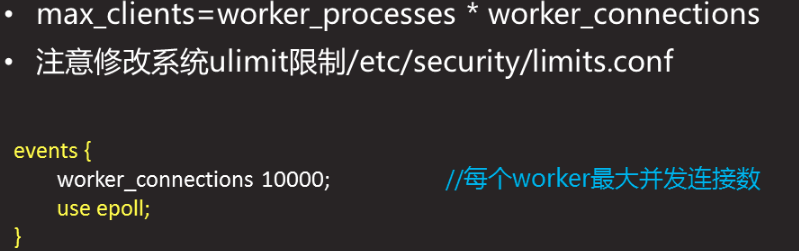
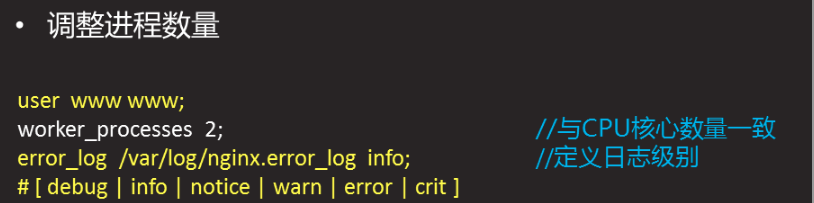


**#缓存的文件类型**

**#缓存多少天**



\*\*\*全局配置优化



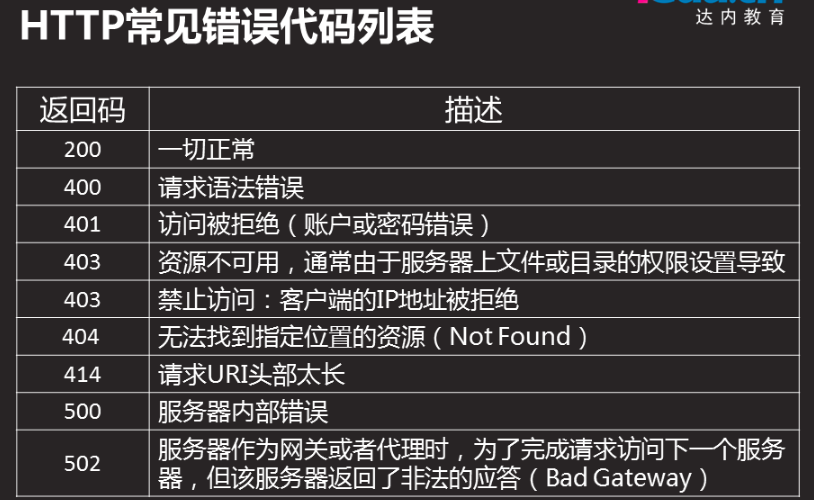
**#缓存出错不报错**

**#访问超多5次将被缓存**

**#缓存文件有效期最长60s**

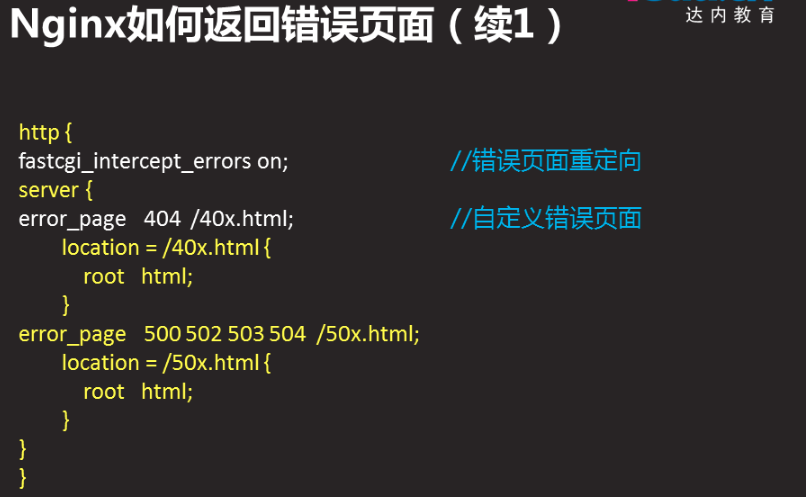
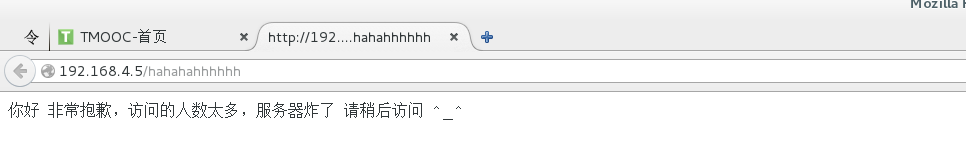
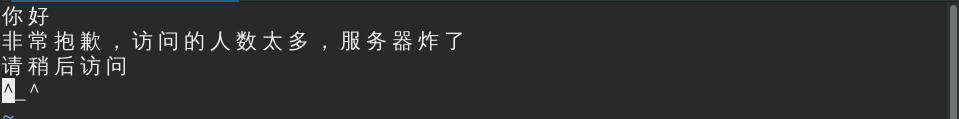
**#20s无人访问删除**

**#打开的文件加入缓存**

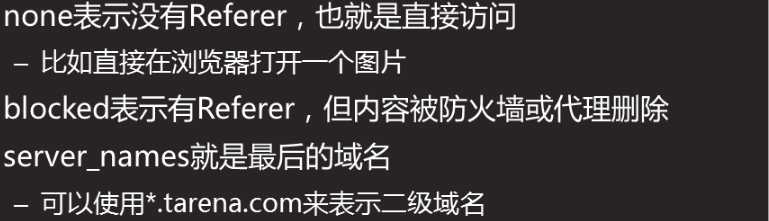
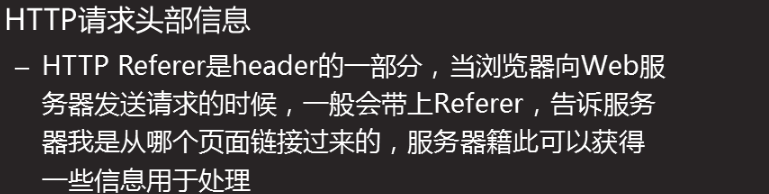
**\*\*\*自定义报错页面**

自定义报错，如404------>返回特定页面或首页





**\*\*\*防盗链**



**防盗链语句**

**referer简介**