# Nginx\_01

# 今日目标6

* 知道Nginx的作用
* 实现安装Nginx与php-fpm
* 实现配置Nginx的虚拟主机
* 了解Nginx的缓存功能
* 了解Nginx的压缩功能
* 编写出日志分割shell脚本

# 一、Nginx介绍

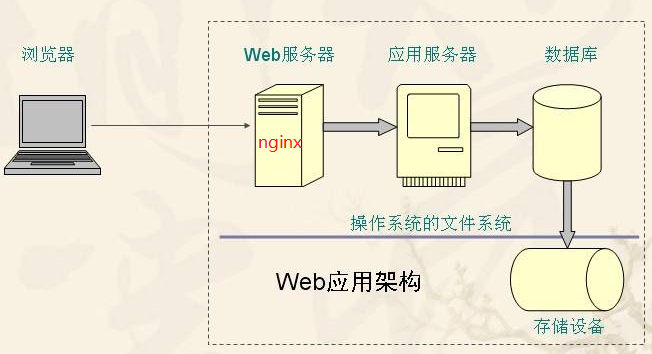
## 1.1、概述

Nginx是开源免费的一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器。其特点是占有内存少，并发能力强，使用nginx网站用户有很多，如百度、京东、新浪、网易、腾讯、淘宝等。

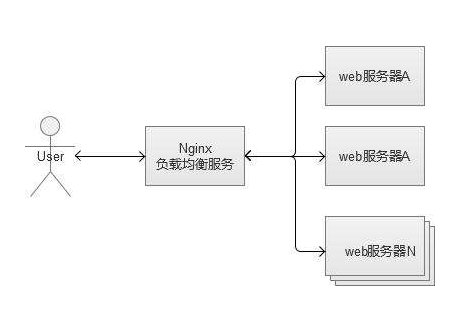
官网：<http://nginx.org/>

中文文档：<http://tengine.taobao.org/nginx_docs/cn/docs/>

web服务器

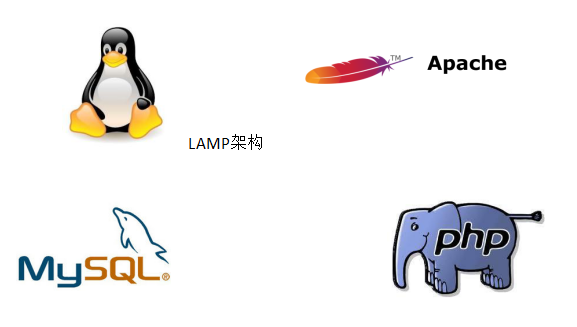


负载均衡



## 1.2、常用PHP网站架构

架构1：Linux + **Apache** +MySQL + PHP



架构2：Linux + **Nginx** + MySQL + PHP 大并发



**Nginx的优点**

* **可以高并发连接**

官方测试Nginx能够支撑5万并发连接，实际生产环境中可以支撑2~4万并发连接数。

原因，主要是Nginx使用了最新的epoll（Linux2.6内核）网络I/O模型，而Apache使用的是传统的Select模型，其比较稳定的Prefork模式为多进程模式，需要经常派生子进程，所以消耗的CPU等服务器资源，要比Nginx高很多。

* **内存消耗少**

Nginx+PHP（FastCGI）服务器，在3万并发连接下，开启10个Nginx进程消耗150MB内存，15MB\*10=150MB，开启的64个PHP-CGI进程消耗1280内存，20MB\*64=1280MB，加上系统自身消耗的内存，总共消耗不到2GB的内存。

* **成本低廉**

购买负载均衡交换机或相关的云负载均衡服务(SLB)都是要花费很高昂的费用的，而Nginx为开源软件，可以免费用，并且可用于商业用途。

* **配置文件非常简单**
* **稳定性高**
* **支持热部署**
* **节省带宽**



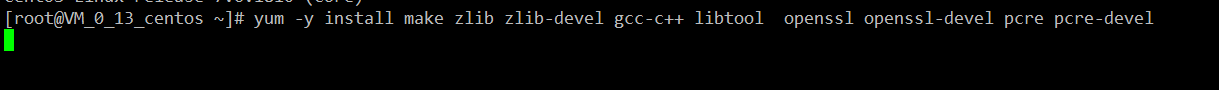
# 二、Nginx的安装与配置

安装nginx环境是在linux下面(centos7.6)

cat /etc/redhat-release

## 2.1、安装依赖

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



## 2.2、安装nginx

安装步骤：

* 下载源码包
* 解压并进入解压目录
* 配置
* 编译和安装

wget http://nginx.org/download/nginx-1.16.1.tar.gz

tar zxf nginx-1.16.1.tar.gz

cd nginx-1.16.1

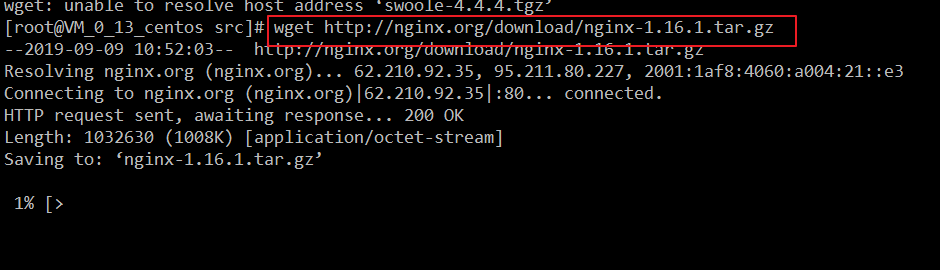
./configure --prefix=/usr/local/nginx --with-http\_stub\_status\_module

make && make install

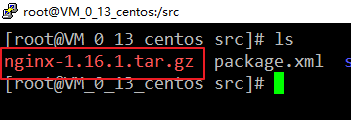
进入nginx官网中，去下载最新的稳定版本

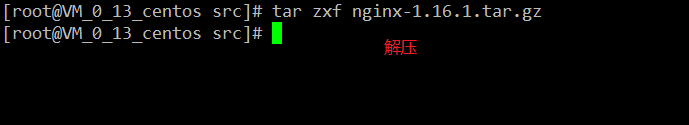


在linux下，使用wget来下载源码包

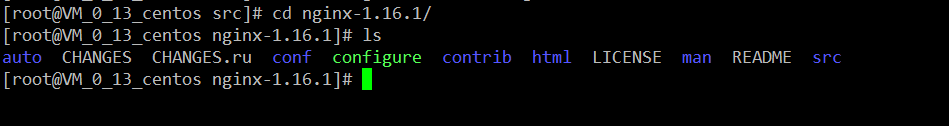


下载完毕后，解压源码包

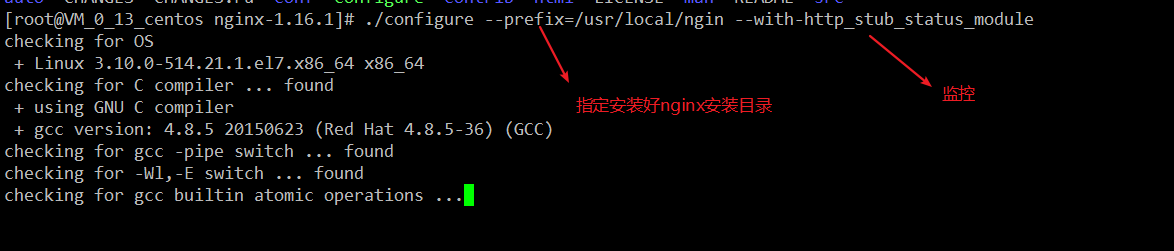


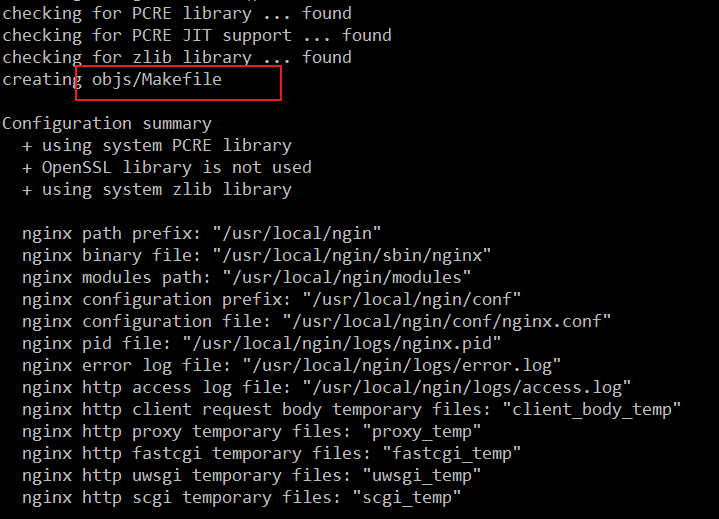


解压完毕后，进入到目录中



使用configure命令来对nginx安装源码检查一下环境

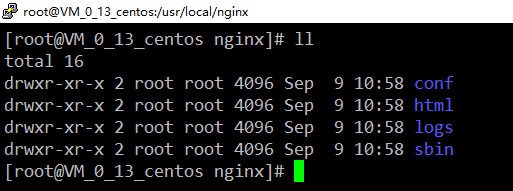




检查没有报错，就可以执行安装

make && make install

注：安装时指定了安装目录 /usr/local/nginx



## 2.3、目录介绍

安装成功后，进入到安装后的目录中

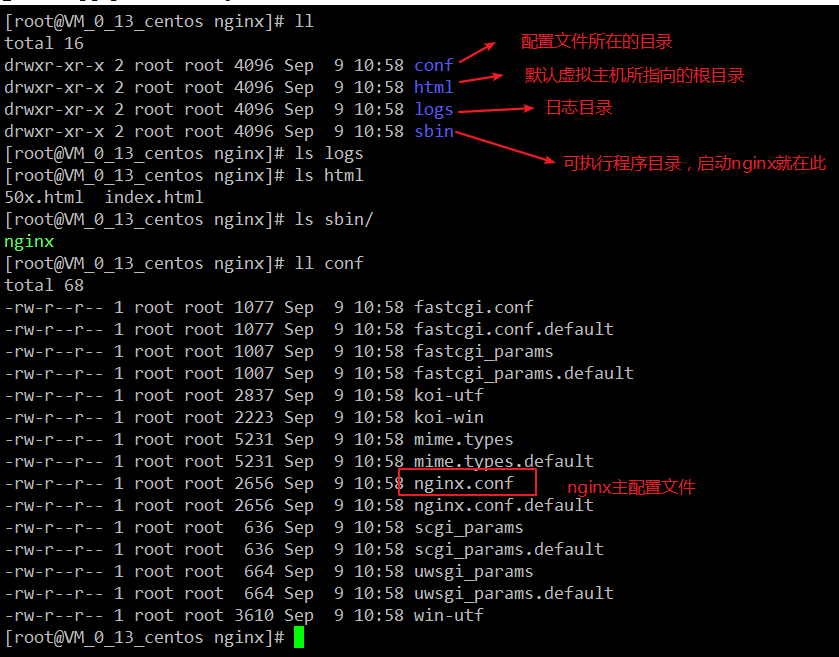
cd /usr/local/nginx, 看到如下4个目录

conf 配置文件 配置虚拟主机文件就在此

html 网页文件

logs 日志文件

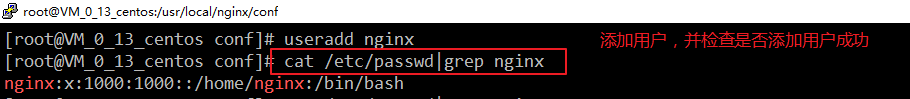
sbin 主要二进制程序 启动服务命令就在此

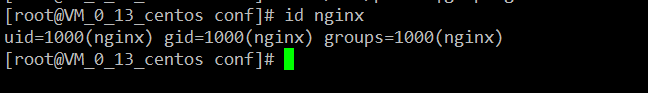


## 2.3、修改nginx运行系统用户

创建 Nginx 运行使用的用户nginx

useradd nginx



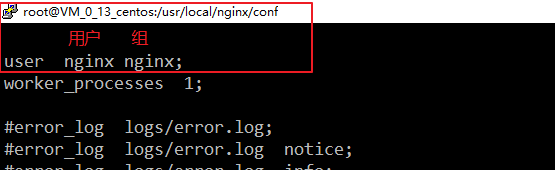


修改nginx.conf配置文件

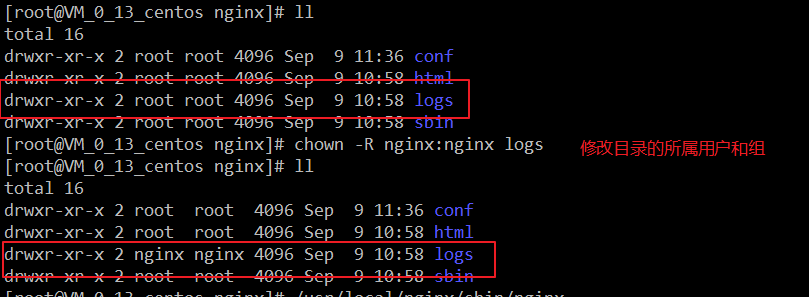
vi nginx.conf

// 用户 组

user nginx nginx;



修改日志目录用户和组



## 2.4、启动nginx

# 启动 nginx 服务

/usr/local/nginx/sbin/nginx

注：启动时一定要确保80端口没有被占用

netstat -tunpl

/usr/local/nginx/sbin/nginx -s reload # 重新载入配置文件

/usr/local/nginx/sbin/nginx -s stop # 停止 Nginx

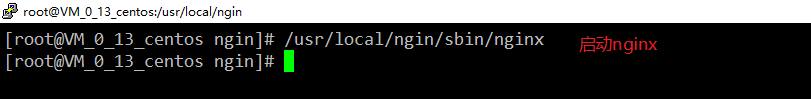
# 设置开机启动

vi /etc/rc.local

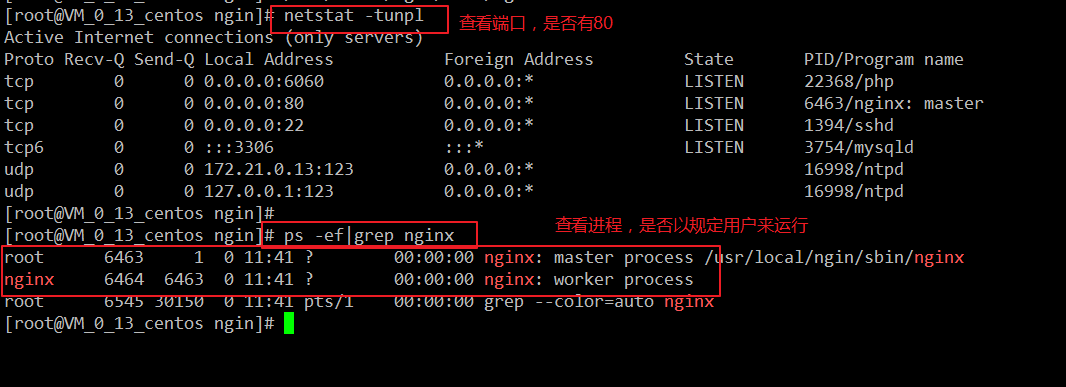
# 添加如下命令

/usr/local/nginx/sbin/nginx

启动nginx

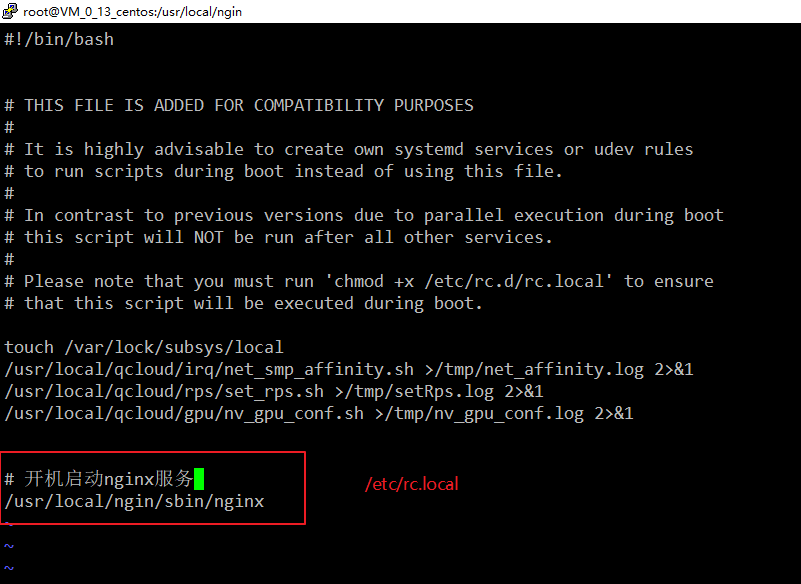


检查nginx是否正常启动

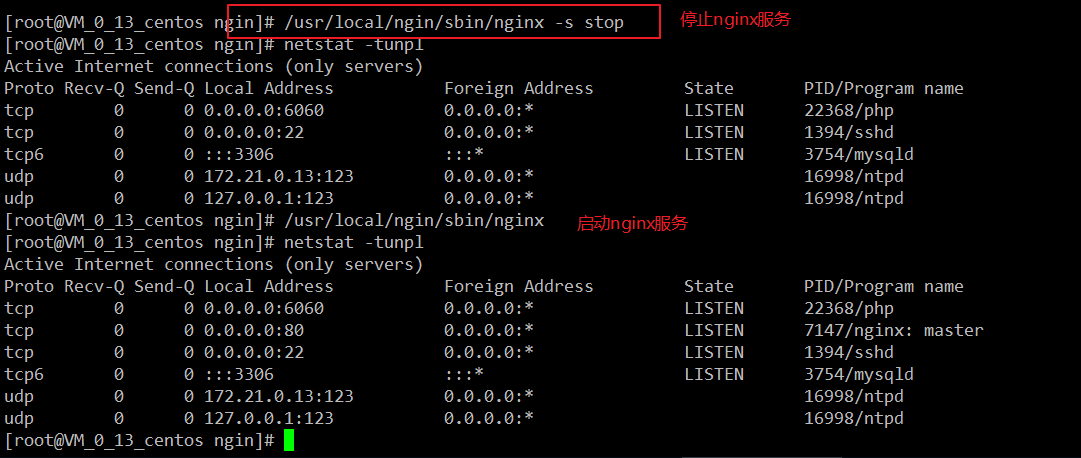


开机自启服务

vi /etc/rc.local



停止nginx服务



## 2.5、配置文件

配置文件默认为安装目录中的conf/nginx.conf文件。

配置文件语法：http段 => server段 => location段

http段：是要包含所有的Server段，即将nginx所有的虚拟主机配置包含进去

server段：一个server即一个虚拟主机配置段落，后期有多少个虚拟主机取决于server段

location段：每个location段即表示该虚拟主机中的一个模块功能配置

注：配置后面分号不能少，{} 和配置之间要有空格



# 三、web服务器

## 3.1、配置虚拟主机

一台服务器上面，有时候会绑定多个不同的域名，这时就需要我们来配置nginx让其支持多域名的虚拟主机。

步骤：

* 在nginx配置文件中新增一个server节点
* 在server节点中，修改对应的站点目录
* 重载nginx服务

server {

listen 80; # 端口

server\_name www.aa.com; # 域名

root /var/www/html; # 根目录

location / {

index index.html; # 默认缺省文件

}

}

先注册一域名，然后备案。

进入域名管理后台，进行域名解析到指定的ip主机上

A记录 ： 域名指向到IP

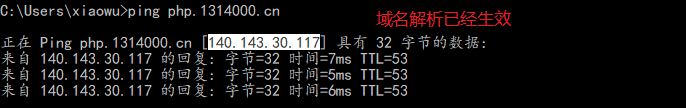
CNAME ： 域名指向到域名 (直播、CDN)

解析一个A记录到指定服务器上





检查解析是否生效



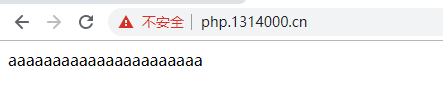
在服务器中配置虚拟主机，让域名指向到对应的项目中



配置完毕后，需要重载nginx服务

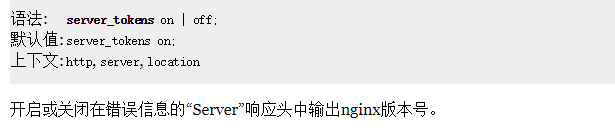
/usr/local/nginx/sbin/nginx -s reload

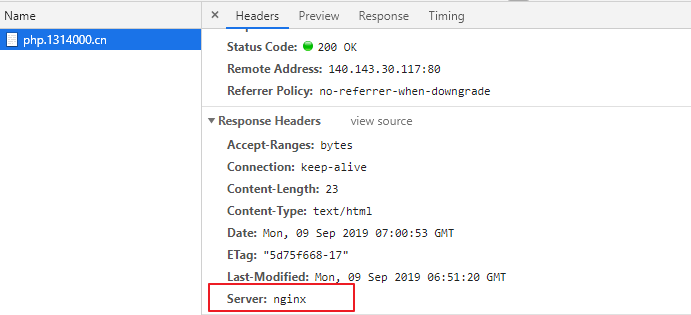
访问



开启或关闭在错误信息的“Server”响应头中输出nginx版本号

server\_tokens off;





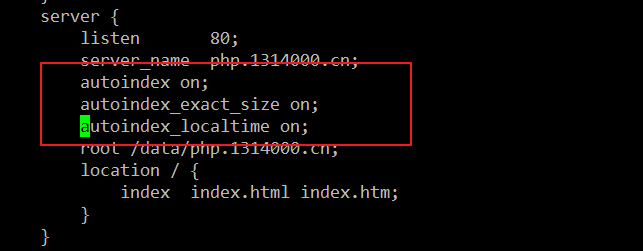
开启列出网站目录列表，建议在生产环境中不要开启，太危险。

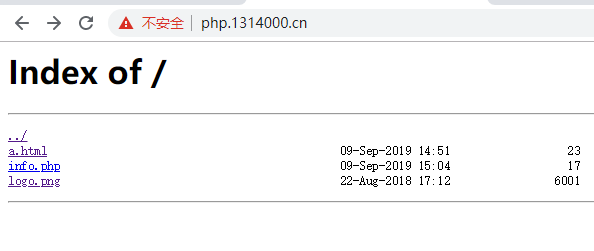
autoindex on;

autoindex\_exact\_size on;

autoindex\_localtime on;







配置文件分目录和文件，使用include来包含，支持模糊指向(\*.conf)

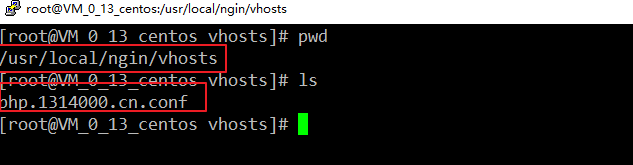
include /usr/local/nginx/vhosts/\*.conf

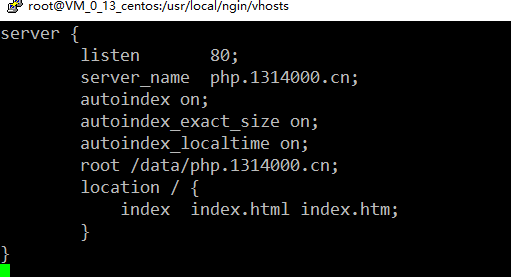
# 重载

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



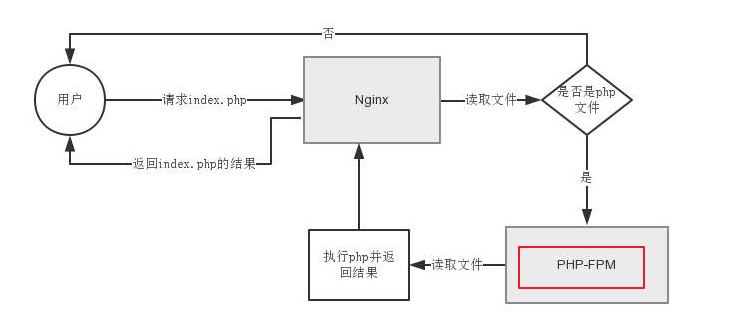
创建对应的目录和添加对应配置文件





## 3.2、支持php

apache因为把php模块集于一身成为一个整体，任何请求访问（静态，动态）都是apache（httpd）自己搞定的， 而nginx仅仅是一个传递的角色。它可以处理静态（html,图片，js，css等）,但php相关的就得由后面的php-fpm来处理。



* 安装php-fpm和启动

# 搜索

yum search fpm|grep ^php

# 安装

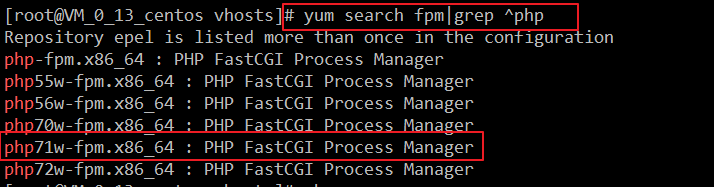
yum install -y php71w-fpm.x86\_64

# 启动

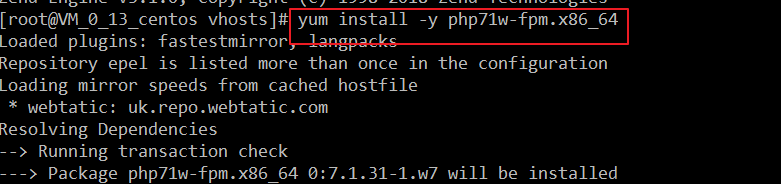
/usr/sbin/php-fpm

默认端口号为 9000

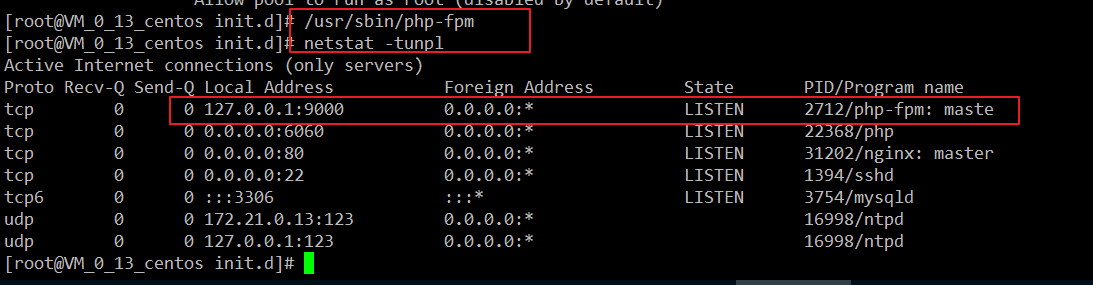
查找php-fpm软件名称



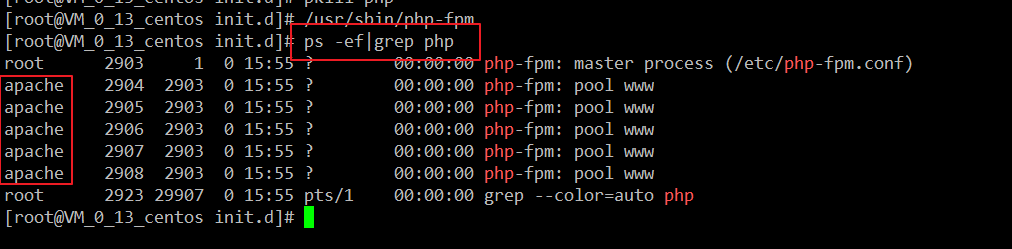
安装



启动php-fpm服务，默认的端口号为9000



查看进程



* 配置nginx支持php

# http节点

# 指定连接到后端Fastcgi的超时时间(单位:秒)

fastcgi\_connect\_timeout 300;

# 向Fastcgi传送请求的超时时间,这个值是指已经完成两次握手后向Fastcgi传送请求的超时时间(单位:秒)

fastcgi\_send\_timeout 300;

# 指定接收Fastcgi应答的超时时间,这个值是指已经完成两次握手后接收Fastcgi应答超时时间(单位:秒)

fastcgi\_read\_timeout 300;

# server节点

location ~ \.php$ {

fastcgi\_pass 127.0.0.1:9000;

# 非必须 默认索引页

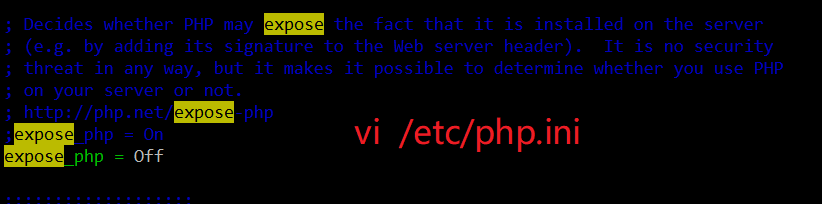
#fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

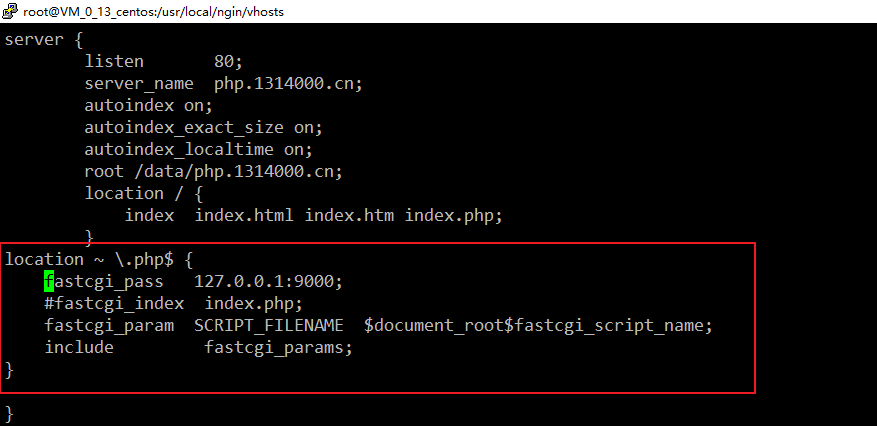
include fastcgi\_params;

}

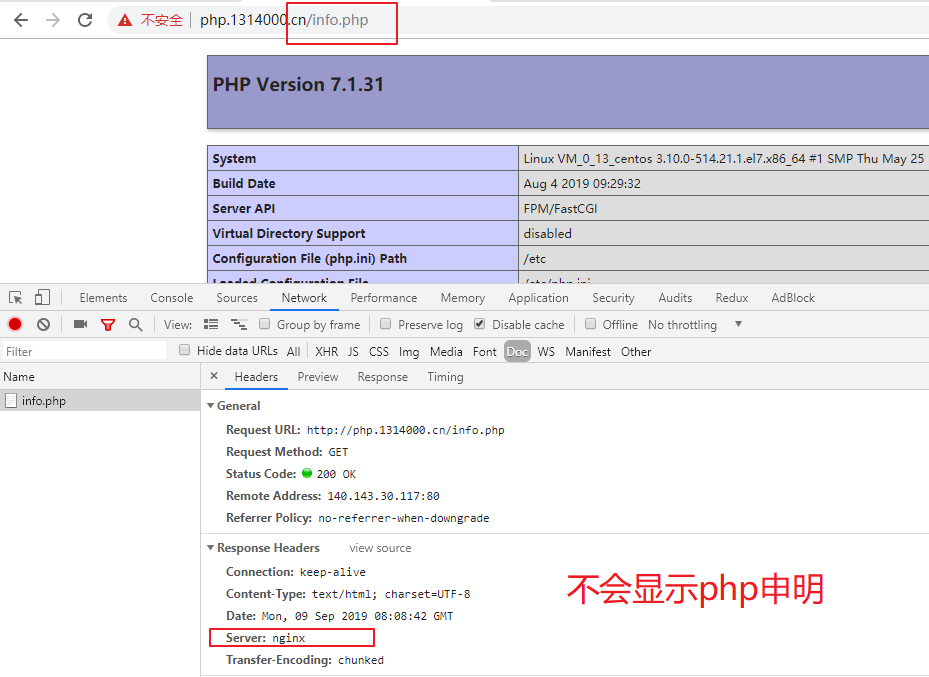
隐藏php申明



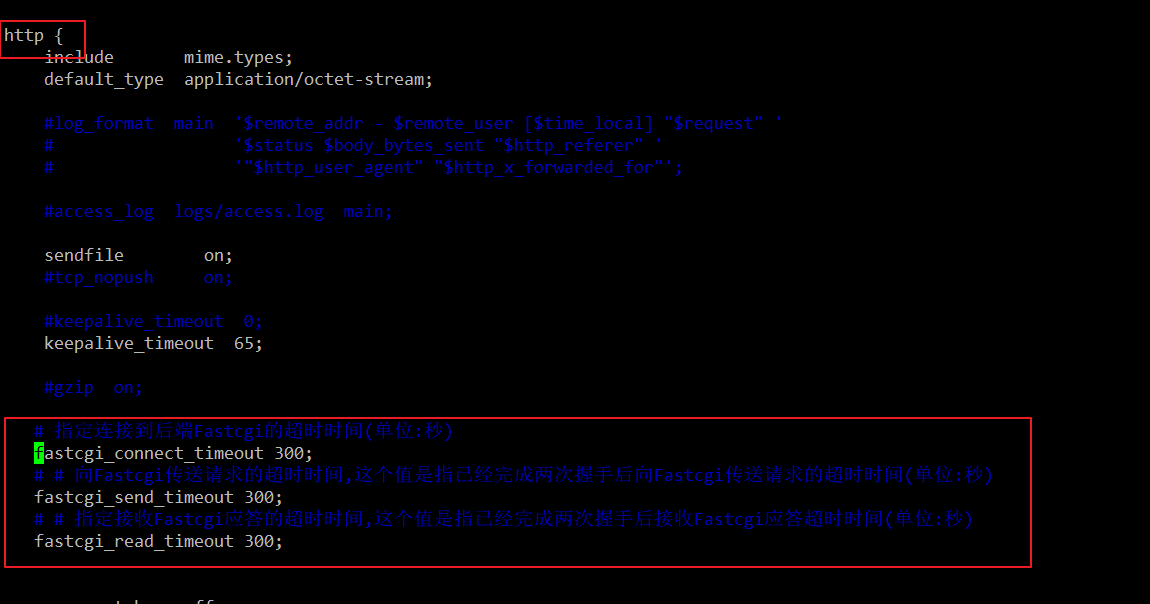
配置nginx虚拟主机支持php



访问



设置nginx与php-fpm连接超时时间



## 3.3、开启gzip

压缩文件大小变小了，传输更快了，提高网站的访问速度和并发量。目前市场上浏览器都是支持GZIP的。IE6以下支持不好，会出现乱码情况。

**操作步骤：**

* 修改配置，开启gzip压缩
* 重载nginx配置
* 测试是否有压缩

在线压缩测试工具<http://tool.chinaz.com/Gzips/>

## http 节点中设置

# 开启gzip

gzip on;

# 设置将被gzip压缩的响应的最小长度

gzip\_min\_length 200;

# 压缩级别 1-9

gzip\_comp\_level 6;

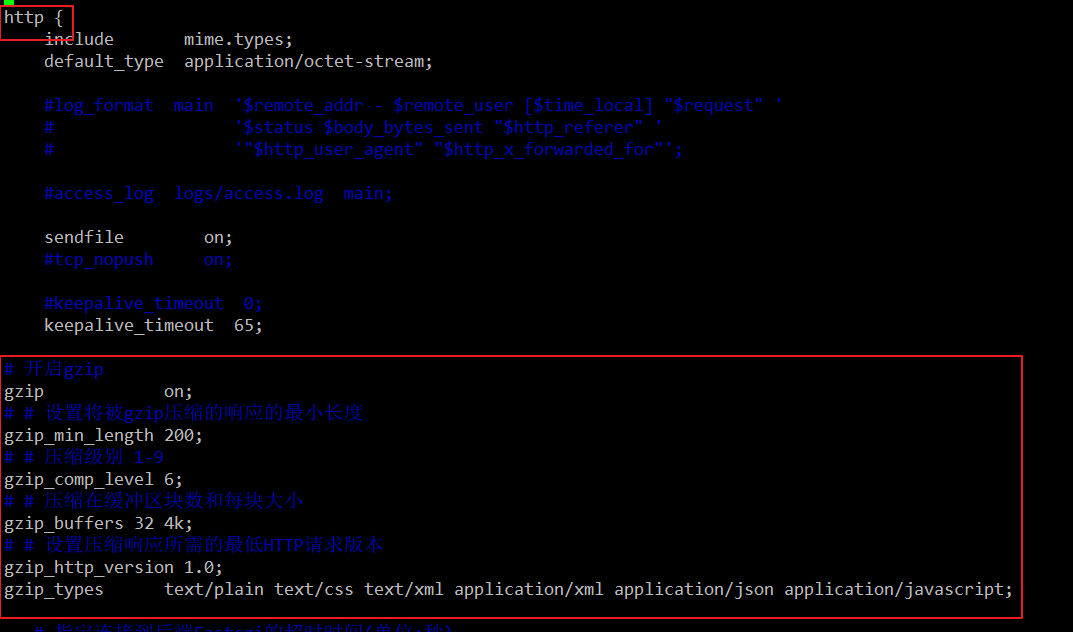
# 压缩在缓冲区块数和每块大小

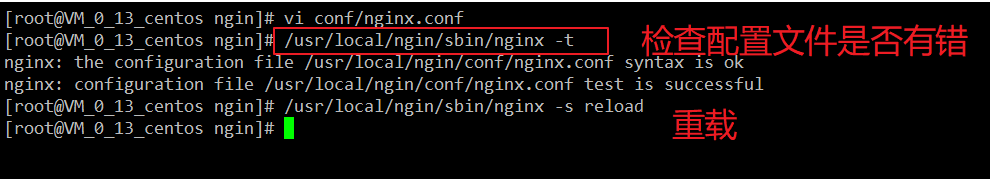
gzip\_buffers 32 4k; # | 16 8k;

# 设置压缩响应所需的最低HTTP请求版本

gzip\_http\_version 1.0;

gzip\_types text/plain text/html text/css text/xml application/xml application/json application/javascript;





实测



## 3.4、浏览器缓存

告知浏览器获取的信息是在某个区间时间段是有效的。在有效期内，不需要到服务器重新获取了,提高访问速度，节省服务器带宽资源。

## 设置在 server 中的 location 节点

# expires 1s 秒

# expires 2m 分

# expires 3h 时

# expires 6d 天

# ~\* 不区分大小写

location ~\* \.(jpg|jpeg|gif|bmp|png|ico|svg|mp4|mp3){

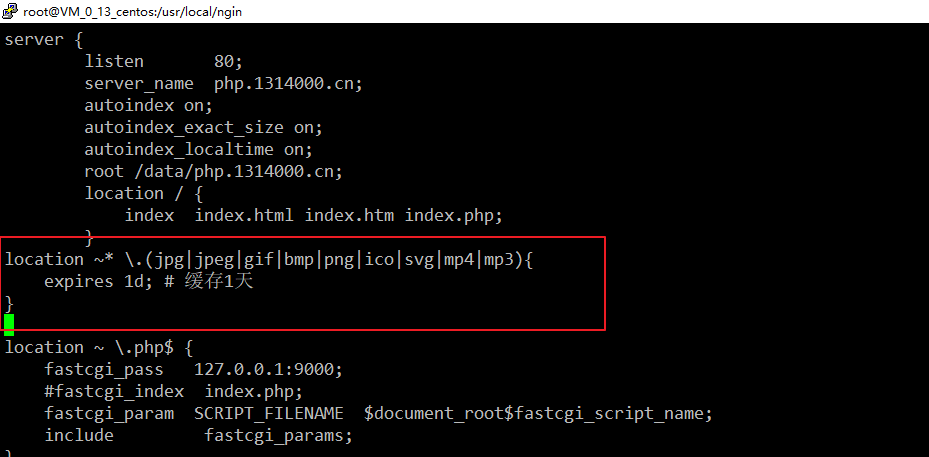
expires 7d; # 缓存1天

}

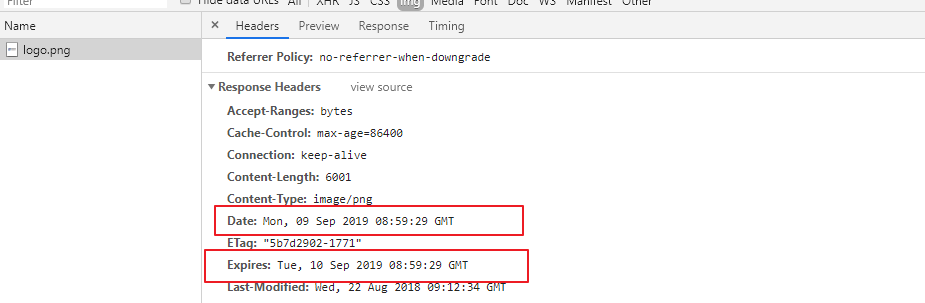
location ~\* \.(css|js){

expires 1d; # 缓存1天

}



实测



## 3.5、图片防盗链

网站中的很多地方都是使用到了图片，如商城中商品中就使用了大量的图片，而为了不使自己辛辛苦苦拍摄的图片被别人调用，同时别人调用也会对服务器产生压力和流量费用，这时就需要防盗链的功能。

location ~\* \.(jpg|jpeg|gif|bmp|png|ico|svg|mp4|mp3)$ {

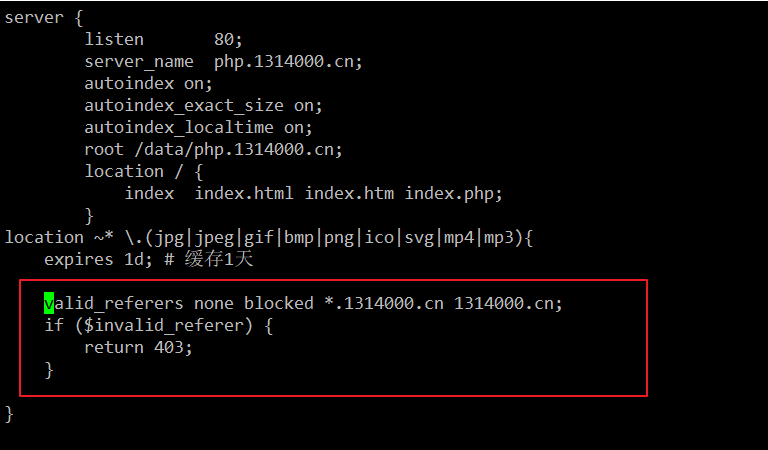
valid\_referers none blocked \*.1314000.cn 1314000.cn;

if ($invalid\_referer) {

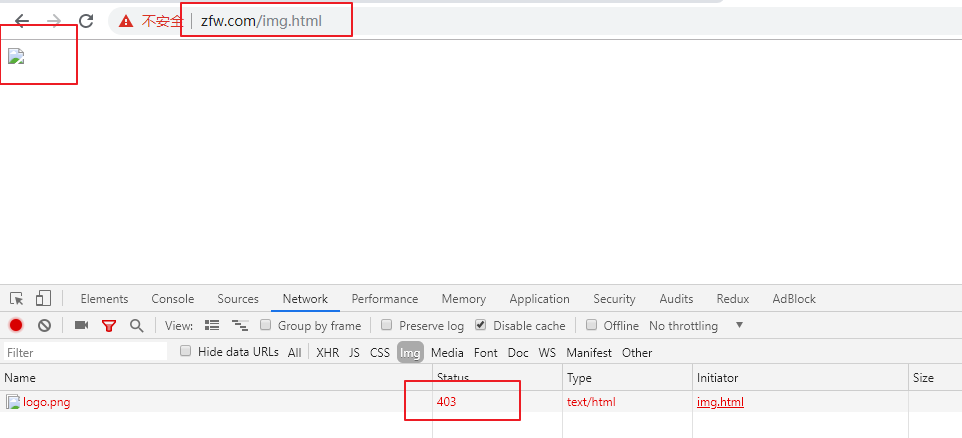
return 403;

}

}



实测

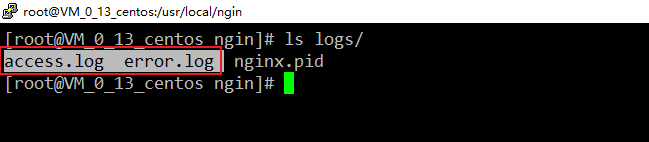


# 四、日志管理

## 4.1、日志记录

Nginx里默认会有两种日志 access.log 成功访问日志 error.log 错误日志

默认路径：nginx安装目录下的logs目录



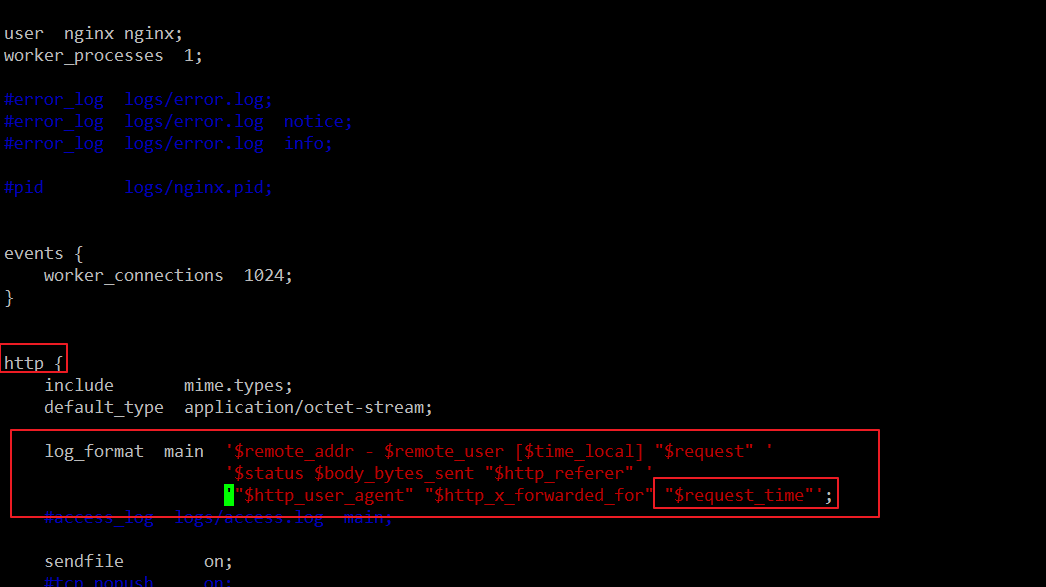
通过配置虚拟主机可以指定日志目录和文件名称

access\_log logs/access.log main; # 日志

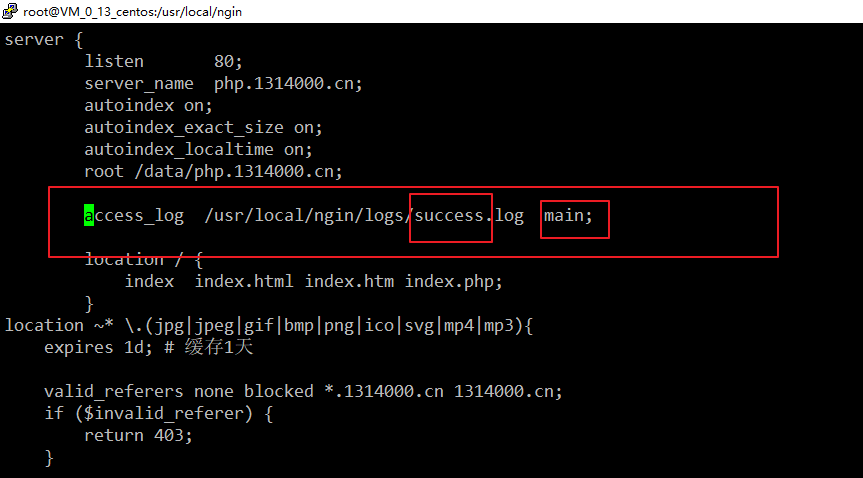
日志格式中变量:

|  |  |
| --- | --- |
| $remote\_addr | 客户端的ip地址(代理服务器，显示代理服务ip) |
| $remote\_user | 用于记录远程客户端的用户名称（一般为“-”） |
| $time\_local | 用于记录访问时间和时区 |
| $request | 用于记录请求的url以及请求方法 |
| $status | 响应状态码，例如：200成功、404页面找不到等。 |
| $body\_bytes\_sent | 给客户端发送的文件主体内容字节数 |
| $http\_user\_agent | 用户所使用的代理（一般为浏览器） |
| $http\_x\_forwarded\_for | 可以记录客户端IP，通过代理服务器来记录客户端的ip地址 |
| $http\_referer | 可以记录用户是从哪个链接访问过来的 |
| $request\_length | 请求长度（包括请求行，标题和请求体） |
| $request\_method | 请求方法，通常是“ GET”或“ POST” |
| $request\_time | 请求处理时间 |
| $uri | 请求中的当前URI |

主配置文件中打开main配置，并添加请求处理时间 vi /usr/local/nginx/conf/nginx.conf

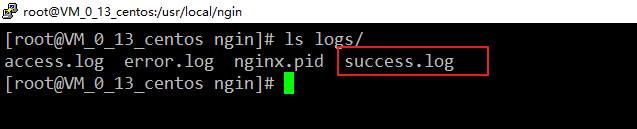


在虚拟主机中指定日志所用的格式和文件名称



重载一次nginx

查看日志生成情况





## 4.2、日志分割

nginx运行时会产生大量的日志，为了方便查询日志，所以有是时候需要将日志文件进行分割。

#!/bin/bash

log\_home="/usr/local/nginx/logs"

log\_path=${log\_home}/$(date +"%Y")/$(date +"%m")

mkdir -p $log\_path

mv $log\_home/access.log $log\_path/access\_$(date -d yesterday +"%Y%m%d").log

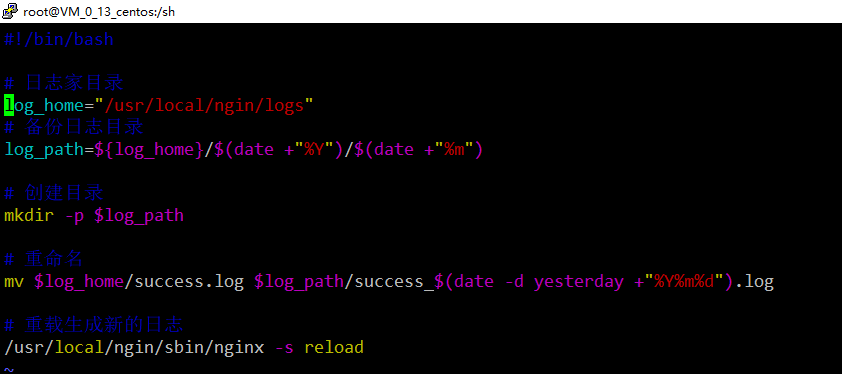
/usr/local/ngin/sbin/nginx -s reload

在linux创建一个专门存放shell脚本的目录

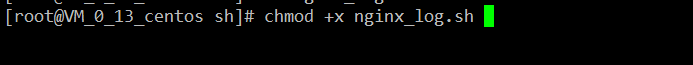


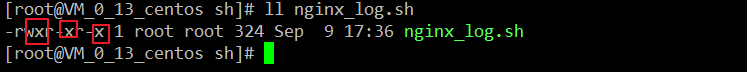
新建一个shell文件



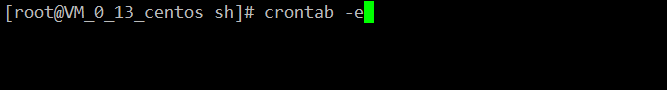


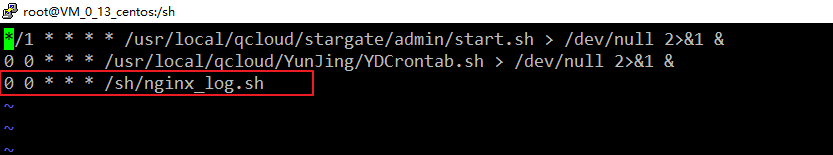
给shell脚本添加可执行权限





把此脚本添加到计划任务





此时脚本每天的零晨就会自动执行一次，实现了日志按天分割功能。

