



链滴

# Centos 【安装 nginx 及基本配置】

作者: [haxLook](#)

原文链接: <https://ld246.com/article/1617154361712>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## Nginx的介绍

Nginx作为负载均衡服务：Nginx 既可以在内部直接支持 Rails 和 PHP 程序对外进行服务，也可以支持作为 HTTP代理服务对外进行服务。Nginx采用C进行编写，不论是系统资源开销还是CPU使用效率都比 Perlbal 要好很多。

由于Nginx采用C进行编写,所有在liunx上面安装时,c的编译环境是必须的。

## 插件环境

### gcc

- 查看版本

gcc -v

- 安装命令

yum -y install gcc

```
| hax_alEms
Last login: Tue Sep  1 17:52:47 2020 from 113.111.81.175
welcome to Alibaba Cloud Elastic Compute Service !

[root@izbp18a99h8g6w1p215asvz ~]# gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Target: x86_64-redhat-linux
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --enable-shared --enable-threads=posix --enable-checking=release --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-languages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initfini-array --disable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Thread model: posix
gcc version 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
[root@izbp18a99h8g6w1p215asvz ~]#
```

我使用的是阿里云的ESC云服务(centos7版本),默认是安装gcc环境.

## pcre,pcre-devel

pcre是一个perl库,包括perl兼容的正则表达式库,nginx的http模块使用pcre来解析正则表达式,以需要安装pcre库。

安装命令

```
yum install -y pcre pcre-devel
```

## zlib

zlib库提供了很多种压缩和解压缩方式nginx使用zlib对http包的内容进行gzip,所以需要安装

安装命令:

```
yum install -y zlib zlib-devel
```

## openssl

openssl是web安全通信的基石。

安装命令

```
yum install -y openssl openssl-devel
```

## nginx安装

linux下载方式,也可以通过自行下载nginx的包,进行手动下载

```
wget http://nginx.org/download/nginx-1.9.9.tar.gz
```

把压缩包解压到usr/local/src/java

```
tar -zxvf nginx-1.9.9.tar.gz
```

进入到 /usr/local/src/java/nginx-1.9.9/ 执行下面3个命令操作

```
cd /usr/local/src/java/nginx-1.9.9/
```

```
./configure
```

```
make
```

```
make install
```

安装完成以后,nginx会自动的安装到 /usr/local/nginx

一般只会出现 conf,html,logs,sbin,这4个目录结构,我这里是因为我已经配置了nginx,并且刷新了产生文件.

```
[root@izbp18a99h8g6w1p215asvz ~]# cd /usr/local/nginx/
[root@izbp18a99h8g6w1p215asvz nginx]# ll
total 36
drwx----- 2 nobody root 4096 Aug 31 23:19 client_body_temp
drwxr-xr-x 2 root root 4096 Sep 1 17:58 conf
drwx----- 2 nobody root 4096 Aug 31 23:19 fastcgi_temp
drwxr-xr-x 2 root root 4096 Aug 31 23:02 html
drwxr-xr-x 2 root root 4096 Aug 31 23:19 logs
drwx----- 12 nobody root 4096 Sep 1 00:57 proxy_temp
drwxr-xr-x 2 root root 4096 Aug 31 23:02 sbin
drwx----- 2 nobody root 4096 Aug 31 23:19 scgi_temp
drwx----- 2 nobody root 4096 Aug 31 23:19 uwsgi_temp
[root@izbp18a99h8g6w1p215asvz nginx]#
```

进行conf文件

```
cd /usr/local/nginx/conf
```

```
vim nginx.conf
```

根据自己的配置,进行nginx的反向代理服务器的配置

下面是我搭建solo的配置信息,server\_name这里填写的域名,我的域名还没有备案通过,所以 们懂的...

```
upstream backend {
    server localhost:8080; # Solo 监听端口
}

server {
    listen 80;
    server_name 47.110.48.1;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        proxy_pass http://backend$request_uri;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        client_max_body_size 10m;
        root html;
        index index.html index.htm;
    }
}
```

## 启动nginx服务

切换目录到/usr/local/nginx/sbin下

```
cd /usr/local/nginx/sbin
```

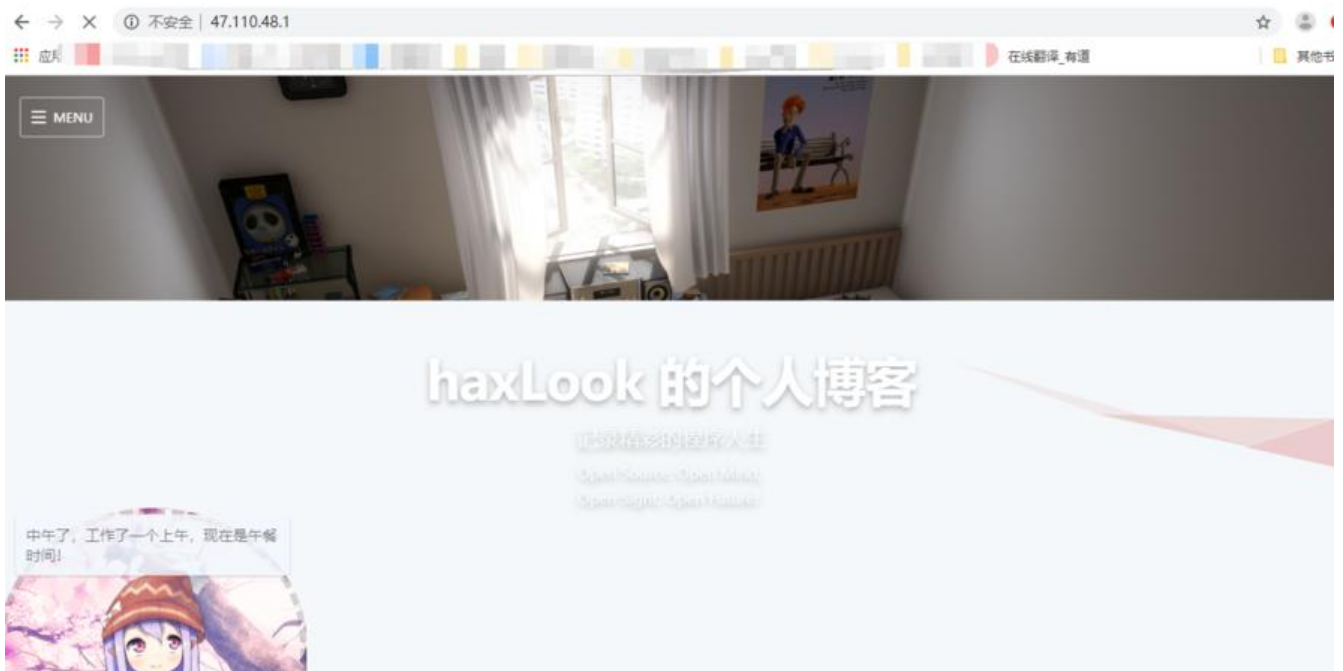
```
./nginx
```

查看nginx服务是否启动成功

```
ps -ef | grep nginx
```

```
"conf/nginx.conf" 128L, 2940C written
[root@izbp18a99h8g6w1p215asvz nginx]# ps -ef |grep nginx
root      2156      1    0 Aug31   ?        00:00:00 nginx: master process ./nginx
nobody    17086    2156  0 Sep01   ?        00:00:00 nginx: worker process
root      18879    18855  0 13:19 pts/2    00:00:00 grep --color=auto nginx
[root@izbp18a99h8g6w1p215asvz nginx]#
```

访问47.110.48.1



nginx.conf的配置文件说明

```
#user nobody;
```

```
worker_processes 1; #工作进程：数目。根据硬件调整，通常等于cpu数量或者2倍cpu数量。
```

```
#错误日志存放路径
```

```
#error_log logs/error.log;
```

```
#error_log logs/error.log notice;
```

```
#error_log logs/error.log info;
```

```
#pid logs/nginx.pid; # nginx进程pid存放路径
```

```
events {
```

```
    worker_connections 1024; # 工作进程的最大连接数量
```

```
}
```

```
http {
```

```
    include mime.types; #指定mime类型，由mime.type来定义
    default_type application/octet-stream;
```

```
    # 日志格式设置
```

```
    #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
```

```
    # '$status $body_bytes_sent "$http_referer" '
```

```
    # '$http_user_agent' "$http_x_forwarded_for";
```

```
    #access_log logs/access.log main; #用log_format指令设置日志格式后，需要用access_log来定日志文件存放路径
```

sendfile on; #指定nginx是否调用sendfile函数来输出文件, 对于普通应用, 必须设置on。  
如果用来进行下载等应用磁盘io重负载应用, 可设着off, 以平衡磁盘与网络io处理速度, 降  
系统uptime。  
#tcp\_nopush on; #此选项允许或禁止使用socket的TCP\_CORK的选项, 此选项仅在sendfile的  
候使用

```
#keepalive_timeout 0; #keepalive超时时间  
keepalive_timeout 65;
```

```
#gzip on; #开启gzip压缩服务
```

```
#虚拟主机
```

```
server {
```

```
    listen 80; #配置监听端口号
```

```
    server_name localhost; #配置访问域名, 域名可以有多个, 用空格隔开
```

```
    #charset koi8-r; #字符集设置
```

```
    #access_log logs/host.access.log main;
```

```
    location / {  
        root html;  
        index index.html index.htm;  
    }
```

```
    #错误跳转页
```

```
    #error_page 404 /404.html;
```

```
    # redirect server error pages to the static page /50x.html
```

```
    #
```

```
    error_page 500 502 503 504 /50x.html;
```

```
    location = /50x.html {
```

```
        root html;
```

```
    }
```

```
    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
```

```
    #
```

```
    #location ~ \.php$ {
```

```
        # proxy_pass http://127.0.0.1;
```

```
    #}
```

```
    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
```

```
    #
```

```
    #location ~ \.php$ { #请求的url过滤, 正则匹配, ~为区分大小写, ~*为不区分大小写。
```

```
        # root html; #根目录
```

```
        # fastcgi_pass 127.0.0.1:9000; #请求转向定义的服务器列表
```

```
        # fastcgi_index index.php; # 如果请求的Fastcgi_index URI是以 / 结束的, 该指令设置的文  
会被附加到URI的后面并保存在变量$fastcig_script_name中
```

```
        # fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
```

```
        # include fastcgi_params;
```

```
    #}
```

```
    # deny access to .htaccess files, if Apache's document root
```

```
    # concurs with nginx's one
```

```

#
#location ~ /\.ht {
#    deny all;
#}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
#server {
#    listen    8000;
#    listen    somename:8080;
#    server_name somename alias another.alias;

#    location / {
#        root    html;
#        index    index.html index.htm;
#    }
#}

# HTTPS server
#
#server {
#    listen    443 ssl; #监听端口
#    server_name localhost; #域名

#    ssl_certificate    cert.pem; #证书位置
#    ssl_certificate_key    cert.key; #私钥位置

#    ssl_session_cache    shared:SSL:1m;
#    ssl_session_timeout    5m;

#    ssl_ciphers    HIGH:!aNULL:!MD5; #密码加密方式
#    ssl_prefer_server_ciphers    on; # ssl_prefer_server_ciphers    on; #

#    location / {
#        root    html;
#        index    index.html index.htm;
#    }
#}
}

```