Nginx安装手册

# nginx安装环境

nginx是C语言开发，建议在linux上运行，本教程使用Centos6.5作为安装环境。

* gcc

安装nginx需要先将官网下载的源码进行编译，编译依赖gcc环境，如果没有gcc环境，需要安装gcc：yum install gcc-c++

* PCRE

PCRE(Perl Compatible Regular Expressions)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，所以需要在linux上安装pcre库。

**yum install -y pcre pcre-devel**

注：pcre-devel是使用pcre开发的一个二次开发库。nginx也需要此库。

* zlib

zlib库提供了很多种压缩和解压缩的方式，nginx使用zlib对http包的内容进行gzip，所以需要在linux上安装zlib库。

**yum install -y zlib zlib-devel**

* openssl

OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议，并提供丰富的应用程序供测试或其它目的使用。

nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库。

**yum install -y openssl openssl-devel**

**下载nginx：**

wget -c https://nginx.org/download/nginx-1.10.1.tar.gz

# 编译安装

将nginx-1.8.0.tar.gz拷贝至linux服务器。

解压：

tar -zxvf nginx-1.8.0.tar.gz

cd nginx-1.8.0

1. Configure（默认这种方法就行了，不用执行下面的，直接./configure就行了）

./configure --help查询详细参数（参考本教程附录部分：nginx编译参数）

参数设置如下：

./configure \

--prefix=/usr/local/nginx \

--pid-path=/var/run/nginx/nginx.pid \

--lock-path=/var/lock/nginx.lock \

--error-log-path=/var/log/nginx/error.log \

--http-log-path=/var/log/nginx/access.log \

--with-http\_gzip\_static\_module \

--http-client-body-temp-path=/var/temp/nginx/client \

--http-proxy-temp-path=/var/temp/nginx/proxy \

--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \

--http-uwsgi-temp-path=/var/temp/nginx/uwsgi \

--http-scgi-temp-path=/var/temp/nginx/scgi

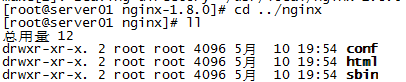
**注意：上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录**

1. 编译安装

make

make install

安装成功查看安装目录 ：



# 启动nginx

cd /usr/local/nginx/sbin/

./nginx

查询nginx进程：



15098是nginx主进程的进程id，15099是nginx工作进程的进程id

**注意：执行./nginx启动nginx，这里可以-c指定加载的nginx配置文件，如下：**

**./nginx -c /usr/local/nginx/conf/nginx.conf**

**如果不指定-c，nginx在启动时默认加载conf/nginx.conf文件，此文件的地址也可以在编译安装nginx时指定./configure的参数（--conf-path= 指向配置文件（nginx.conf））**

# 停止nginx

方式1，快速停止：

cd /usr/local/nginx/sbin

./nginx -s stop

此方式相当于先查出nginx进程id再使用kill命令强制杀掉进程。

方式2，完整停止(建议使用)：

cd /usr/local/nginx/sbin

./nginx -s quit

此方式停止步骤是待nginx进程处理任务完毕进行停止。

# 重启nginx

方式1，先停止再启动（建议使用）：

对nginx进行重启相当于先停止nginx再启动nginx，即先执行停止命令再执行启动命令。

如下：

./nginx -s quit

./nginx

方式2，重新加载配置文件：

当nginx的配置文件nginx.conf修改后，要想让配置生效需要重启nginx，使用-s reload不用先停止nginx再启动nginx即可将配置信息在nginx中生效，如下：

./nginx -s reload

# 测试

nginx安装成功，启动nginx，即可访问虚拟机上的nginx：



到这说明nginx上安装成功。

如果没有出现上图所示，可能是nginx默认的端口80端口没有打开的原因，请执行如下操作：

firewall-cmd --zone=public --add-port=80/tcp --permanent

之后重启防火墙：

systemctl stop firewalld.service

systemctl start firewalld.service

# 开机自启动nginx

## 编写shell脚本

这里使用的是编写shell脚本的方式来处理

vi /etc/init.d/nginx (输入下面的代码)

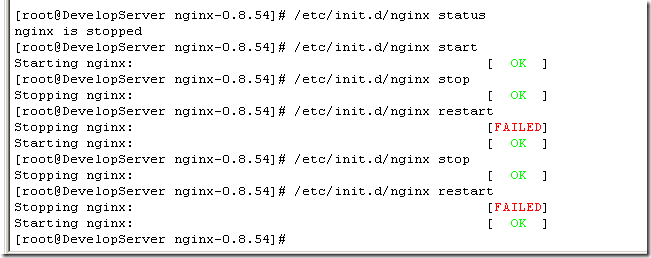
|  |
| --- |
| #!/bin/bash  # nginx Startup script for the Nginx HTTP Server  # it is v.0.0.2 version.  # chkconfig: - 85 15  # description: Nginx is a high-performance web and proxy server.  # It has a lot of features, but it's not for everyone.  # processname: nginx  # pidfile: /var/run/nginx.pid  # config: /usr/local/nginx/conf/nginx.conf  nginxd=/usr/local/nginx/sbin/nginx  nginx\_config=/usr/local/nginx/conf/nginx.conf  nginx\_pid=/var/run/nginx.pid  RETVAL=0  prog="nginx"  # Source function library.  . /etc/rc.d/init.d/functions  # Source networking configuration.  . /etc/sysconfig/network  # Check that networking is up.  [ ${NETWORKING} = "no" ] && exit 0  [ -x $nginxd ] || exit 0  # Start nginx daemons functions.  start() {  if [ -e $nginx\_pid ];then  echo "nginx already running...."  exit 1  fi  echo -n $"Starting $prog: "  daemon $nginxd -c ${nginx\_config}  RETVAL=$?  echo  [ $RETVAL = 0 ] && touch /var/lock/subsys/nginx  return $RETVAL  }  # Stop nginx daemons functions.  stop() {  echo -n $"Stopping $prog: "  killproc $nginxd  RETVAL=$?  echo  [ $RETVAL = 0 ] && rm -f /var/lock/subsys/nginx /var/run/nginx.pid  }  # reload nginx service functions.  reload() {  echo -n $"Reloading $prog: "  #kill -HUP `cat ${nginx\_pid}`  killproc $nginxd -HUP  RETVAL=$?  echo  }  # See how we were called.  case "$1" in  start)  start  ;;  stop)  stop  ;;  reload)  reload  ;;  restart)  stop  start  ;;  status)  status $prog  RETVAL=$?  ;;  \*)  echo $"Usage: $prog {start|stop|restart|reload|status|help}"  exit 1  esac  exit $RETVAL |

:wq 保存并退出

## 设置文件的访问权限

chmod a+x /etc/init.d/nginx (a+x ==> all user can execute 所有用户可执行)

这样在控制台就很容易的操作nginx了：查看Nginx当前状态、启动Nginx、停止Nginx、重启Nginx…



如果修改了nginx的配置文件nginx.conf，也可以使用上面的命令重新加载新的配置文件并运行，可以将此命令加入到rc.local文件中，这样开机的时候nginx就默认启动了

## 加入到rc.local文件中

vi /etc/rc.local

加入一行 /etc/init.d/nginx start 保存并退出，下次重启会生效。

如果遇到无法访问ftp目录下的图片，页面报404的话，这样干：

进入nginx的conf目录，编辑nginx.conf，即vi nginx.conf

之后添加如下内容

|  |
| --- |
| server{  listen 80;  server\_name 192.168.228.153; // ip地址  location /  {  root /home/ftpuser/www/; // ftp图片路径  }  } |

之后会遇到403的错误，别急这样改就行了，vi nginx.conf

添加 user root; 就ok了

当然记得每一步保存后，重新启动nginx