# Nginx+keepalived

# Nginx安装手册

<http://nginx.org/en/docs/>

1 **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。

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(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 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**

2 **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

./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

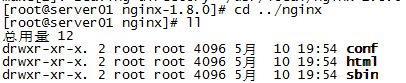
**注意：上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录**

2、 编译安装

make

make  install

安装成功查看安装目录 ：



3 **启动nginx**

cd /usr/local/nginx/sbin/

./nginx

查询nginx进程：

https://images2018.cnblogs.com/blog/1486308/201809/1486308-20180910125932368-1060620686.png

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））**

4 **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。

5 **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

6 **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。

7 **开机自启动nginx**

7.1 **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

7.2 **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

7.3 **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。

### Module ngx\_http\_upstream\_module

upstream backend {

server backend1.example.com weight=5;

server backend2.example.com:8080;

server unix:/tmp/backend3;

server backup1.example.com:8080 backup;

server backup2.example.com:8080 backup;

}

server {

location / {

proxy\_pass http://backend;

}

}

### Module ngx\_http\_proxy\_module

location / {

proxy\_pass http://localhost:8000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

}

### nginx实现动静分离

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

upstream static\_pools {

server 192.168.1.106:8080;

}

upstream dynamic\_pools {

server 192.168.1.103:8080;

}

server {

listen 80;

location / {

root html;

index index.html index.htm;

proxy\_pass http://dynamic\_pools;

}

location ~ .\*.(gif|jpg|jpeg|png|bmp|swf|css|js)$ { //如果请求包含这些后缀，请求被转发到静态池，否则转发到动态池

proxy\_pass http://static\_pools;

}

}

}

# keepalived(nginx的高可用)安装文档

1. **安装环境**

su - root

yum -y install kernel-devel\*

yum -y install openssl-\*

yum -y install popt-devel

yum -y install lrzsz

yum -y install openssh-clients

yum -y install libnl libnl-devel popt

2. **安装keepalived**

2.1. **安装keepalived**

将keepalived-1.2.15.tar.gz上传到服务器/usr/local/下。

cd /usr/local

tar -zxvf keepalived-1.2.15.tar.gz

cd keepalived-1.2.15

执行配置命令

./configure --prefix=/usr/local/keepalived

3、编译

make

4、安装

make install

至此安装成功

5、拷贝执行文件

cp /usr/local/keepalived/sbin/keepalived /usr/sbin/

6、将init.d文件拷贝到etc下,加入开机启动项

cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/keepalived

7、将keepalived文件拷贝到etc下，加入网卡配置

cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/

8、创建keepalived文件夹

mkdir -p /etc/keepalived

9、将keepalived配置文件拷贝到etc下

cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/keepalived.conf

10、添加可执行权限

chmod +x /etc/init.d/keepalived

2.2. **加入开机启动：**

chkconfig --add keepalived #添加时必须保证/etc/init.d/keepalived存在

chkconfig keepalived on

添加完可查询系统服务是否存在：chkconfig --list

2.3. **启动keepalived**

启动：service keepalived start

停止：service keepalived stop

重启：service keepalived restart

3. **配置日志文件**

1.将keepalived日志输出到local0：

vi /etc/sysconfig/keepalived

KEEPALIVED\_OPTIONS="-D -d -S 0"

2.在/etc/rsyslog.conf里添加:

local0.\*  /var/log/keepalived.log

3.重新启动keepalived和rsyslog服务：

service rsyslog restart

service keepalived restart

4. **打开防火墙的通讯地址**

iptables -A INPUT -d 224.0.0.18 -j ACCEPT

/etc/rc.d/init.d/iptables save

### 配置文件

主:

global\_defs {

    notification\_email {

        a@abc.com

        b@abc.com

    }

    notification\_email\_from alert@abc.com

    smtp\_server smtp.abc.com

    smtp\_connect\_timeout 30

    enable\_traps

    router\_id LVS\_01

}

vrrp\_instance VI\_1 {

    state MASTER

    interface eth0

    virtual\_router\_id 51

    priority 150

    advert\_int 1

    authentication {

        auth\_type PASS

        auth\_pass 1111

    }

    virtual\_ipaddress {

        10.0.0.17/24

    }

}

备:

global\_defs {

    notification\_email {

        a@abc.com

        b@abc.com

    }

    notification\_email\_from alert@abc.com

    smtp\_server smtp.abc.com

    smtp\_connect\_timeout 30

    enable\_traps

    router\_id LVS\_02

}

vrrp\_instance VI\_1 {

    state BACKUP

    interface eth0

    virtual\_router\_id 51

    priority 100

    advert\_int 1

    authentication {

        auth\_type PASS

        auth\_pass 1111

    }

    virtual\_ipaddress {

        10.0.0.17/24

    }

}

### IP切换

启动keepalived后,

ip addr |grep 10.0.0.17

查询只在主的一端可以查看到ip,表明配置和启动正常

如果两端都可以查询到ip,则表明裂脑了

如果配置和启动正常,如果主的关闭,则ip会飘到备的上面

此时,主再次启动,则主会接管ip,备会让出ip;

# 添加虚拟ip

ip addr add 10.0.0.18/24 dev eth0

ip addr del 10.0.0.18/24 dev eth0

ifconfig eth0:0 10.0.0.19 up

ifconfig eth0:0 10.0.0.19 down

keepalived多实例不能做双主,如果一个服务上有两个实例,则做一主一备