**安装必须组件**

yum install gc gcc gcc-c++ pcre-devel zlib-devel openssl-devel

**创建群组**

groupadd nginx

**创建一个用户，不允许登陆和不创主目录**

useradd -s /sbin/nologin -g nginx -M nginx

上传文件到 /usr/src目录

解压

tar zxvf nginx-1.7.8.tar.gz

编译基本能运行的nginx

./configure --user=nginx --group=nginx --prefix=/usr/local/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_gzip\_static\_module

make

make install

查看ngin进程和端口号  
[root@www ~]# netstat -ntlp | grep nginx   
tcp        0      0 0.0.0.0:80              0.0.0.0:\*              LISTEN      4415/nginx: master

step3：控制nginx服务的命令  
1、启动：nginx  
2、停止：nginx -s stop  
3、退出：nginx -s quit  
4、重启：nginx -s reopen  
5、重新加载：nginx -s reload  
6、平滑启动：kill -HUP pid（kill -HUP `cat /var/run/nginx.pid`）

由于使用源码编译安装Nginx，不会像yum install那样自动生成Nginx服务控制脚本，所以需要自己添加一个操作脚本。谷歌上搜了一下，shell脚本的代码基本类似，因此自己稍微整理改动了下。

下面的shell脚本内容，需要根据Nginx实际编译安装的配置，修改NGINX\_CONF\_FILE="/etc/nginx/nginx.conf"和lockfile=/var/lock/nginx.lock这两个参数（即前文《2015博客升级记(三)：CentOS 7.1编译安装Nginx1.9.0》中的configure配置）。

#! /bin/bash

#

# nginx - this script starts and stops the nginx daemon

#

# chkconfig: - 85 15

# description: Nginx is an HTTP(S) server, HTTP(S) reverse \

# proxy and IMAP/POP3 proxy server

#

# processname: nginx

# config: /etc/nginx/nginx.conf

# pidfile: /var/run/nginx/nginx.pid

# Source function library.

. /etc/rc.d/init.d/functions

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

[ "$NETWORKING" = "no" ] && exit 0

nginx="/usr/sbin/nginx"

prog=$(basename $nginx)

NGINX\_CONF\_FILE="/etc/nginx/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/nginx.lock

start() {

[ -x $nginx ] || exit 5

[ -f $NGINX\_CONF\_FILE ] || exit 6

echo -n $"Starting $prog: "

daemon $nginx -c $NGINX\_CONF\_FILE

retval=$?

echo

[ $retval -eq 0 ] && touch $lockfile

return $retval

}

stop() {

echo -n $"Stopping $prog: "

killproc $prog -QUIT

retval=$?

echo

[ $retval -eq 0 ] && rm -f $lockfile

return $retval

}

restart() {

configtest || return $?

stop

sleep 1

start

}

reload() {

configtest || return $?

echo -n $"Reloading $prog: "

killproc $nginx -HUP

RETVAL=$?

echo

}

force\_reload() {

restart

}

configtest() {

$nginx -t -c $NGINX\_CONF\_FILE

}

rh\_status() {

status $prog

}

rh\_status\_q() {

rh\_status >/dev/null 2>&1

}

case "$1" in

start)

rh\_status\_q && exit 0

$1

;;

stop)

rh\_status\_q || exit 0

$1

;;

restart|configtest)

$1

;;

reload)

rh\_status\_q || exit 7

$1

;;

force-reload)

force\_reload

;;

status)

rh\_status

;;

condrestart|try-restart)

rh\_status\_q || exit 0

;;

\*)

echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"

exit 2

esac

将上面的shell脚本内容以Unix格式(不能用dos哦)保存并命名为nginx，然后上传到centos系统的/etc/init.d/目录，接着添加可执行权限，最后执行添加到开机启动的命令：chkconfig --add nginx、chkconfig nginx on即可。

最后，我们可以使用如下命令对Nginx服务进行操作：

######启动Nginx服务

[root@typecodes ~]# service nginx start

######停止Nginx服务

[root@typecodes ~]# service nginx stop

######重启Nginx服务

[root@typecodes ~]# service nginx restart

######Nginx服务的状态

[root@typecodes ~]# service nginx status

######在Nginx服务启动的状态下，重新加载nginx.conf这个配置文件

[root@typecodes ~]# service nginx reload

centos 7以上是用Systemd进行系统初始化的，Systemd 是 Linux 系统中最新的初始化系统（init），它主要的设计目标是克服 sysvinit 固有的缺点，提高系统的启动速度。关于Systemd的详情介绍在[这里](http://www.ibm.com/developerworks/cn/linux/1407_liuming_init3/)。

Systemd服务文件以.service结尾，比如现在要建立nginx为开机启动，如果用yum install命令安装的，yum命令会自动创建nginx.service文件，直接用命令

|  |  |
| --- | --- |
| 1 | systemcel enable nginx.service |

设置开机启动即可。  
在这里我是用源码编译安装的，所以要手动创建nginx.service服务文件。  
开机没有登陆情况下就能运行的程序，存在系统服务（system）里，即：

|  |  |
| --- | --- |
| 1 | /lib/systemd/system/ |

**1.在系统服务目录里创建nginx.service文件**

|  |  |
| --- | --- |
| 1 | vi /lib/systemd/system/nginx.service |

内容如下

|  |  |
| --- | --- |
|  | [Unit]  Description=nginx  After=network.target    [Service]  Type=forking  ExecStart=/usr/local/nginx/sbin/nginx  ExecReload=/usr/local/nginx/sbin/nginx -s reload  ExecStop=/usr/local/nginx/sbin/nginx -s quit  PrivateTmp=true    [Install]  WantedBy=multi-user.target |

[Unit]:服务的说明  
Description:描述服务  
After:描述服务类别  
[Service]服务运行参数的设置  
Type=forking是后台运行的形式  
ExecStart为服务的具体运行命令  
ExecReload为重启命令  
ExecStop为停止命令  
PrivateTmp=True表示给服务分配独立的临时空间  
注意：[Service]的启动、重启、停止命令全部要求使用绝对路径  
[Install]运行级别下服务安装的相关设置，可设置为多用户，即系统运行级别为3

保存退出。

**2.设置开机启动**

|  |  |
| --- | --- |
| 1 | systemctl enable nginx.service |

**3.其他命令**

启动nginx服务

|  |  |
| --- | --- |
| 1 | systemctl start nginx.service |

设置开机自启动

|  |  |
| --- | --- |
|  | systemctl enable nginx.service |

停止开机自启动

|  |  |
| --- | --- |
|  | systemctl disable nginx.service |
| 查看服务当前状态 | systemctl status nginx.service |
| 重新启动服务 | systemctl restart nginx.service |

查看所有已启动的服务

|  |  |
| --- | --- |
|  | systemctl list-units --type=service |

**4.Systemd 命令和 sysvinit 命令的对照表**

****

**5.Sysvinit 运行级别和 systemd 目标的对应表**



  原文地址：<http://www.dohooe.com/2016/03/03/352.html?utm_source=tuicool&utm_medium=referral>