

一、安装nginx依赖的软件

nginx是C写的，需要用GCC编译；nginx中的rewrite module需要PCRE；nginx中的gzip module需要zlib；nginx中的HTTP SSL module需要OpenSSL。

1.1 gcc安装

因为后面安装需要编译，所以事先得先安装gcc编译。阿里云主机已经默认安装了 gcc，如果是自己安装的虚拟机，那么需要先安装一下 gcc：

1.1.1 yum安装

```
yum -y install gcc gcc-c++ autoconf automake make

# 升级gcc为9.3
# 查看gcc版本是否在5.3以上，centos7.6默认安装4.8.5
gcc -v
# 升级gcc到5.3及以上，如下：
升级到gcc 9.3：
yum -y install centos-release-scl
yum -y install devtoolset-9-gcc devtoolset-9-gcc-c++ devtoolset-9-binutils
scl enable devtoolset-9 bash

#需要注意的是scl命令启用只是临时的，退出shell或重启就会恢复原系统gcc版本。
#如果要长期使用gcc 9.3的话：
echo "source /opt/rh/devtoolset-9/enable" >>/etc/profile
#这样退出shell重新打开就是新版的gcc了
```

1.1.2 rpm安装

gcc是linux下的一个编译程序，是C程序的编译工具。

GCC(GNU Compiler Collection) 是 GNU(GNU's Not Unix) 计划提供的编译器家族，它能够支持 C, C++, Objective-C, Fortran, Java 和 Ada 等等程序设计语言前端，同时能够运行在 x86, x86-64, IA-64, PowerPC, SPARC 和 Alpha 等等几乎目前所有的硬件平台上。鉴于这些特征，以及 GCC 编译代码的高效性，使得 GCC 成为绝大多数自由软件开发编译的首选工具。虽然对于程序员们来说，编译器只是一个工具，除了开发和维护人员，很少有人关注编译器的发展，但是 GCC 的影响力是如此之大，它的性能提升甚至有望改善所有的自由软件的运行效率，同时它的内部结构的变化也体现出现代编译器发展的新特征。

下载依赖包：

官网：<https://pkgs.org/>

百度云下载（已下载好所有依赖包）：

链接：https://pan.baidu.com/s/1B_4et9GsH51LKWKTD5bitQ

提取码：82b4

使用WinSCP上传到服务器后，运行如下命令：

```
rpm -Uvh *.rpm --nodeps --force
```






安装好，查询下：

```
gcc -v
```

```
[root@localhost gcc]# gcc -v
使用内建 specs。
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
目标: x86_64-redhat-linux
配置为: ../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-cxx-exceptions --disable-libmwind-exceptions --enable-gnu-unwind-exceptions --enable-lto --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-ar-y --disable-libgcj --with-isl=/build/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install --with-cloog=/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
线程模型: posix
gcc 版本 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
[root@localhost gcc]#
```

离线升级gcc为9.3，编译安装章节中有在线升级。下载gcc升级包：

电脑 > 工作 (D:) > Tools > codeTools > gcc > 升级9.3

名称	修改日期	类型	大小
 gcc-9.3.0.tar.gz	2020/5/8 10:51	360压缩	121,231 KB
 gmp-6.1.0.tar.bz2	2020/5/8 11:06	360压缩	2,328 KB
 isl-0.18.tar.bz2	2020/5/8 11:06	360压缩	1,620 KB
 mpc-1.0.3.tar.gz	2020/5/8 11:05	360压缩	655 KB
 mpfr-3.1.4.tar.bz2	2020/5/8 11:05	360压缩	1,250 KB

解压gcc-9.3.0.tar.gz：

```
tar -zxvf gcc-9.3.0.tar.gz
```

将另外四个上传到解压目录gcc-9.3.0下。

进入gcc-9.3.0目录下：

```
cd gcc-9.3.0/
./contrib/download_prerequisites
```

若出现找不到wget或tar: bzip2: 无法 exec: 没有那个文件或目录错误，需要安装wget和bzip2：

```
yum install -y wget
yum install -y bzip2
```

离线安装可自行百度

安装完成后再次命令：

```
./contrib/download_prerequisites
```

#创建预编译目录

```
mkdir build && cd build
```

#设置编译选项并编译

```
../configure --prefix=/usr/local/gcc-9.3.0 --enable-bootstrap --enable-checking=release --enable-languages=c,c++ --disable-multilib
```

```
#安装
#编译生成makefile文件
make
#安装GCC
make install

#安装后的设置
#设置环境变量
touch /etc/profile.d/gcc.sh
sudo chmod 777 /etc/profile.d/gcc.sh
sudo echo -e '\nexport PATH=/usr/local/gcc-9.3.0/bin:$PATH\n' >>
/etc/profile.d/gcc.sh && source /etc/profile.d/gcc.sh

#设置头文件
sudo ln -sv /usr/local/gcc/include/ /usr/include/gcc

#设置库文件
touch /etc/ld.so.conf.d/gcc.conf
sudo chmod 777 /etc/ld.so.conf.d/gcc.conf
sudo echo -e "/usr/local/gcc/lib64" >> /etc/ld.so.conf.d/gcc.conf

#加载动态连接库
sudo ldconfig -v
ldconfig -p |grep gcc

#测试版本号
gcc -v
```

1.2 zlib安装

1.2.1 yum安装

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

1.2.2 源码安装

1、下载

zlib官网: <http://www.zlib.net/>

下载zlib-1.2.11: <http://prdownloads.sourceforge.net/libpng/zlib-1.2.11.tar.gz>

百度云:

链接: https://pan.baidu.com/s/1t_nqoz4UnsCKNzfM8T2sWA

提取码: mb1y

2、解压并进入zlib代码根目录

通过WinSCP将下载的压缩包上传到服务器。

```
tar -zxvf zlib-1.2.11.tar.gz  
  
cd zlib-1.2.11
```

3、配置、编译、安装

```
# 配置  
./configure  
# 编译  
make  
# 安装  
make install  
  
# 查看  
whereis zlib
```

1.3 PCRE安装

1.3.1 yum安装

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

1.3.2 源码安装

1、下载

PCRE官网: <http://www.pcre.org/>

下载pcre-8.41: <http://downloads.sourceforge.net/project/pcre/pcre/8.41/pcre-8.41.tar.gz>

百度云:

链接: https://pan.baidu.com/s/1t_nqoz4UnsCKNzfM8T2sWA

提取码: mb1y

2、解压并进入PCRE代码根目录

通过WinSCP将下载的压缩包上传到服务器。

```
tar -zxvf pcre-8.41.tar.gz  
  
cd pcre-8.41
```

3、配置、编译、安装

```
# 配置
./configure
# 编译
make
# 安装
make install

# 查看
pcre-config --version
```

1.4 OpenSSL安装

1.4.1 yum安装

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

1.4.2 源码安装

1、下载

OpenSSL官网: <https://www.openssl.org/>

下载pcre-8.41: <https://www.openssl.org/source/openssl-1.0.2n.tar.gz>百度云: 链接: https://pan.baidu.com/s/1t_nqoz4UnsCKNzfM8T2sWA 提取码: mb1y

2、解压并进入openssl代码根目录

通过WinSCP将下载的压缩包上传到服务器。

```
tar -zxvf openssl-1.0.2n.tar.gz

cd openssl-1.0.2n
```

3、配置、编译、安装

```
# 配置
./config
# 编译
make
# 安装
make install

# 查看版本
openssl version

whereis openssl
```

二、安装nginx

2.1 下载

官网下载nginx-1.14.2: <http://nginx.org/download/>

百度云:

链接: https://pan.baidu.com/s/1t_nqoz4UnsCKNzfM8T2sWA

提取码: mb1y

2.2 解压并进入nginx代码根目录

通过WinSCP将下载的压缩包上传到服务器。

```
tar zxvf nginx-1.14.2.tar.gz
```

```
cd nginx-1.14.2
```

2.3 配置

```
# --with-http_stub_status_module 开启状态查询
# --with-http_ssl_module 开启ssl加密
# --with-pcre=../pcre-8.41 --with-zlib=../zlib-1.2.11 --with-openssl=../openssl-1.0.2n 指定pcre和zlib和openssl版本
./configure --with-http_stub_status_module --with-http_ssl_module --with-pcre=../pcre-8.41 --with-zlib=../zlib-1.2.11 --with-openssl=../openssl-1.0.2n
```

2.4 编译安装

```
make
```

```
make install
```

2.5 检查nginx.conf配置正确性

```
/usr/local/nginx/sbin/nginx -t
```

```
[root@localhost nginx-1.14.2]# /usr/local/nginx/sbin/nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
[root@localhost nginx-1.14.2]#
```

2.6 启动nginx

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

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

2.7 nginx 常用命令

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

- 1、启动nginx，直接运行nginx的可执行文件

```
./nginx
```

- 2、停止nginx

```
./nginx -s stop
```

```
# 优雅关闭  
./nginx -s quit
```

- 3、重新加载配置文件：nginx -s reload 或者 kill -HUP 主进程号

```
./nginx -s reload
```

- 4、测试配置文件，检查配置文件语法是否正确，然后试图打开文件涉及的配置：nginx -t 配置文件

```
./nginx -t
```

- 5、查看nginx版本信息：

```
./nginx -v
```

- 6、查看nginx版本信息，编译版本，和配置参数

```
./nginx -V
```

- 7、重启日志文件，备份日志文件时常用：./nginx -s reopen 或者 kill -USR1 主进程号

```
./nginx -s reopen
```

2.8 nginx加入到开机自动启动

- 1、开机自启

先编辑系统启动脚本vim /etc/rc.local 这个文件是系统启动后会自动执行的,我们就将启动命令加入到这个文件中

```
vim /etc/rc.local

/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf &
>/tmp/nginx.log 2>&1

chmod +x /etc/rc.d/rc.local
```

或者:

```
cd /lib/systemd/system/

vim nginx.service
[Unit]
Description=nginx service
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

# 保存退出
# 加载生效
systemctl daemon-reload

# 重启后生效, 发现nginx自启, 并且可以使用下面的命令了
```

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

# systemctl start nginx	启动nginx服务
# systemctl stop nginx	停止服务
# systemctl restart nginx	重新启动服务
# systemctl list-units --type=service	查看所有已启动的服务
# systemctl status nginx	查看服务当前状态
# systemctl enable nginx	设置开机自启动
# systemctl disable nginx	停止开机自启动

2、修改全局的环境变量，使得nginx命令在任意目录下都能执行,例如：停止nginx -s stop ,重新加载：nginx -s reload ,启动：nginx

```
vim /etc/profile
export PATH=$PATH:/usr/local/nginx/sbin

# 使之生效
source /etc/profile

# 停止
nginx -s stop
# 重新加载
nginx -s reload
# 启动
nginx
```

```
#Nginx的默认安装位置
/usr/local/nginx

# nginx的配置文件
/usr/local/nginx/conf/nginx.conf

# 默认的网页文件
/usr/local/nginx/html
```