利用脚本也能编译和安装软件,但是制作rpm包的好处是方便管理,例如卸载软件问题,直接yum remove xxx 就行了

制作nginx rpm包的过程:

1. yum install rpm-build rpmdevtools
2. rpmdev-setuptree 会在家目录下生成rpmbuild目录

[root@localhost rpmbuild]# ls

BUILD RPMS SOURCES SPECS SRPMS

1. 将nginx的tar.gz包和第三方模块放入SOURCES目录

[root@www SOURCES]# ls

nginx-1.8.1.tar.gz nginx-3rd

1. 在SPECS目录下创建nginx.spec文件,此时的文件会默认生成一些内容

nginx.spec:

|  |
| --- |
| Name: nginx  Version: 1.8.1  Release: 1%{?dist}  Summary: nginx web server  Group: Applications/webServer  License: GPL  URL: http://www.a.com  #源码名称  Source0: %{name}-%{version}.tar.gz  BuildRequires: gcc,make  #Requires: pcre-devel,geoip-devel  %description  %prep  #cd 到SOURCES并解压里面的tar.gz包  %setup -q  %build  ./configure \  --user=nginx \  --group=nginx \  --prefix=/usr/share/nginx \  --sbin-path=/usr/sbin/nginx \  --conf-path=/etc/nginx/nginx.conf \  --error-log-path=/var/log/nginx/error.log \  --http-log-path=/var/log/nginx/access.log \  --pid-path=/var/run/nginx.pid \  --lock-path=/var/lock/subsys/nginx\  --with-http\_secure\_link\_module \  --with-http\_random\_index\_module \  --with-http\_ssl\_module \  --with-http\_realip\_module \  --with-http\_addition\_module \  --with-http\_sub\_module \  --with-http\_dav\_module \  --with-http\_flv\_module \  --with-http\_gzip\_static\_module \  --with-http\_stub\_status\_module \  --with-ipv6 \  --with-http\_perl\_module \  --with-http\_geoip\_module \  --with-mail \  --with-mail\_ssl\_module \  --add-module=%{\_sourcedir}/nginx-3rd/nginx-upstream-fair \  --add-module=%{\_sourcedir}/nginx-3rd/nginx\_mod\_h264\_streaming \  --add-module=%{\_sourcedir}/nginx-3rd/ngx\_cache\_purge  #--with-ld-opt='-L -Wl,-rpath,/usr/local/luajit/lib'  #--add-module=/root/rpmbuild/BUILD/nginx-1.8.1/lua-nginx-module-0.9.5rc2  #--add-module=/root/nginx/nginx-3rd/mod\_zip \  #--add-module=/root/nginx/nginx-3rd/ngx\_devel\_kit \  #--http-client-body-temp-path=/var/lib/nginx/tmp/client\_body \  #--http-proxy-temp-path=/var/lib/nginx/tmp/proxy \  #--http-fastcgi-temp-path=/var/lib/nginx/tmp/fastcgi \  #nginx-upstream-fair 比内建负载均衡模块更加好的负载均衡模块  #nginx\_mod\_h264\_streaming map4等流媒体  #ngx\_cache\_purge清楚缓存模块,不需要任何配置  #--prefix 指示web目录的路径 相当于 httpd的/var/www/html目录  make %{?\_smp\_mflags}  #安装阶段  %install  %make\_install  %clean  %files  /etc/  /usr/  /var/  %doc  %changelog |

1. rpmbuild -bb nginx.spec 此时就生成了rpm包在RPMS目录下
2. rpm -qi nginx 查看安装好的nginx的信息

[root@www ~]# rpm -qi nginx

Name : nginx Relocations: (not relocatable)

Version : 1.8.1 Vendor: (none)

Release : 1.el6 Build Date: Tue 10 Jul 2018 06:09:39 AM PDT

Install Date: Wed 10 Oct 2018 01:38:17 AM PDT Build Host: www.nat.com

Group : Applications/webServer Source RPM: nginx-1.8.1-1.el6.src.rpm

Size : 10832657 License: GPL

Signature : (none)

URL : http://www.a.com

Summary : nginx web server

Description :

注意的问题:

1. 编译生成的nginx,运行会报错误: nginx: [emerg] getpwnam("nginx") failed

办法: useradd -s /sbin/nologin -M nginx

1. nginx.spec中的宏变量怎么查看: rpm –showrc

主要的宏变量:

%{\_topdir} %{getenv:HOME}/rpmbuild

%{\_builddir} %{\_topdir}/BUILD

%{\_rpmdir} %{\_topdir}/RPMS

%{\_sourcedir} %{\_topdir}/SOURCES

%{\_specdir} %{\_topdir}/SPECS

%{\_srcrpmdir} %{\_topdir}/SRPMS

%{\_buildrootdir} %{\_topdir}/BUILDROOT

1. 在nginx.spec中%file下面的三行 /etc/,/usr/,/var/这么写的目的是存入rpm包,如果不写这三行则安装生成的rpm包后是没有/etc/nginx,/var/nginx,/usr/nginx等目录的,

这三行是根据BUILDROOT目录中有这三个目录

[root@www nginx-1.8.1-1.el6.x86\_64]# pwd

/root/rpmbuild/BUILDROOT/nginx-1.8.1-1.el6.x86\_64

[root@www nginx-1.8.1-1.el6.x86\_64]# ls

etc usr var