1.安装fpm

```
yum -y install ruby rubygems ruby-devel rpm-build
gem sources -a http://ruby.taobao.org/
gem sources --remove http://rubygems.org/
gem install fpm
```

2.编译nginx并创建虚拟目录

```
mkdir -pv /opt/nginx/{etc/rc.d/init.d,sysconfig}
tar -zxvf nginx-1.6.2.tar.gz
cd nginx-1.6.2
./configure --prefix=/usr/local/nginx --pid-path=/usr/local/nginx/nginx.pid --lock-path=/var/lock/nginx.lock --user=nginx --group=nginx --with-http_ssl_module --with-http_dav_module --with-http_flv_module --with-http_realip_module --with-http_gzip_static_module --with-http_stub_status_module --with-mail --with-mail_ssl_module --with-debug --http-client-body-temp-path=/usr/local/nginx/client --http-proxy-temp-path=/usr/local/nginx/proxy --http-fastcgi-temp-path=/usr/local/nginx/fastcgi --http-uwsgi-temp-path=/usr/local/nginx/uwsgi --http-scgi-temp-path=/usr/local/nginx/scgi
make
make install DESTDIR=/opt/nginx
```

3.查看下/opt/nginx的目录结构

```
├── after_install.sh
├── after_remove.sh
├── etc
│   ├── rc.d
│   │   └── init.d
│   │       └── nginx
│   └── sysconfig
└── usr
    └── local
        └── nginx
            ├── conf
            │   ├── fastcgi.conf
            │   ├── fastcgi.conf.default
            │   ├── fastcgi_params
            │   ├── fastcgi_params.default
            │   ├── koi-utf
            │   ├── koi-win
            │   ├── mime.types
            │   ├── mime.types.default
            │   ├── nginx.conf
            │   ├── nginx.conf.default
            │   ├── scgi_params
            │   ├── scgi_params.default
            │   ├── uwsgi_params
            │   ├── uwsgi_params.default
            │   ├── vhost
            │   └── win-utf
            ├── html
            │   ├── 50x.html
            │   └── index.html
            ├── logs
            └── sbin
                └── nginx
```

这里我把已经更改过的nginx配置存放到了里面，以及nginx的启动脚本放到了/opt/nginx/etc/rc.d/init.d/下面，两个shell脚本一个是在安装完成后创建nginx运行用户，和卸载rpm包之后执行的操作

```
root@saltstack. ~# cat /opt/nginx/after_install.sh
#!/bin/bash
useradd www
root@saltstack. ~# cat /opt/nginx/after_remove.sh
#!/bin/bash
rm -rf /usr/local/nginx
```

nginx启动脚本如下:

```bash
#!/bin/bash
# chkconfig: 2345 80 12
# processname: nginx
# qq:455471484
source /etc/init.d/functions


[ -r "$SYSCONFIG" ] && source "$SYSCONFIG"
RETVAL=0
start() {
    echo -n $"Starting Nginx:"
    daemon /usr/local/nginx/sbin/nginx
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/nginx
    return $RETVAL
}

stop() {
   echo -n $"Stoping Nginx:"
   daemon kill -QUIT `cat /usr/local/nginx/nginx.pid`
   RETVAL=$?
   echo
   [ $RETVAL -eq 0 ] && rm -rf /var/lock/subsys/nginx
   return $RETVAL
}

restart() {
   echo -n $"Restart Nginx:"
   daemon /usr/local/nginx/sbin/nginx -s reload
   RETVAL=$?
   echo
   return $RETVAL

}


case "$1" in
   start)
      start
      ;;
   stop)
      stop
      ;;
   restart|reload)
      restart
      ;;
esac
exit $RETVAL
```

虚拟目录下的东西在你打完fpm之后文件都会跟里面的一样，也就是说，如果你们公司有自己的配置文件模板，可以预先直接打到rpm包里面，下面就可以使用fpm命令进行build了。fpm提供了很多参数，可以--help看下根据需求~

4.打包命令:

```
fpm -f -s dir -t rpm -n qding-nginx --epoch 0 -v1.6.2 --iteration 1.el6 -C /opt/nginx/ -p /root/ -d 'openssl-devel' -d 'gcc' -d 'gcc-c++' -d 'pcre-devel' -d 'zlib-devel' --verbose --category 'Development/Languages' --description 'nginx applications' --url 'nginx.org' --license 'BSD' -m 'bianji' --no-rpm-sign --after-install /opt/nginx/after_install.sh  --after-remove after_remove.sh --workdir /opt/nginx/ etc usr
```

我使用的参数说明:

```
FPM参数详解
Usage: fpm [options]

-p, --package PACKAGEFILE        管理的软件包

-n, --name PACKAGENAME           定义生成的软件包的名字

-v, --version VERSION            定义生成的软件包的版本

    --iteration ITERATION        为软件包设置 iteration
值 ('release' for RPM).

    --epoch EPOCH                为软件包设置 epoch值

-d, —depends DEPENDENCY          设置软件包的依赖关系


-C DIRECTORY                     在搜索files前先进入该目录

-t PACKAGE_TYPE                  设置目标包的类型

-s SOURCE_TYPE                   设置需要转换的包类型
--after-remote           卸载后执行脚本
--after-install 安装后执行脚本
```

执行以下，这个输出还是很友好的:

```
fpm -f -s dir -t rpm -n qding-nginx --epoch 0 -v1.6.2 --
iteration 1.el6 -C /opt/nginx/ -p /root/ -d 'openssl-devel' -
d 'gcc' -d 'gcc-c++' -d 'pcre-devel' -d 'zlib-devel' --
verbose --category 'Development/Languages' --
description 'nginx applications' --url 'nginx.org' --
license 'BSD' -m 'bianji' --no-rpm-sign --after-install /opt/
nginx/after_install.sh  --after-remove after_remove.sh --
workdir /opt/nginx/ etc usr
Setting workdir {:workdir=>"/opt/nginx/", :level=>:info}
Setting from flags: category=Development/
Languages {:level=>:info}
Setting from flags: description=nginx applications {:level=>:
info}
Setting from flags: epoch=0 {:level=>:info}
Setting from flags: iteration=1.el6 {:level=>:info}
Setting from flags: license=BSD {:level=>:info}
Setting from flags: maintainer=bianji {:level=>:info}
Setting from flags: name=qding-nginx {:level=>:info}
Setting from flags: url=nginx.org {:level=>:info}
Setting from flags: version=1.6.2 {:level=>:info}
Converting dir to rpm {:level=>:info}
Force flag given. Overwriting package at /root/qding-
nginx-1.6.2-1.el6.x86_64.rpm {:level=>:warn}
Reading template {:path=>"/usr/lib/ruby/gems/1.8/gems/
fpm-1.3.3/templates/rpm.erb", :level=>:info}
Running rpmbuild {:args=>["rpmbuild", "-bb", "--
define", "buildroot /opt/nginx/package-rpm-
build20150415-14369-365eos/BUILD", "--define", "_topdir /opt/
nginx/package-rpm-build20150415-14369-365eos", "--
define", "_sourcedir /opt/nginx/package-rpm-
build20150415-14369-365eos", "--define", "_rpmdir /opt/nginx/
package-rpm-build20150415-14369-365eos/RPMS", "--
define", "_tmppath /opt/nginx/", "/opt/nginx/package-rpm-
build20150415-14369-365eos/SPECS/qding-
nginx.spec"], :level=>:info}
Executing(%prep): /bin/sh -e /opt/nginx/rpm-
tmp.fFuA4s {:level=>:info}
Executing(%build): /bin/sh -e /opt/nginx/rpm-
tmp.HMWMva {:level=>:info}
Executing(%install): /bin/sh -e /opt/nginx/rpm-
tmp.N2vnXR {:level=>:info}
Processing files: qding-
nginx-1.6.2-1.el6.x86_64 {:level=>:info}
Wrote: /opt/nginx/package-rpm-build20150415-14369-365eos/
RPMS/x86_64/qding-
nginx-1.6.2-1.el6.x86_64.rpm {:level=>:info}
Executing(%clean): /bin/sh -e /opt/nginx/rpm-
tmp.lyFWXl {:level=>:info}
Created package {:path=>"/root/qding-
nginx-1.6.2-1.el6.x86_64.rpm"}
```