

编译安装nginx1.9.7+php7.0.0服务器环境 - Web 烤猫

- **推荐** 9 推荐
- **收藏** 65 收藏, 4.1k 浏览

一直以来我都通过网上的一些材料去搭建lnmp环境，通过直接yum安装nginx mysql php等软件。但是为了原生态的编译安装最新的软件版本，我决定自己亲手搭建lnmp环境，采用最新的nginx1.9.7(昨天出了1.9.8)和php7来研究如何搭建起nginx最新版本和php7的环境。

nginx的编译安装

编译环境

在linux使用make方式安装，需要保证linux已经具备比较OK的编译环境，例如gcc等编译工具。一般而言，服务器提供商在安装的系统中已经默认集成了这些软件，但是为了保险起见，我们还是通过一些较为基础的方式，把这些依赖包都跑一遍，以防在之后的编译中出差错。

```
$ yum -y install gcc gcc-c++ autoconf automake libtool make  
cmake  
$ yum -y install zlib zlib-devel openssl openssl-devel pcre-  
devel
```

zlib: 为nginx提供gzip模块，需要zlib库支持

openssl: 为nginx提供ssl功能

pcre: 为支持地址重写rewrite功能

确保pcre已正确安装

在上面的基础环境安装中，我们已经yum安装了pcre，但是部分服务器上并没有pcre

的安装包，所以我们可以通过下载tar包自行编译安装pcre。

搜索pcre，进入其官网，找到最新的版本，复制tar.gz的下载连接，执行如下操作：

```
$ wget http://pcre/xxx/xxx/xxx最新版xxx.tar.gz
$ tar zxvf xxxx.tar.gz
$ cd xxxx
$ ./configure
$ make install
```

这样就可以保证安装pcre了。

创建用来运行nginx的用户及组

我们创建一个新的用户和用户组来运行nginx，这样可以把nginx和root分开，保证nginx不具备root权限。但是，我们并不希望nginx成为一个真实的可以登陆到远程进行操作的用戶，所以，我们并不给它创建家目录，在useradd的时候，用-M参数：

```
$ groupadd nginx
$ useradd -g nginx -M nginx
```

-g参数为nginx用户指定了一个组。-M参数保证其不自动生成home目录。

但通过上面的用户创建之后，nginx用户可以通过设置一个密码登陆到服务器，这个不是我们想要的，我们禁用它的ssh登陆权限.禁止用户登陆也很方便，只需要修改配置文件中有关用户和用户组的信息即可。

```
$ vi /etc/passwd
```

找到nginx，将后面的/bin/bash改为/sbin/nologin即可。

OK，用户处理完毕。

编译安装Nginx

前面讲了这么多，都还没有说到重点，接下来我们来编译安装nginx。先进入nginx的官方网站，找到最新版本的tar.gz包的链接，复制链接，然后执行下面的动作：

```
$ wget http://xxxxxxxxxx/nginx1.7.x.tar.gz
$ tar zxvf nginx1.7.x.tar.gz
$ cd nginx1.7.x
```

接下来我们需要执行./configure，不同的开发者有不同的习惯，对于刚入门的用户而言，不是很喜欢麻烦的去进行配置，总是希望默认就是最好的，但是实际情况恰好相反，走上linux的道，就请热爱折腾，既然选择编译安装，就请热爱make。

我希望把软件安装在/user/local下面，一个软件一个目录，在以后卸载的时候也比较方便清理。软件的目录也按照linux的目录形式，用conf、etc、sbin、run等进行归类管理，所以，我最终的./configure配置如下：

```
$ ./configure --prefix=/usr/local/nginx \
--pid-path=/usr/local/nginx/run/nginx.pid \
--with-http_ssl_module \
--user=nginx \
--group=nginx \
--with-pcre \
--without-mail_pop3_module \
--without-mail_imap_module \
--without-mail_smtp_module
```

末尾三个是禁用nginx作为邮件代理服务器，我一般只用服务器作为网站或数据库的服务器，所以这里把它们禁用掉，你如果想搭建的是邮件服务器，那么就应该去阅读nginx搭建邮件服务器的教程。

你可以认真阅读一下./configure的结果，看看有没有报错，或者加载的模块是不是都齐全，如果一切OK，那么往下继续，如果感觉不对，可以用./configure --help认真阅读一下。

```
$ make
$ make install
```

make的地方有一个小技巧，如果服务器是双核，可以通过-j2来指定用双核进行编译，-j4代表4核编译。

安装到这里就结束了，但是，安装完可没完事儿，nginx还没有运行起来，你可以先去看看安装的结果，并且运行nginx服务器：

```
$ cd /usr/local/nginx
$ ls
$ sbin/nginx
```

这样就运行起来了，访问你的服务器ip，看看能否看到nginx的欢迎页面吧。（不要让其他软件占用80端口哦）默认情况下网页文件放在/usr/local/nginx/html下，不符合我们的使用习惯，这个需要修改nginx的配置文件来修改，不过即使不修改，我们也是可以正常使用的，我们就不详细解释nginx的配置了。

nginx服务的载入

但是，make编译安装的软件，可不像yum安装的服务，我们熟悉的service命令并不起效，不然你用service nginx restart试试看。这是因为service调用/etc/ini.d/目录下的程序完成，而该目录下并不存在nginx这个程序。那么这个时候怎么重启nginx呢？如下操作：

```
$ /usr/local/nginx/sbin/nginx -s reload
```

这个操作可以重新加载nginx的配置文件，相当于重启(当配置文件出错时，不会重启)。如果一定要重启整个服务，那只能通过杀死nginx进程，然后在运行程序了。

不过为了使用我们熟悉的service操作，这里提供一个[程序](#)，放到/etc/ini.d/目录下，并执行：

```
$ chmod +x /etc/init.d/nginx
$ chkconfig --add nginx
$ chkconfig nginx on
```

这样就可以通过service nginx restart等方法来操作nginx了。你可以把程序下载下来，简单研究一下，如果你的nginx安装路径和我的不同，还要修改程序开头的变量设置。

PHP7的编译安装

依赖环境

惯例是先解决一些编译的依赖包

```
$ yum -y install libxml2 libxml2-devel openssl openssl-devel  
curl-devel libjpeg-devel libpng-devel freetype-devel  
libmcrypt-devel
```

编译安装php7

在《[lamp升级php至php7](#)》一文中，我谈到了如何在apache的服务器环境中升级php到7，而到nginx环境下，我们不再使用php的apxs模块，而是直接使用php-fpm模块。接下来，我们来尝试编译安装php7。

先从官方网站下载php7，并且解压，由于上面这篇文章已经有了相关步骤，就不做过多详解：

```
$ wget http://am1.php.net/get/php-  
7.0.0.tar.gz/from/this/mirror  
$ tar zvxf php-7.0.0.tar.gz  
$ cd php-7.0.0
```

接下来要进行编译前的配置，和上面一篇文章不同，我们不提供apxs参数，相反，我们提供php-fpm相关参数：

```
$ ./configure --prefix=/usr/local/php7 \  
--with-config-file-path=/usr/local/php7/etc \  
--with-config-file-scan-dir=/usr/local/php7/etc/php.d \  
--with-mcrypt=/usr/include \  
--with-openssl=/usr/local/ssl
```

```
--enable-mysqlnd \  
--with-mysqli \  
--with-pdo-mysql \  
--enable-fpm \  
--with-fpm-user=nginx \  
--with-fpm-group=nginx \  
--with-gd \  
--with-iconv \  
--with-zlib \  
--enable-xml \  
--enable-shmop \  
--enable-sysvsem \  
--enable-inline-optimization \  
--enable-mbregex \  
--enable-mbstring \  
--enable-ftp \  
--enable-gd-native-ttf \  
--with-openssl \  
--enable-pcntl \  
--enable-sockets \  
--with-xmlrpc \  
--enable-zip \  
--enable-soap \  
--without-pear \  
--with-gettext \  
--enable-session \  
--with-curl \  
--with-jpeg-dir \  
--with-freetype-dir \  
--enable-opcache
```

配置无误后执行：

```
$ make  
$ make install
```

同样可以使用-j2哦。如果安装成功，OK，那么php7的安装就OK了。

调整php配置

默认安装好之后，你会发现/usr/local/php7/etc下面没有php.ini文件，这个去哪里要呢？在php7的源码安装包都有。

```
$ cd /usr/src/php-7.0.0/  
$ ls
```

可以看到有两个php.ini-xxx文件，我们可以分别vi打开来看下，一个是产品模式，一个是开发模式。

```
$ cp php.ini-production /usr/local/php7/etc/php.ini  
$ vi /usr/local/php7/etc/php.ini
```

可以看到php的配置。本文就不做过多的配置解释了。

启用php-fpm服务

上面我们在编译php7的时候，已经将fpm模块编译了，那么接下来，我们要启用php-fpm。但是默认情况下它的配置文件和服务都没有启用，所以要我们自己来搞定。

搞定配置文件：

```
$ cd /usr/local/php7/etc  
$ mv php-fpm.conf.default php-fpm.conf  
$ mv php-fpm.d/www.conf.default php-fpm.d/www.conf
```

php-fpm的具体配置我们也不做深入去详解，因为在编译之前./configure的时候，我们都已经确定了一些配置，比如运行fpm的用户和用户组之类的，所以默认配置应该不会出现路径问题和权限问题。

搞定php-fpm的服务载入：

就像上面的nginx一样，我们希望使用service php-fpm start|stop|restart这些操作来实

现服务的重启，但没有像nginx那么复杂，php编译好之后，给我们提供了一个php-fpm的程序，不需要我再编写分享了。这个文件放在php编译源码目录中：

```
$ cd /usr/src/php-7.0.0/sapi/fpm
$ ls
$ cp init.d.php-fpm /etc/init.d/php-fpm
$ chmod +x /etc/init.d/php-fpm
$ chkconfig --add php-fpm
$ chkconfig php-fpm on
```

通过上面这个操作，我们就可以使用`sevice php-fpm start`来启用php-fpm了。用`ps -ef | grep php-fpm`看看进程吧。

nginx代理php实现访问

通过上面的操作，nginx和php-fpm服务都被我们跑起来了，但是php-fpm走的是127.0.0.1:9000，外网是无法访问的，而且我们也不可能直接通过php-fpm给外网提供服务，我们用nginx去代理9000端口执行php。

实际上这个过程只需要对nginx进行配置即可，fpm已经在后台运行了，我们需要在nginx的配置文件中增加代理的规则，即可让用户在访问80端口，请求php的时候，交由后端的fpm去执行，并返回结果。

```
$ vi /usr/local/nginx/conf/nginx.conf
```

如果你大致了解过nginx的配置，应该能够很快分辨出这个配置文件里面的结构，并且知道server代表一个虚拟主机，要增加虚拟主机就再增加一个server，而且这个conf文件中也给出了例子。那么怎么代理php-fpm呢？找到：

```
#location ~ /\.php$ {
#    root          html;
#    fastcgi_pass  127.0.0.1:9000;
#    fastcgi_index  index.php;
#    fastcgi_param  SCRIPT_FILENAME
/script$fastcgi_script_name;
```



```
# include fastcgi_params;
#}
```

把前面的#注释符号去掉，把script改为\$document_root最终如下：

```
location ~ /\.php$ {
    root            html;
    fastcgi_pass    127.0.0.1:9000;
    fastcgi_index   index.php;
    fastcgi_param   SCRIPT_FILENAME
/$document_root$fastcgi_script_name;
    include         fastcgi_params;
}
```

这样就OK了，重新载入nginx配置即可

```
$ service nginx reload
```

然后到/usr/local/nginx/html去写一个php文档，进行测试吧。

如果你的程序能够正常运行起来，用ip作为外网访问地址访问成功，那么恭喜你，本篇文章的目的就达到了。